CLASS XII

INFORMATICS PRACTICES

CENTRAL BOARD OF SECONDARY EDUCATION
Shiksha Kendra, 2, Community Centre, Preet Vihar, Delhi-110 092 India

नया आगाज़

आज समय की माँग पर
आगाज़ नया इक होगा
निरंतर योग्यता के निर्णय से
परिणाम आकलन होगा।

परिवर्तन नियम जीवन का
नियम अब नया बनेगा
अब परिणामों के भय से
नहीं बालक कोई डरेगा

निरंतर योग्यता के निर्णय से
परिणाम आकलन होगा।

बदले शिक्षा का स्वरूप
नई खिले आशा की धूप
अब किसी कोमल-से मन पर
कोई बोझ न होगा

निरंतर योग्यता के निर्णय से
परिणाम आकलन होगा।
नई राह पर चलकर मंज़िल को हमें पाना है
इस नए प्रयास को हमने सफल बनाना है
बेहतर शिक्षा से बदले देश, ऐसे इसे अपनाए
शिक्षक, शिक्षा और शिक्षित
बस आगे बढते जाएँ
बस आगे बढते जाएँ
बस आगे बढते जाएँ.....

A Text Book on

INFORMATICS PRACTICES

CENTRAL BOARD OF SECONDARY EDUCATION

Shiksha Kendra, 2, Community Centre,
Preet Vihar, Delhi-110 092 India

A text book on Informatics Practices, Class XII.

PRICE : Rs 200/-

FIRST EDITION 2011 © CBSE, India

COPIES: 10,000

Reprint April 2013 (2,000 Copies)
Reprint April 2013 (4,000 Copies)
Reprint October 2013 (5,000 Copies)
Reprint March 2014 (3,000 Copies)
Reprint December 2014 (10,000 Copies)

PUBLISHED BY : The Secretary, Central Board of Secondary Education, Shiksha Kendra, 2, Community Centre, Preet Vihar, Delhi-110092

DESIGN, LAYOUT : Multi Graphics, 8A/101 WEA, Karol Bagh, New Delhi-110005, Phone : 25783846

PRINTED BY : A ONE OFFSET PRINTERS 5/34 Kirti Nagar Indl. Area,New Delhi-110015 Ph. : 25916255, 25916280

FOREWORD

Invention of Machines led to take up heavier, hazardous and heavy duty tasks beyond limits of human capacity. With the advent of computer, imagination got expanded networking worldwide. ICT (Information and Communication Technology) deals with information to compute / process and to communicate to other users. The ICT Policy in School education aims at augmenting creativity and promoting global competitiveness.

With this backdrop care is to be taken to impart ICT instructions so as to promote critical thinking, collaborative learning and multi discipline problem-solving skills.

The curriculum of "Informatics Practices" has been revised accordingly with focus on rapid application development using IDE, data processing and domain-specific applications, e.g. e-Governance, e-Business and e-Learning. Use of open source software is encouraged.

In its first attempt, the text book of Informatics Practices for Class XI has been published. The text book of Informatics Practices for Class XII is ready. I am happy to release the same. In the Syllabus of IP for Class XII a unit on Networking and Open Standards has been introduced.. RDBMS (Relational Database Management System) is dealt with using MYSQL in place of SQL & PL/SQL using oracle. A new unit on IT Applications has been added with three focused domains to enhance understanding the above tools and techniques to solve real life problems by designing both front and back end with proper data connectivity.

The text book includes puzzles in the beginning of the chapters to motivate students for collective and joyful learning of the subject. Cue to future trend may ignite curiosity.

My thanks are due to the text book development team aptly steered by Prof. Om Vikas.

Appreciation is also due to Mrs. Chitralekha Gurumuthy, former Director (Acad.) and (Late) Dr.(Smt.) Srijata Das, Education Officer, for Planning, coordinating and executing this initiative resulting into this publication.

I hope that all students and teachers will benefit from this publication. Their feedback will be highly appreciated for further improvement.

VINEET JOSHI
CHAIRMAN

भारत का संविधान

उद्देशिका

हम, भारत के लोग, भारत को एक सम्पूर्ण [1]प्रभुत्व-संपन्न समाजवादी पंथनिरपेक्ष लोकतंत्रात्मक गणराज्य बनाने के लिए, तथा उसके समस्त नागरिकों को:

सामाजिक, आर्थिक और राजनैतिक न्याय,
विचार, अभिव्यक्ति, विश्वास, धर्म
और उपासना की स्वतंत्रता,
प्रतिष्ठा और अवसर की समता
प्राप्त कराने के लिए
तथा उन सब में व्यक्ति की गरिमा
[2]और राष्ट्र की एकता और अखंडता
सुनिश्चित करने वाली बंधुता बढ़ाने के लिए

दृढ़संकल्प होकर अपनी इस संविधान सभा में आज तारीख 26 नवम्बर, 1949 ई॰ को एतद्द्वारा इस संविधान को अंगीकृत, अधिनियमित और आत्मार्पित करते हैं।

1. संविधान (बयालीसवां संशोधन) अधिनियम, 1976 की धारा 2 द्वारा (3.1.1977) से "प्रभुत्व-संपन्न लोकतंत्रात्मक गणराज्य" के स्थान पर प्रतिस्थापित।
2. संविधान (बयालीसवां संशोधन) अधिनियम, 1976 की धारा 2 द्वारा (3.1.1977) से "राष्ट्र की एकता" के स्थान पर प्रतिस्थापित।

भाग 4 क

मूल कर्त्तव्य

51 क. मूल कर्त्तव्य – भारत के प्रत्येक नागरिक का यह कर्त्तव्य होगा कि वह –

(क) संविधान का पालन करे और उसके आदर्शों, संस्थाओं, राष्ट्रध्वज और राष्ट्रगान का आदर करे;

(ख) स्वतंत्रता के लिए हमारे राष्ट्रीय आंदोलन को प्रेरित करने वाले उच्च आदर्शों को हृदय में संजोए रखे और उनका पालन करे;

(ग) भारत की प्रभुता, एकता और अखंडता की रक्षा करे और उसे अक्षुण्ण रखे;

(घ) देश की रक्षा करे और आह्वान किए जाने पर राष्ट्र की सेवा करे;

(ङ) भारत के सभी लोगों में समरसता और समान भ्रातृत्व की भावना का निर्माण करे जो धर्म, भाषा और प्रदेश या वर्ग पर आधारित सभी भेदभाव से परे हों, ऐसी प्रथाओं का त्याग करे जो स्त्रियों के सम्मान के विरुद्ध हैं;

(च) हमारी सामासिक संस्कृति की गौरवशाली परंपरा का महत्त्व समझे और उसका परिरक्षण करे;

(छ) प्राकृतिक पर्यावरण की जिसके अंतर्गत वन, झील, नदी, और वन्य जीव हैं, रक्षा करे और उसका संवर्धन करे तथा प्राणी मात्र के प्रति दयाभाव रखे;

(ज) वैज्ञानिक दृष्टिकोण, मानववाद और ज्ञानार्जन तथा सुधार की भावना का विकास करे;

(झ) सार्वजनिक संपत्ति को सुरक्षित रखे और हिंसा से दूर रहे;

(ञ) व्यक्तिगत और सामूहिक गतिविधियों के सभी क्षेत्रों में उत्कर्ष की ओर बढ़ने का सतत प्रयास करे जिससे राष्ट्र निरंतर बढ़ते हुए प्रयत्न और उपलब्धि की नई उंचाइयों को छू ले;

[1](ट) यदि माता-पिता या संरक्षक है, छह वर्ष से चौदह वर्ष तक की आयु वाले अपने, यथास्थिति, बालक या प्रतिपाल्य के लिये शिक्षा के अवसर प्रदान करे।

1. संविधान (छयासीवां संशोधन) अधिनियम, 2002 की धारा 4 द्वारा प्रतिस्थापित।

THE CONSTITUTION OF INDIA

PREAMBLE

WE, THE PEOPLE OF INDIA, having solemnly resolved to constitute India into a [1]**SOVEREIGN SOCIALIST SECULAR DEMOCRATIC REPUBLIC** and to secure to all its citizens :

JUSTICE, social, economic and political;

LIBERTY of thought, expression, belief, faith and worship;

EQUALITY of status and of opportunity; and to promote among them all

FRATERNITY assuring the dignity of the individual and the[2] unity and integrity of the Nation;

IN OUR CONSTITUENT ASSEMBLY this twenty-sixth day of November, 1949, do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

1. Subs, by the Constitution (Forty-Second Amendment) Act. 1976, sec. 2, for "Sovereign Democratic Republic" (w.e.f. 3.1.1977)
2. Subs, by the Constitution (Forty-Second Amendment) Act. 1976, sec. 2, for "unity of the Nation" (w.e.f. 3.1.1977)

THE CONSTITUTION OF INDIA

Chapter IV A

FUNDAMENTAL DUTIES

ARTICLE 51A

Fundamental Duties - It shall be the duty of every citizen of India-

(a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;

(b) to cherish and follow the noble ideals which inspired our national struggle for freedom;

(c) to uphold and protect the sovereignty, unity and integrity of India;

(d) to defend the country and render national service when called upon to do so;

(e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;

(f) to value and preserve the rich heritage of our composite culture;

(g) to protect and improve the natural environment including forests, lakes, rivers, wild life and to have compassion for living creatures;

(h) to develop the scientific temper, humanism and the spirit of inquiry and reform;

(i) to safeguard public property and to abjure violence;

(j) to strive towards excellence in all spheres of individual and collective activity so that the nation constantly rises to higher levels of endeavour and achievement;

[1](k) who is a parent or guardian to provide opportunities for education to his/her child or, as the case may be, ward between age of 6 and 14 years.

1. Subs. by the Constitution (Eighty - Sixth Amendment) Act, 2002

ACKNOWLEDGEMENTS

CBSE ADVISORS

- Shri Vineet Joshi, Chairman, CBSE
- Smt. Sadhna Parashar, Director (Academic, Research, Training & Innovation)

CONVENOR & EDITOR

- Prof. Om Vikas, Former Director, IIITM, Gwalior

DEVELOPMENT TEAM

- Mukesh Kumar, DPS, RK Puram, Delhi
- Mrs. Nancy Sehgal, Mata Jai Kaur Public School, Ashok Vihar, Delhi
- Mrs. Gurpreet Kaur, GD Goenka School, Vasant Kunj, Delhi
- Yogesh Kumar, Mira Model School, Janak Puri, New Delhi
- Mrs. Ritu Arora, DPS, Gurgaon
- Mrs. Divya Jain, Apeejay, Noida

MEMBER COORDINATOR

- (Late) Dr. Srijata Das
- Ms. Kshipra Verma, Education Officer, CBSE, Delhi

CONTENTS

INFORMATICS PRACTICES


CBSE
CONTINUOUS AND COMPREHENSIVE EVALUATION
KNOW - AS YOU GROW

COMPUTER NETWORKING

Learning objectives

After learning this chapter the students will be able to:

- Define what a Computer Network is
- List the benefits of networking
- List different wired and wireless media for communication
- Identify different network devices
- Identify the type of network on the basis of area covered
- Describe various terms associated with computer networks.
- List various security threats to computer networks
- List the preventive and corrective measures against these threats

Have you ever worked on Internet? Have you ever used an ATM? Have you ever given the print command on a computer to get the printout on a printer attached to some other computer? Have you ever transferred songs from your computer to a cell phone or vice-versa? If the answer to any of these questions is YES, then you have experienced and utilized the services of a computer network. In this chapter you will study about various types of computer networks, their benefits, and what is required to create computer networks.

Puzzle [1]

Can you place 10 coins in such a way that they lie in 5 straight lines and on each line there are exactly 4 coins?

Such situations are faced by the designers of computer networks. They have to think of various ways of interconnecting the computers so that the specific goals of network setup are achieved.

Networking - a brief overview

A computer network is a collection of interconnected computers and other devices which are able to communicate with each other. In this context, the term interconnected means that there exists a path through which data can be transmitted from one computer/device to another.

> A **computer network** is a collection of interconnected computers and other devices which are able to communicate with each other and share hardware and software resources.

Why do we have computer networks? Are there any advantages of networked computers over stand alone computers? Yes, there are many. A few of these advantages are:

- Resource Sharing
- Cost saving
- Collaborative user interaction
- Time saving
- Increased storage

Let us discuss these advantages in some details.

Resource Sharing:

In a networked computer lab when print command is given on one computer the document may be printed by the printer which is attached to some other computer. This printer is able to print documents from multiple users on the network. It means that the printer is being shared by more than one users. This is an example of resource sharing, and this is possible only when the computers are interconnected to form a network. Similarly other resources like Hard Disk, DVD Drive, and Scanner etc. can also be shared on a computer network. Software resources like Application Softwares, Anti-Virus tools etc. can also be shared on computer networks. You can very easily conclude that this resource sharing also leads to cost-saving.

Collaborative User Interaction:

Let us take one more example of a school. Here we assume that all the computers in the school are connected to one main computer (called server). After the exams, teachers have to exchange marks with each other for result preparation. So all the teachers enter the marks class wise for their respective subjects at one centralized location and from there the class teachers can copy the marks of their classes and can prepare the result. This way data is shared on a computer network. Similar is the case with other computerized organizations also.

Let us take another example of collaborative user interaction. If we are working on a computer which is a part of a computer network, we can communicate with any other user of the network through e-mail or chatting. It takes negligible time to send and receive messages and watch live videos of one another irrespective of terrestrial distances. If the e-mail or chatting is done for some useful purpose, it leads to **increased productivity, cost-saving** as well as **time-saving.**

Increased Storage:

On a network, same data may be replicated on multiple computers to ensure the availability of data in the case of some computer getting faulty. For example, when you save your java applications on your computer, you can also store their copies on some other networked computers in your lab. This way your work will be available even if your computer develops some fault or somehow your programs are deleted from your computer. Similarly, on large networks also the data is replicated on multiple computers as if a huge storage area is available to store multiple copies of the data.

All these advantages are there for a small organization like a school as well as for big business organizations and for governments. Today, small as well as big organizations, and governments keep their data on secured large scale computers called servers. They share this data with authorized users. This ensures security. Customer care cells of companies share the resources and data and they also communicate among themselves as well as with customers with the help of computer networks only.

Networking Hardware

To form a computer network a lot of hardware devices are required. Some of these devices along with their functionalities are mentioned below :

Transmission Media

Computers on a network are able to share data and other resources. They are also able to communicate among themselves. To make all this possible there must be some medium over which the data can travel from one computer to another. A medium of data transmission over a computer network is called a channel or a transmission medium. Channels may be guided (wired) or unguided (wireless).

> A transmission medium is a medium of data transfer over a network. It can be wired or wireless.

Wired Media

A number of various types of cables are used to transfer data over computer networks. These are Twisted Pair Cable, Co-axial Cable, and Optical Fiber Cable. Let us know about these in some details.

Twisted Pair Cable - This is probably the most widely used cable for creating small computer networks. It contains four twisted pairs covered in an outer shield. These pairs are colour coded. An RJ-45 connecter is used to connect this cable to a computer. It is of two types:

UTP (Unshielded Twisted Pair): As the name suggests in UTP cables individual pairs are not shielded.

UTP Cable

UTP Cable
With RJ-45 Connector

Characteristics of UTP cable:

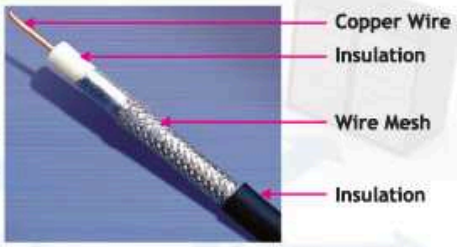
- It is a low-cost cable available for setting up small networks.
- It is a thin (External diameter app. 0.43cm) and flexible cable and therefore it offers ease of installation.
- It can carry data upto a length of 100m at a stretch.

STP (Shielded Twisted pair): It is the same cable as the UTP, but with each pair shielded individually. An outer shield then covers all the pairs like in UTP. STP data connectors are used to connect STP cable to the computer. RJ-45 connectors can also be used to connect this cable to a computer.

Characteristics of STP cable:

- As compared to UTP, STP offers better immunity against internal and external electromagnetic interferences.
- It is expensive than UTP cable.
- As compared to UTP cable, STP cable is difficult to install.

Co-axial cable (or coax) - A coaxial cable consists of two conductors that share a common axis. The inner conductor is a straight wire and the outer conductor is a shield that might be braided or a foil.

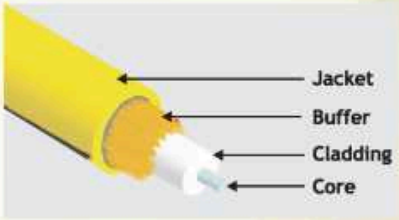


Characteristics of Co-axial cable:

- It can carry data for a larger distance (185m - 500m) at a stretch.
- Less susceptible to electromagnetic fields
- Bulkier and less flexible than twisted pair.
- Due to its thickness (1cm diameter) and less flexibility, it is difficult to install as compared to twisted pair cable.

Earlier coaxial cable was also used for connecting computers in small networks but now UTP/STP cables are more commonly used for this purpose.

Optical Fiber cable - Optical Fibers are long, thin strands of glass about the thickness of a human hair. They are arranged in bundles called optical fiber cables and used to transmit data through light signals over long distances.



An optical fiber has following parts:

- **Core** - It is the thin glass rod at the center through which the light travels
- **Cladding** - It is the outer optical material surrounding the core that reflects the light back into the core
- **Buffer coating** - It is the plastic coating that protects the cable from damage and moisture

These optical fibers are arranged in bundles of hundreds and thousands and are protected by the cable's outer covering, called jacket.

Characteristics of Optical Fiber Cable:

- It can carry data for a very large distance at a stretch.
- Not susceptible to electromagnetic fields
- Specially skilled people are required to install optical fiber cables.
- Till date it is the most expensive and at the same time the most efficient cable available for computer networks.

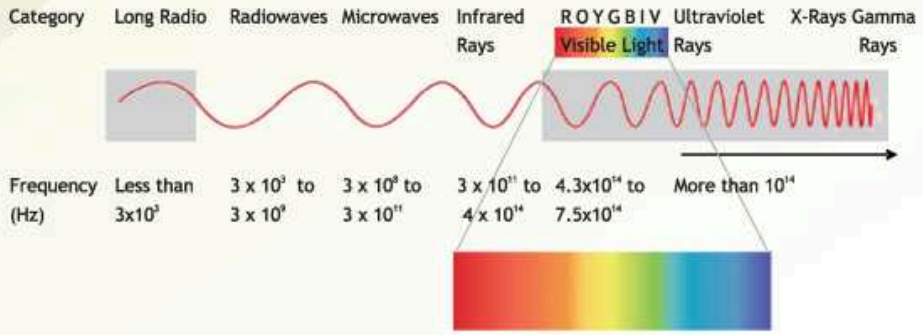
Comparison of wired media

Cable → / Paramater ↓	Twisted Pair Cable	Coaxial Cable	Optical Fiber Cable
Data Transfer Rate	10Mbps-10Gbps	100 Mbps	More than100 Gbps
Data Transfer Range	100 m	185m-500 m	-
Interference Susceptibility	More	Less than Ethernet cable	NIL
Cost of Cable	Least Cost	More than Ethernet	Very Expensive

Wireless Media

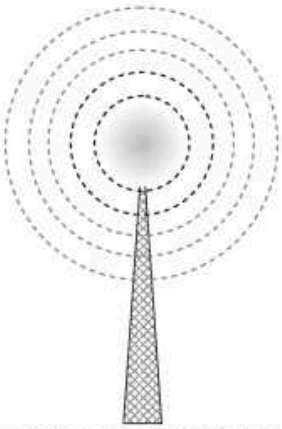
Electromagnetic waves are used for wireless communication over computer networks. Frequencies of waves are measured in Hertz (Hz). As the frequencies of electromagnetic waves change, their properties also change. Based on their frequencies, electromagnetic waves are categorized into various categories. These categories are (in increasing order of frequencies): radio waves, microwaves, infrared radiation, visible light, ultraviolet radiation, X-rays, and Gamma rays.

Electromagnetic Spectrum



Out of these only radio waves, microwaves, and infrared rays are used for wireless communication. Let us now study about these waves in some details.

Radio Waves - Radio waves have a frequency range of 3 KHz to 3GHz. Radio waves are used for communication over distances ranging from a few meters (in walkie-talkies) upto covering an entire city. These waves are easy to generate, can travel long distances and can penetrate buildings easily. That's why they are widely used for communication, both indoors and outdoors. Cordless phones, AM and FM radio broadcast, Garage door openers etc. are examples of radio wave transmission.



Omni Directional Radio Waves

Characteristics of Radio Wave Transmission:

- These waves are omni-directional, so the transmitting and receiving antennas need not be aligned.

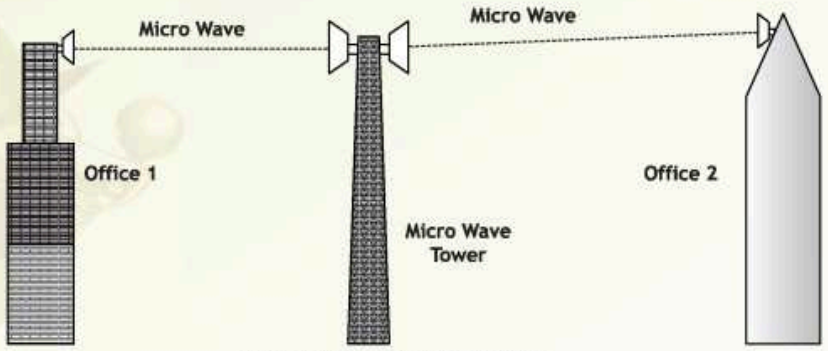
  (Recall when you throw a stone in a pond, circular waves are generated and spread outwards. Similarly, radio waves are generated by the transmitter and spread in all the directions.)

- Relatively inexpensive than wired media.
- It offers ease of communication over difficult terrain
- The transmission can be interfered by motors or other electrical equipment
- Permission from concerned authorities is required for use of radio wave transmission
- Less secure mode of transmission

Micro Waves - Micro waves have a frequency range of 300MHz (0.3 GHz) to 300 GHz. This range has some overlapping portion (0.3GHz - 3GHz) with radio waves as there is no clear-cut demarcation between radio waves and micro waves. Microwaves travel in straight lines and cannot penetrate any solid object. Therefore for long distance microwave communication, high towers are built and microwave antennas are put on their tops. Distance between two microwave towers depends on many factors including frequency of the waves being used and heights of the towers. These waves travel in straight lines and therefore the sending and receiving antennas have to be aligned with each other.

An example of usage of microwaves for communication is as follows:

In the big cities where land is very costly and a lot of formalities have to be completed to get permission to dig land for cabling, microwave antennas can be put on top of high rise buildings and communication can be started in a short time.



Micro Wave Communication

Characteristics of Micro Wave Transmission:

- Free from land acquisition rights
- Relatively inexpensive than wired media
- Offers ease of communication over difficult terrain
- The transmission is in straight lines so the transmitting and receiving antennas need to be properly aligned (line of sight transmission)

Infrared Waves - Infrared waves have a frequency range of 300 GHz to 400 THz. If you recall VIBGYOR spectrum of light, you will also recall that red light has the lowest frequency (400THz - 484THz) in this spectrum. Infrared waves are so called because they have a frequency range of just less than that of red light. These waves are used for short range communication (approx. 5m) in a variety of wireless communications, monitoring, and control applications. Home-entertainment remote-control devices, Cordless mouse, and Intrusion detectors are some of the devices that utilize infrared communication. These waves are easy to build but have a major drawback- they do not pass through solid objects (try standing between your remote control and your television and see if it still works). On the other hand, these waves do not pass through solid walls is a plus point also. Because of this, infrared system in one room of a building will not interfere with a similar system in adjacent rooms (you cannot control TV in another room with the remote in your hand in a room).

Characteristics of Infrared Wave Transmission:

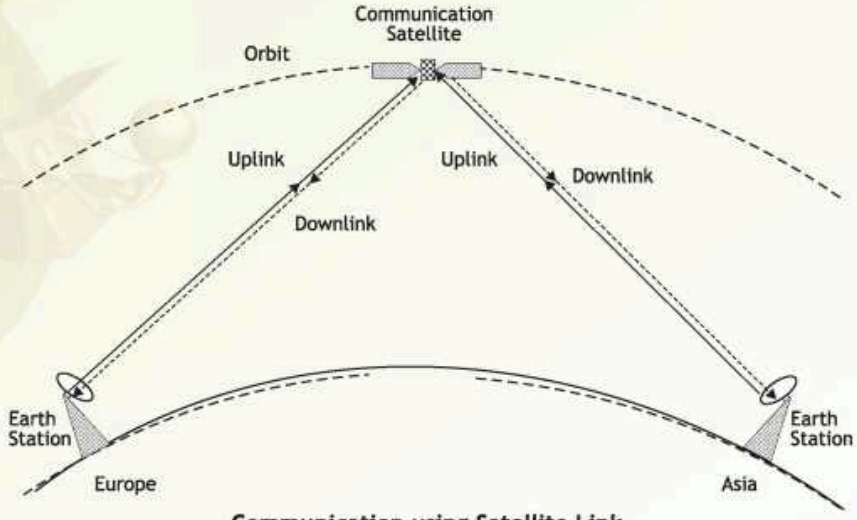
- It is a line of sight transmission; therefore information passed to one device is not leaked to another device.
- No government license is required for their use
- It is a line of sight transmission, therefore at a time only two devices can communicate.
- The waves do not cross any solid object in between
- Performance drops with longer distances

Bluetooth - Bluetooth technology uses radio waves in the frequency range of 2.402 GHz to 2.480 GHz. This technology is used for short range communication (approx. 10m) in a variety of devices for wireless communication. Baby monitors, door openers, and cell phones are some of the devices that utilize Bluetooth communication.

Characteristics of Bluetooth Transmission:

- Line of sight between communicating devices is not required. (Think Why?)
- Bluetooth can connect upto eight devices simultaneously.
- Slow data transfer rate (upto 1Mbps).

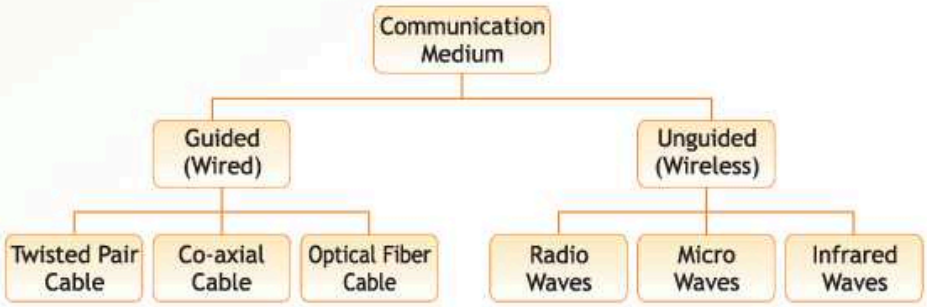
Satellite Link - Satellite links are used for very long distance wireless communication which may range from intercity to intercontinental. Transmission from the earth to a satellite is known as **uplink**. Transmission from a satellite to the earth is known as **downlink**. There are multiple micro wave frequency bands which are used for satellites links. Frequency used for uplink varies from 1.6 GHz to 30.0 GHz and that for downlink varies from 1.5GHz to 20.0GHz. Downlink frequency is always lower than the uplink frequency. For example, the uplink frequency is 6.0GHz, and the corresponding downlink frequency is 4.0 GHz. A communications satellite is a relay station in orbit above the earth that receives, regenerates, and redirects signals carried on a specific frequency. The satellite system is very expensive but its area coverage and fringe benefits compensate for the expenses. Communication satellites are normally owned by governments or by government approved organizations of various countries



Communication using Satellite Link

Characteristics of Transmission using satellite link:

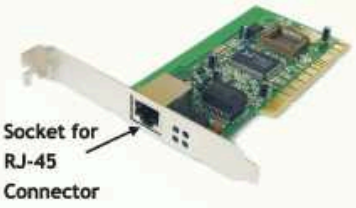
- Satellites cover large area of earth
- Since communication over very long distances is possible, this becomes a commercially attractive option.
- This system is expensive
- Requires legal permissions.



Network Devices

Other than the transmission media many other devices are required to form computer networks. Some of these devices are:

NIC: Any computer which has to be a part of a computer network must have an NIC (Network Interface Card / Unit) installed in it. A computer communicates with other computers on a network with the help of an NIC only. Now a days, in most of the PCs and the laptops, NIC is an integral part of the motherboard.



A Network Interface Card

*An **NIC** (Network Interface Card) is a device that enables a computer to connect to a network and communicate.*

Hub: Hub is a device that allows us to connect multiple computers/devices together in a network. A hub has ports into which the cables from individual computers' NICs are inserted. This way each computer's NIC is connected to hub and hence all the computers are connected together. Whenever a computer has to send some information to some other computer(s), the information is sent by the NIC to the hub. Then the hub re-transmits this information to the other computers attached to it. The computer(s) for which the information is intended receive(s) this information and accept(s) it. Other computers on the network simply reject this information.

Hub/Switch

*A **Hub** is an electronic device that connects several nodes to form a network and redirects the received information to all the connected nodes in broadcast mode.*

Switch: A switch is an intelligent hub. It looks exactly like a hub. It has the same function as that of a hub: to connect multiple computers/devices in a network. But the difference between the two is in the way they re-transmit the received information. Unlike a hub, instead of broadcasting (sending to each device attached to it) the received information, a switch sends the information selectively only to those computers for which it is intended. This makes a switch more efficient than a hub.

*A **Switch** is an intelligent device that connects several nodes to form a network and redirects the received information only to the intended node(s).*

Repeater: When the data is transmitted over a network for long distances, the data signal gets weak after certain distance. This distance depends on the data transfer range of transmission channel being used and can be from a few meters to a few kilometers. If the signal becomes weak, it cannot reach its destination. Therefore, some device is required which can re-strengthen the data signal before it gets too weak. Repeater is such a device. A repeater regenerates the received signal and re-transmits it to its destination

*A **Repeater** is a device that is used to regenerate a signal which is on its way through a communication channel. A repeater regenerates the received signal and re-transmits it to its destination.*

Gateway: There are a large number of computer networks in this world. As common examples you can consider your school's computer network, ATM network of a bank, a big company's computer network spread over a city, etc. There are thousands of computer networks that exist. These networks use different hardware and software. Many times these networks need to communicate with each other. For example, companies X, Y, and Z do business with each other and therefore they want to interconnect their computer networks. Another example is the internet which contains a large number of different types of networks spread over the globe. Different networks are sometimes incompatible

with each other. It is like a group of persons using different languages for conversation. When two or more networks using different hardware and software have to be connected, some device is needed which can translate one network's language into the other's. A gateway is a device, which is used to connect different types of networks. A gateway is capable of understanding address architectures used in different networks and seamlessly translate between these address architectures.

*A **Gateway** is a device, which is used to connect different types of networks and perform the necessary translation so that the connected networks can communicate properly.*

Network Topologies

Before we start discussion on network topologies, let us understand the term 'NODE'.

Any device (Computer, Scanner, Printer, etc.) which is directly connected to a computer network is called a node. Suppose you are working on a PC at your home and then you connect it to internet. As soon as it becomes a part of internet (which is a computer network), it becomes a node. Similarly, in your school, all the computers which are linked to school's computer network are nodes.

*A **Node** is a device, which is directly connected to a computer network. It can be a computer or any other device like printer, scanner etc.*

Once we know about different communication media and devices to form a computer network, we can procure these media and devices and start constructing a computer network. Suppose we have 10 computers and we want to interconnect them to form a network. How can we interconnect them?

Recall the puzzle given in the beginning of this lesson. Is that not similar to the problem of connecting nodes in a network?

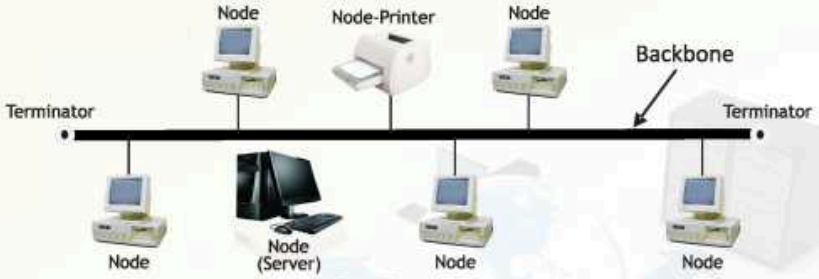
Similarly, when we have to connect computers/devices in a network, there may be certain conditions which have to be satisfied. Depending upon these conditions, there may be different ways of interconnecting the computers/devices. The way in which the

computers/devices are physically interconnected to form a network is called a Topology.

> *A **Topology** is an arrangement of physical connections among nodes in a network.*

There exist different network topologies. Let us discuss a few of them.

Bus Topology: In bus topology all the nodes are connected to a main cable called backbone. If any node has to send some information to any other node, it sends the signal to the backbone. The signal travels through the entire length of the backbone and is received by the node for which it is intended. A small device called terminator is attached at each end of the backbone. When the signal reaches the end of backbone, it is absorbed by the terminator and the backbone gets free to carry another signal. This prevents the reflection of signal back on the cable and hence eliminates the chances of signal interference.

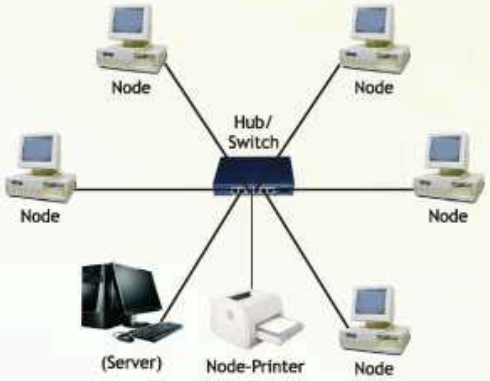


Bus Topology

Characteristics of Bus topology:

- It is easy to install.
- It requires less cable length and hence it is cost effective.
- Failure of a node does not affect the network.
- In case of cable (backbone) or terminator fault, the entire network breaks down.
- Fault diagnosis is difficult.
- At a time only one node can transmit data.

Star Topology: In star topology each node is directly connected to a hub/switch. If any node has to send some information to any other node, it sends the signal to the hub/switch. This signal is then broadcast (in case of a hub) to all the nodes but is accepted by the intended node(s). In the case of a switch the signal is sent only to the intended node(s).



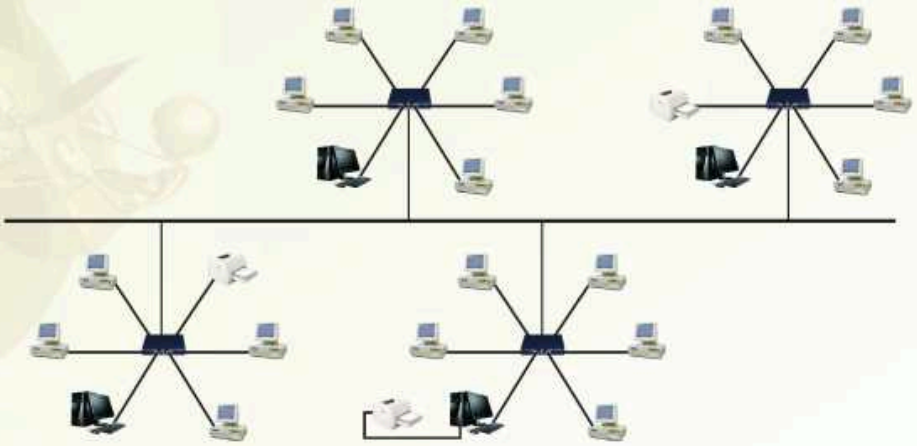
Bus Topology

Star topology generally requires more cable than bus topology.

Characteristics of Star topology:

- It is more efficient topology as compared to bus topology.
- It is easy to install
- It is easy to diagnose the fault in Star topology.
- It is easy to expand depending on the specifications of central hub/switch
- Failure of hub/switch leads to failure of entire network
- It requires more cable length as compared to bus topology.

Tree Topology: Tree topology is a combination of bus and star topologies. It is used to combine multiple star topology networks. All the stars are connected together like a bus. This bus-star hybrid approach supports future expandability of the network



Tree Topology

Characteristics of Tree topology:

- It offers easy way of network expansion
- Even if one network (star) fails, the other networks remain connected and working.

Network Protocols

A computer network can be set up after procuring all the required hardware. But the network devices will be able to communicate with each other only after they know the rules of communication. Think of a group discussion session where all the participants can speak. Each participant can speak a lot. But the group discussion can be started and can be conducted well if each participant knows the rules of discussion. If all the participants start speaking suddenly and suddenly they stop, or a person stands up randomly and starts interrupting the discussion, or any such thing happens, the group discussion cannot be conducted at all. A more complex situation exists in the case of computer networks where the participating devices have no common sense and they follow each defined rule to the T. So it is essential that the rules of communication are very well defined. A set of rules is also known as a protocol.

A **network protocol** is a set of rules for communication among networked devices. Protocols generally includes rules of how and when a device can send or receive the data, how is the sent data packaged, and how it reaches its destination.

There are a number of protocols defined for computer networks. Here we discuss three of them - HTTP, TCP/IP, PPP.

HTTP (Hyper Text Transfer Protocol): HTTP is used to transfer all files and other data (collectively called resources) from one computer to another on the world wide web. When an HTTP client (a browser) sends a request to an HTTP server (web server) , the server sends responses back to the client. This transfer of requests and responses is done following HTTP protocol.

TCP/IP (Transmission Control Protocol / Internet Protocol): It is the basic protocol of the Internet. Communication between two computers on internet is done using TCP/IP protocol. TCP/IP can also be used as a communications protocol in a private network. TCP/IP is a two-layer protocol. When data is to be sent from one computer to another over internet, it is first broken into smaller packets which are actually sent. When these packets are received by the receiver computer, they are assembled into the original message. This job of dividing the original message into packets and re-assembling the received packets into the original message is done following TCP protocol. Internet protocol is followed to ensure that each of these packets gets to the right destination. Different packets from the same message may be routed differently, but they reach the same destination and are reassembled there.

PPP (Point to Point Protocol): It is a protocol for direct communication between two computers, typically a personal computer connected by phone line to a server. Most Internet service providers (ISPs) use PPP for customer dial-up access to the Internet. PPP is used over many types of physical networks including cellular telephone, serial cable, phone line, trunk line, specialized radio links, and fiber optic links.

Know More!

There are a lot of other communication protocols like SMTP, POP, UDP etc. You can explore the net to find more about these protocols.

Types of Networks

A computer network may span any amount of geographical area. It can be on a table, in a room, in a building, in a city, in a country, across continents or around the world. On the basis of area covered computer networks are classified as:

- PAN - Personal Area Network
- LAN - Local Area Network
- MAN - Metropolitan Area Network
- WAN -Wide Area Network

PAN (Personal Area Network): A PAN is a network of Communicating devices (Computer, Phone, MP3/MP4 Player, Camera etc.) in the proximity of an individual. It can cover an area of a few meters radius.

A PAN

When you have to transfer songs from one cell phone to another, you set up a PAN of two phones. When files are transferred from a PC to an MP3 player, a PAN is set up between the two. There can also be multiple devices in PAN. A PAN can be set up using guided media (USB cable) or unguided media (Bluetooth, Infrared).

LAN (Local Area Network): A LAN is a network of computing/Communicating devices in a room, building, or campus. It can cover an area of a few meters to a few kilometers radius. A networked office building, school, or home usually contains a single LAN, though sometimes one building can contain a few small LANs (Like some schools have independent LANs in each computer lab.). Occasionally a LAN can span a group of nearby buildings.

Wireless LAN

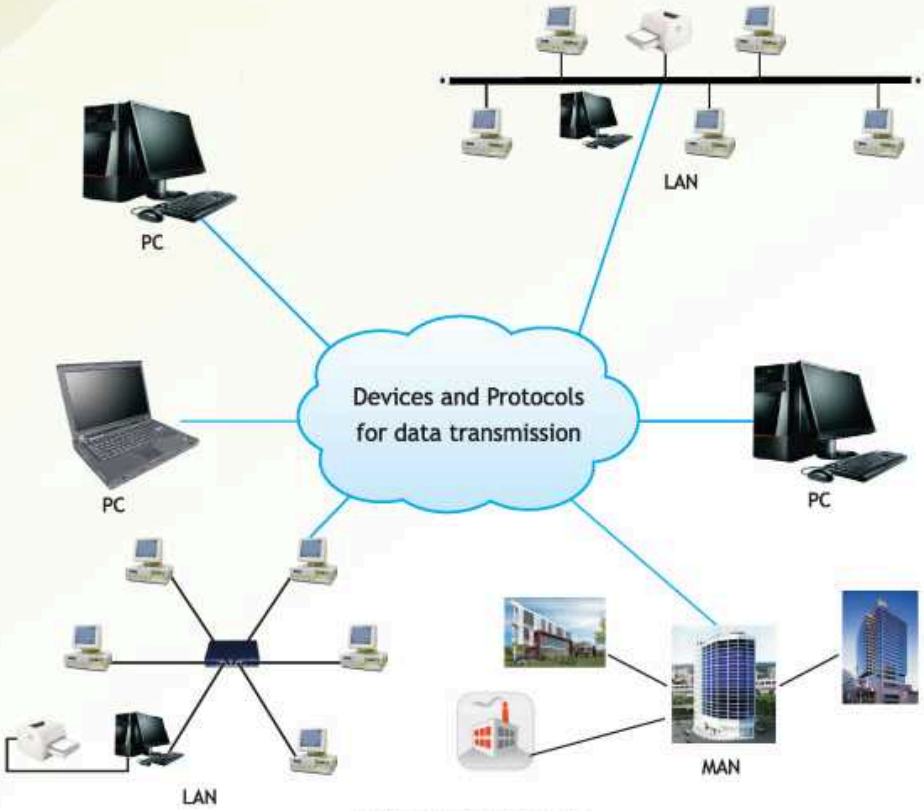
In addition to operating in a limited space, a LAN is owned, controlled, and managed by a single person or organization.

A LAN can be set up using wired media (UTP cables, Co-axial cables etc.) or wireless media (Infrared, radio waves). If a LAN is setup using unguided media, it is known as WLAN (wireless LAN).

MAN (Metropolitan Area Network): A MAN is a network of computing/communicating devices within a city. It can cover an area of a few kilometers to a few hundred kilometers radius. A network of schools, or banks, or Government offices etc., within a city, are examples of MANs. A MAN is usually formed by interconnecting a number of LANs and individual computers. All types of communication media (guided and unguided) are used to set up a MAN. A MAN is typically owned and operated by a single entity such as a government body or a large corporation. A good example of a MAN is the interconnected offices of a state government.

A Metropolitan Area Network

WAN (Wide Area Network): A WAN is a network of computing/communicating devices crossing the limits of a city, country, or continent. It can cover an area of over hundreds of kilometer radius. A network of ATMs, BANKs, National Government Offices, International Organizations' Offices etc., spread over a country, continent, or covering many continents are examples of WANs. WANs usually contain a number of interconnected individual computers, LANs, MANs, and maybe other WANs. All types of communication media (guided and unguided) are used to set up a WAN. The best known example of a WAN is the internet.



A Wide Area Network

The following table summarizes the characteristics of PANs, LANs, MANs, and WANs.

Network → / Parameter ↓	PAN	LAN	MAN	WAN
Area Covered	Small Area (Upto 10m radius)	A few meters to a few kilometers (Upto 10Km radius)	A city and its vicinity (Upto 100Km radius)	Entire country, continent, or globe (No upper limit)
Error Rates	Lowest	Lowest	Moderate	Highest
Transmission Speed	High Speed	High Speed	Moderate Speed	Low speed
Networking Cost	Negligible	Inexpensive	moderately expensive equipment	Expensive

Identification of computers and users over a network

Once a network has been set up, the nodes can communicate among themselves. But for proper communication, the nodes should be uniquely identifiable. If a node X sends some information for node Y on a network, then it is mandatory that nodes X and Y are uniquely identifiable on the network. Let us see how this is achieved.

MAC (Media Access Control) address: Each NIC has a universally unique address assigned to it by its manufacturer. This address is known as the MAC (Media Access Control) address of the card. It means that a machine with an NIC can be identified uniquely through its NIC's MAC address. MAC address of an NIC is permanent and does never change.

MAC addresses are 12-digit hexadecimal (or 48 bit) numbers. By convention, MAC addresses are usually written in one of the following two formats:

MM:MM:MM:SS:SS:SS

MM-MM-MM-SS-SS-SS

The first half (MM:MM:MM) of a MAC address contains the ID number of the adapter manufacturer. The second half (SS:SS:SS) of a MAC address represents the serial number assigned to the adapter (NIC) by its manufacturer.

For example, in the following MAC address,

00:A0:C9 : 14:C8:35

The prefix 00:A0:C9 indicates that the manufacturer is Intel Corporation. And the last three numbers 14:C8:35 are given by the manufacturer (Intel in this example) to this NIC.

*A **MAC (Media Access Control)** address is a unique 12 digit (6 digits for manufacturer code and 6 digits for serial number) hexadecimal number assigned to each NIC. MAC address of an NIC never changes.*

IP Address: Every machine in a network has another unique identifying number, called its IP Address. An IP address is a group of four bytes (or 32 bits) each of which can be a number from 0 to 255. A typical IP address looks like this:

59.177.134.72

To make it easier for us to remember, IP addresses are normally expressed in decimal format as a "dotted decimal number" like the one above.

On a network, IP address of a machine, and not the MAC address of its NIC, is used to identify it. Do you recall IP protocol? IP protocol identifies a machine with its IP address to route the packets.

MAC address is used only when a specific machine is to be targeted. For example, suppose we want to block a specific PC to access some network resource. If we use the PCs IP address, then the PC is not blocked permanently as its IP address may change when it connects to the network next time. Instead, if the PCs MAC address is used for the purpose, then the job is done!

*An **IP (Internet Protocol)** address is a unique 4 digit hexadecimal number assigned to each node on a network. IP address settings of a node can be changed by the user*

Know More!

There are two versions of IP addresses: version 4 (IPv4) and version 6 (IPv6). IPv6 uses 128 bits (IPv4 uses 32 bits) for an IP address. Using IPv4 only 2^{32} (approximately 4 billion) distinct devices can be addressed. In the current scenario and in the near future, this is a small fraction of the number of objects which need to be allocated IP addresses. As the human population and also the number of devices which need to be online (mobile phones, home appliances, personal communication devices, etc.) is increasing at a very fast pace, IPv4 addresses are being exhausted. To address this problem IPv6 was developed and it is now being deployed. Using IPv6 2^{128} (approximately 4 Bn x 4 Bn x 4 Bn x 4 Bn) distinct devices can be addressed. IPv6 also includes an important feature: a set of possible migration and transition plans from IPv4. There are many other features of IPv6. You can explore the internet and find more information about IPv6.

IP Address Vs MAC Address: You might wonder how an IP address differs from a MAC address. In fact, the IP address is assigned by the network administrator or the internet service provider while the MAC address is assigned by the manufacturer. Thus if a computer is transferred from one network to another, its IP address gets changed where as the MAC address remains the same.

From the IP address it is usually possible to track the tentative location of the computer but this is not the case with a MAC address.

Domain Name: So, whenever we have to communicate with a computer on internet, we can do so by using its IP address. But it is practically impossible for a person to remember the IP addresses of all the computers one may have to communicate with. Therefore, a system has been developed which assigns names to some computers (web servers) and maintains a database of these names and corresponding IP addresses. These names are called Domain Names. Examples of some domain names are cbse.nic.in, sikkimipr.org, indianrailway.gov.in etc. Domain names are used in URLs to identify particular Web servers. For example, in the URL *http://www.cbse.nic.in/welcome.htm*, the domain name is *www.cbse.nic.in*.

A domain name usually has more than one parts: top level domain name or primary domain name and sub-domain name(s). For example, in the domain name ***cbse.nic.in***, ***in*** is the primary domain name; ***nic*** is the sub-domain of ***in***; ***cbse*** is the sub-domain of ***nic***.

There are only a limited number of top level domains, and these are divided into two categories: Generic Domain Names and Country-Specific Domain Names. For example:

Generic Domain Names:

- **com** - commercial business
- **edu** - Educational institutions
- **gov** - Government agencies
- **mil** - Military
- **net** - Network organizations
- **org** - Organizations (nonprofit)

Country Specific Domain Names:

- **.in** - India
- **au** - Australia
- **ca** - Canada
- **.ch** - China
- **.nz** - New Zealand
- **.pk** - Pakistan
- **.jp** - Japan
- **.us** - United States of America

*In context of internet, a **Domain Name** is a name assigned to a server through Domain Name System (DNS). A domain name usually has more than one parts: top level domain name or primary domain name and sub-domain name(s).*

Domain Name Resolution: Domain Name Resolution is the process of getting the corresponding IP address from a domain name. It happens as follows:

Suppose you mention a URL in the web-browser to visit a website. The browser first checks your computer to find if the IP address of the server corresponding to the Domain Name (embedded in the URL) is present. If this address is present then with the help of this address, the corresponding server is contacted and then the website opens in your browser. Otherwise the browser sends this domain name to some specific servers (called domain name servers) to find the corresponding IP address. Once the IP address is known, the server is contacted and then the website opens in your browser.

***Domain Name Resolution** is the process of getting corresponding IP address from a domain name.*

Know More!

The Internet Corporation for Assigned Names and Numbers (ICANN) is an internationally organized, non-profit corporation that has responsibility for Internet Protocol (IP) address space allocation, protocol identifier assignment, generic (gTLD) and country code (ccTLD) Top-Level Domain name system management, and root server system management functions. These services were originally performed under U.S. Government contract by the Internet Assigned Numbers Authority (IANA) and other entities. ICANN now performs the IANA function.

Network Security

Computer networks are communication highways on which the data travels. Data travels on a network when an e-mail is sent or received, any transaction is made using a credit or a debit card, a web site is accessed, chatting is done, or any other work is done on a network. Data travelling over a network is vulnerable to attacks and thefts. There are some people who want to illegally access this data for any reason: using a credit/debit card data for shopping, using a company's data to sell it to some rival company, to find the kind of web sites a person accesses, or may be hundreds of other reasons. These people

may sometimes also want to destroy the data on its way, or block the services of some resources or sites. As the data goes from one point to another point on the Internet, for example, it may pass through several points along the way, giving other users the opportunity to access, and even alter it. Even other users on your system may maliciously transform your data. Unauthorized access to your system may be obtained by intruders, who then use advanced knowledge to impersonate you, steal information or even deny your access to your own resources.

Therefore, there is always a threat of some kind of attacks on computer networks' security. Some kinds of attacks on network security are as follows:

Denial of service attacks: A Denial of Service (DoS) attack is an attempt to make one or more network resources unavailable to their legitimate users. Examples of such attacks are:

- **Denial of Access to Information:** Corrupting, Encrypting, or changing the status of information so that it is not accessible to its legitimate user.
- **Denial of Access to Application:** Forced shutting of an application as soon as the user opens it.
- **Denial of Access to Resources:** Blocking a resource, may be a printer or scanner or USB port, of a computer from proper working.
- **Denial of Access to a Website:** Continuously sending bulk requests to a website so that it is not available to any other user.

Intrusion Problems: An Intrusion problem is an attempt to mischievously steal some information from someone's computer. Examples of Intrusion are:

Snooping - Have you ever tried to read someone else's slam book secretly? This is snooping. Have you ever tried to read what someone else is writing in his/her letter/email to someone else? This is snooping. Do you think your parents try to secretly check your mobile phone to find its contents? If yes, then what your parents are doing is snooping.

In context of network security, snooping refers to gaining unauthorised access to another person's or organization's data. This may be done in a number of ways:

- By getting someone's login information by casually watching what he/she is typing.
- Reading the files on someone's computer in an unauthorised manner
- Using some softwares which keeps track of the activities and data being sent or received on someone's computer.

> ***Snooping*** *refers to gaining unauthorised access to another person's or organization's data.*

Eavesdropping - Do you ever find that when you are talking to someone else, another person is secretly trying to listen to your talks? What that person is doing is 'eavesdropping'. Have you ever tried to secretly listen to the conversation between two teachers regarding your class? If yes, then what you have done is 'eavesdropping'.

In context of network security Eavesdropping refers to unauthorised access to another person's or organization's data while the data is on its way on the network.

This may be done in a number of ways:

- By setting up parallel telephone lines.
- By installing some software (spyware) in the target computer.
- By installing some receiver which captures the data while on its way.

> ***Eavesdropping*** *refers to gaining unauthorised access to another person's or organization's data while the data is on its way on the network*

It is the responsibility of a person/organization to protect its network from such attacks. The term **Network Security** refers to all activities undertaken to protect a computer network from attacks to its security. This covers protection of computers in the premises and data on the network from both internal and external attacks. To protect a network from security attacks, a number of steps are taken. These steps include:

Login-Password: By assigning login names and strong passwords to the users of a system, it can be ensured that only authorized people access a computer. This helps in increasing the computer as well as network security. A strong password is the one which is easy to

remember for the user but difficult (almost impossible) for the others to guess successfully. On the other hand a weak password is generally the name of a person or organization, the registration number of a vehicle, year of birth of a person and so on. A weak password can be cracked in a few attempts. Examples of strong passwords may be raavanisdead (raavan is dead - with spaces removed), 2aur2paanch (name of an old Hindi movie) or anything like that.

Firewall: A firewall is a hardware device or a software that is installed to monitor the data entering the computer/Network or leaving it. A firewall permits only that data to enter or leave a computer/Network for which permissions have been granted by the computer/network administrator.

Anti Virus Software: These days anti-virus softwares work against not only the virus but also against almost all kinds of malware. Therefore by installing a full version legal (not the pirated one or freeware) anti-virus software network security can be increased.

File Permissions: A data file on a computer can be created, read, or modified. An application file can be run or executed. Accordingly, different rights (privileges) can be given to different users of a computer to perform one or more of these tasks. For example, suppose there is a data file containing students' marks in a school. The system administrator has created that file. If there is a need to create another such file, then again only the system administrator can create it. So, system administrator is given the 'Create' right. A teacher can read this file to go through the marks of the students. She can also enter or alter the marks in this file. Therefore a teacher has 'Read', 'Write', and 'Modify' rights to this file. The students can only view their marks; therefore, the students are given only the 'Read' right to this file. If report cards have to be printed for a class, the class teacher makes a request to the computer operator, who can run a program to print the report cards. This computer operator has been given the 'Execute' right to the application program which prints the report cards. This way by giving appropriate rights to respective users, computer security as well as network security can be increased.

Setting up a computer network

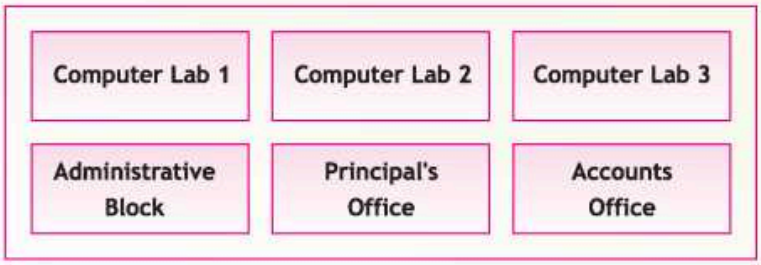
Let us now take an example to see how can a computer network be setup.

Think of a hypothetical Educational Society (say XYZ Educational Society) with its head

office in Chennai (Tamil Nadu) and schools in various parts of the globe. The society is setting up a new senior secondary school, 'SF School', in Bahadurgarh (Haryana).

The 'SF School' will have 3 computer labs with 30 computers in each, 1 Accounts office with 3 computers, 1 Administrative block with 5 computers, and 1 Principal's office with 1 computer. Let us see how a computer network can be set up in the school.

First of all we can draw a rough sketch of the school with computers at various locations as follows:



Independent LANs can be set up in each of the following buildings:

Computer Lab1, Computer Lab2, Computer Lab3, Administrative Block, Accounts Office.

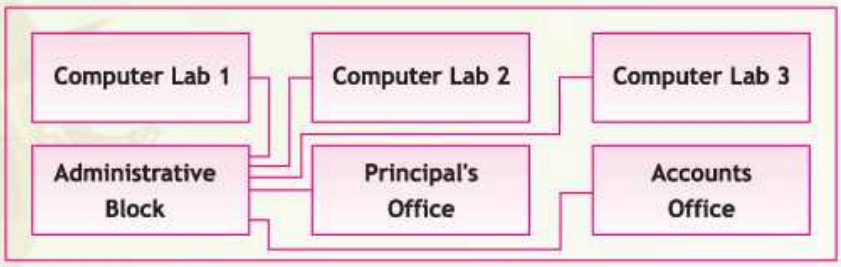
These LANs can be set up in STAR topology using UTP cable (economical, reliable, and easily available). For this 1 switch (with suitable number of ports) will be required in each of these buildings. More than one switches can be used in computer labs if a switch with more than 30 ports is not available.

Two internet connections (broadband for high speed) can be procured in Administrative Office. Two connections should be procured from two different ISPs so that

(i) Internet connection in Administrative office can be kept separate from the other Computer labs where students may do a lot of experimentation.

(ii) If one internet connection is not working, the other can be used in case of urgent requirements.

These buildings can then be connected as follows:



This interconnection will ensure that each building is directly connected to Admisnistrative block. This way internet connection will be available in each building irrespective of the status of the other building.

Server (if any) of the school may be placed in Administrative block so that it remains safe (physically) and a firewall can be set up so that the whole network remains safe from any kinds of virus or intrusion attacks.

There is no need to do any extra efforts or expenses to link the school to its head office. This can be taken care of using the internet connections.

SUMMARY

- A computer network is a collection of interconnected computers and other devices which are able to communicate with each other.
- Communication Media are the links over which the data is sent from one node to another over a network.
- Hardware devices used to form a network are: Cables, NIC, Hub, Switch, Repeater, and Gateway.
- Repeaters are used to regenerate the signal which gets weakened during its transmission. Repeater are generally used to connect two networks which are more than 80 m apart.
- Gateway is a device which connects two different types of networks.
- On the basis of area covered by a network it is categorized as PAN, LAN, MAN, or WAN.

- Physical arrangement of computers in a network is called network topology. Basic network topologies are bus, star and tree.
- Each machine on a network is identified by its IP address.
- Conversion of domain name into IP address is done by DNS (Domain Name Server)
- Denial of service, snooping and intrusion are some common threats for network security.
- Network security can be increased by applying some preventive methods.

Future Trends

4G Technology

The scenario of how computers are networked is changing very fast. Before full implementation of a technology, a new technology is visible at the horizon. One such new technology is 4G. 4G stands for fourth generation of mobile technology. Change from one generation to another involves a major advancement in the technology used. 1G technology was used in the first mobile phones. 1G used analog radio signals. 1G was introduced in 1980s and continued until 1992 when 2G was introduced. 2G technology used a digital format and introduced text messaging. 2G also introduced data services for mobiles, starting with SMS. 3G technology has introduced more efficient ways of carrying data, making it possible to have faster web-services, live chat, fast downloading, video conferencing etc. over mobile phones. Today we are living in the world of 3G. Soon, 4G will rule the mobile market. Unlike previous generations of mobile technology, 4G mobile technology will be used for internet access on computers also, and it will be totally wireless! 4G will provide internet access, high quality streaming video and "anytime, anywhere" voice and data transmission at a much faster speed than 3G. The "anytime, anywhere" feature of 4G is also referred to as "MAGIC" (Mobile multimedia; Anytime/anywhere; Global mobility support; Integrated wireless solution; Customized personal services).

You can explore more about 4G technology on the internet.

Cloud Computing

This is an emerging area of demand based resource sharing, resulting into drastic saving of energy and cost. This is also referred to as 'Green IT'.

You can explore more about Cloud Computing on the internet.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. Which of the following topologies is a combination of more than one topologies?
   a. Bus b. Tree
   c. Star d. None of these
2. Which of the following is used for wireless communication?
   a. Optical Fiber b. UTP cable
   c. Radio Waves d. Coaxial Cable
3. Which of the following is not a transmission medium?
   a. Telephone Network b. Coaxial Cable
   c. Modem d. Microwaves
4. IP addresses of two computers on a network:
   a. Can be the same b. Cannot be the same
   c. Are not defined d. Must match with a third computer
5. Bluetooth can be used for
   a. Long distance communication b. Short distance communication
   c. In mobile phones only d. None of the above
6. Micro waves are
   a. Uni directional b. Omni directional
   c. Guided media d. Not used for communication.

7. Snooping is
   a. A threat to data security
   b. Not a threat to data security
   c. Good for laptops
   d. A topology
8. A repeater
   a. Regenerates the received signal
   b. Destroys the received signal
   c. Can be used as a hub
   d. None of the above
9. Satellite links are generally used for
   a. PANs
   b. LANs
   c. MANs
   d. All of the above
10. A domain name maps to
   a. A URL
   b. An IP address
   c. A website
   d. All of the above

ANSWER THE FOLLOWING QUESTIONS

1. What is a computer network? What are its advantages?
2. What is meant be communication channels? Give two examples of guided media and two examples of unguided media.
3. Which communication channel(s) is/are suitable in each of the following situations:
   a. Setting up a LAN
   b. Transfer of data from a laptop to a mobile phone.
   c. Transfer of data from one mobile phone to another.
   d. Creating a remote control that can control multiple devices in a home.
   e. Very fast communication between two offices in two different countries.
   f. Communication in a hilly area
   g. Communication within a city and its vicinity where cost of cabling is too high.
4. Why is a switch called an intelligent hub?

5. When is a repeater used in a computer network?
6. Diagramatically show how would you connect 6 PCs, 1 server, 1 printer, and 2 scanners in
   a. Star topology b. Bus topology
7. Two engineers in the same room have connected their Palm-tops using bluetooth for working on a Group presentation. Out of the following, what kind of Network have they formed?

   LAN, MAN, PAN, WAN
8. What is a MAC address? What is the difference between a MAC address and an IP address?
9. Give some examples of domain names and URLs. How is a domain name different from a URL?
10. What is domain name resolution?
11. Define Network security? What kind of attacks can be made on data and computer networks?
12. List some methods which are used for network security.
13. Differentiate (with examples wherever possible) between :
    a. LAN and MAN b. MAN and WAN
    c. Hub and Switch d. Guided and Unguided media
14. Write one advantage of star topology over bus topology and one advantage of bus topology over star topology.
15. Ishika Industries has set up its new production unit and sales office at Ranchi. The company compound has 4 buildings as shown in the diagram below:

Distances between these buildings are as follows:

Administrative Office to Factory A	150 m
Factory A to Factory B	50 m
Factory B to Sales Office	100m
Sales Office to Administrative office	200m
Administrative Office to Factory B	125 m

Number of Computers in each of the buildings is follows:

Administrative Office	15
Factory A	25
Factory B	18
Sales Office	15

1. Suggest a cable layout of connections between the buildings so that each building is directly connected to Administrative Office.
2. Suggest the most suitable place (i.e. building) to house the server of this production unit with a suitable reason.
3. Suggest the placement of the following devices with justification:
   (i) Repeater
   (ii) Hub/Switch
4. The Administrative office of this unit is to be linked with the head office situated in Patiala (Punjab). What will be the most economical way to do this? Justify your answer.

LAB EXERCISES

1. Find the IP addresses of at least five computers in your school.
2. Find the MAC addresses of at least 2 computers in your lab. Then verify their manufacturer's name on the net.
3. Find the layout of LAN in your school's labs. If you think some modifications can be done in the layout, note these down in your notebook.
4. Find the name of Internet Service Provider of your school.
5. Find the IP address of your school's web site.

TEAM BASED TIME BOUND EXERCISES

(Team size recommended: 3 students each team)

1. A school building is divided into 4 blocks (A, B, C, and D). Each block is at a distance of 25m from its adjacent blocks. Each block has 1 computer lab with 15 computers each. Each block also has some other rooms (maximum 10) with 1 computer each. The school has only one internet connection. The computer network in the school has to be restructured with the following goals in mind:
   - Each lab has to have an independent LAN.
   - All the computers in the school should have internet access. (The school does not want to have any other new internet connection.)
   - For internet access a computer should be dependent only on one point and not on multiple points. It means that if the internet connection is active, any computer should be able to access it directly irrespective of whether some other computer in the school is ON or OFF.
   - Any two computers in the school should be able to communicate with each other irrespective of whether any other computer in the school is ON or OFF.

   The job of each team is to design a layout for this new network structure. Each team has to specify
   - The layout of the network structure diagrammatically.

- Topology/topologies is/are used in the layout.
- How the design will be able to fulfill all the mentioned requirements.

2. Find the average price and specifications of each of the following network devices in the market: Switch , Cable (UTP), LAN Card. Each team has to specify:
   - Different types of switches available in the market, their prices, and number of ports.
   - Names of brands of UTP Cable available in the market, along with their prices.
   - Different types of LAN cards available in the market along with their prices.

OPEN SOURCE CONCEPTS

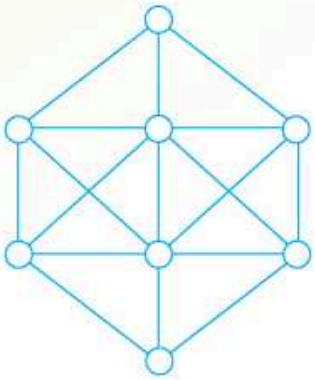
Learning objectives

After learning this chapter the student will be able to:

- Understand about OSS/FOSS/FLOSS
- Cite different examples of OSS
- Describe open document format
- Understand character encoding in Indian languages
- Know about open type/true type/static/dynamic fonts

Puzzle [2]

Write the numbers 1 to 8 in the given circles so that no two numbers joined by a line differ by 1.



Introduction

Computers and internet have transformed our lives. Software is required to work on a computer but the software that we buy or download only comes in the **compiled** ready-

to-run version. It is next to impossible to modify the compiled version of the software. At times we feel the need to change certain features of the software but are unable to do so.

In this chapter we will study about software which are developed collaboratively and they can be modified as well. Such software are available in many forms like Free Software, Open Source Software (OSS), Free Open Source Software (FOSS) and Free/Liberal Open Source Software (FLOSS).Well, have you noticed something common in all these terms. That's right! It is the word free. People often correlate this word with the cost. They think that these software are available for free. Practically, these software can be acquired at very little or no cost. But, here, "free" means freedom to use.

These software can be studied, copied, redistributed freely and even modified according to one's need without seeking any kind of permission. In order to modify such software the developers also provide the source code to the users.

There do exist software which are actually "free" in the sense of price. These are known as Freeware. Lots of freeware can be downloaded from the internet for various different purposes such as currency converters, drawing graphs and charts etc. But freeware may not come with the source code. Therefore freeware differ from free software. The focus in this chapter is on free software rather than freeware.

The first formal definition of "free software" was given in 1983 by Richard Stallman, a long time member of the hacker community at the MIT Artificial Intelligence Laboratory. He insisted that a free software should give the following four freedoms to users:

- Freedom 0: The freedom to run the program for any purpose.
- Freedom 1: The freedom to study how the program works, and change it to make it do what you wish.
- Freedom 2: The freedom to redistribute copies so as to help your neighbour.
- Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

Examples of free software include the Linux Kernel, MySQL Relational Database, Apache web server, OpenOffice.org office suite and TeX and LaTeX typesetting systems.

Later on, the term "free software" was recoined as "open source software (OSS)" and soon after as "free open source software (FOSS)". In order to avoid the ambiguity in the word "free", in 2001, FOSS was termed as FLOSS, short form of "free/liberal open source software".

Know More!

You can get more information on open source software from opensource.org.

NRCFOSS :

National Resource Centre for Free and Open Source Software (NRCFOSS) is an initiative of the Department of Information Technology, Ministry of Communications & Information Technology, Government of India. NRCFOSS contributes to the growth of FOSS in India through Research & Development, Human Resource Development, Networking & Entrepreneurship development, as well as serves as the reference point for all FOSS related activities in the country.

Examples of FLOSS/FOSS :

Operating systems and Desktop environments

- **Linux** - Operating system kernel
- **Ubuntu** - Linux distribution with full complement of software for everyday use.
- **Google Chrome OS** - Lightweight operating system based around the web browser
- **Android smart** - phone operating system - by Google / Open Handset Alliance
- **Symbian smart** - phone operating system - by Nokia

Graphics and multimedia

- **GIMP** - Bitmap graphics editor, similar to Adobe Photoshop
- **Blender** - Advanced 3D modelling and rendering application.
- **Songbird** - similar to iTunes with built-in browser.

- **Audacity** - digital audio editor
- **Ardour** - digital audio workstation
- **F-Spot** - Photo manager

Office software

- **OpenOffice.org** - office productivity software. Comparable to Microsoft Office. It uses an open file format and can read and write Microsoft Office files
- **NeoOffice** - Mac OS X version of OpenOffice
- **PDFCreator** - creates PDFs from any Windows program.

Internet related software

- **Apache webserver** - web server
- **Mozilla Foundation**
  - **Mozilla Firefox** - web browser
  - **Mozilla Thunderbird** - mail client
- **Google Chrome** - Google's web browser

Other

- **Celestia** - 3D space simulation software.
- **Flight Gear** - flight simulator
- **Second life** - virtual world viewer
- **Wine** - a compatibility layer for computers running Linux that enables them to run many applications that were originally written for MS Windows

Programming related

- **Eclipse** - software framework and Java IDE
- **PHP** - server-side programming language
- **PERL** - Dynamic programming language

- **Python** - versatile, clean and powerful programming language used for cross-platform desktop applications, server-side scripting for websites, and scripting within java and .net environments
- **MySQL** - Database management system
- **Java** - Programming language

More comprehensive list of open-source software can be found at :
http://en.wikipedia.org/wiki/List_of_open_source_software_packages

GNU/Linux - The GNU Project was launched in 1983 by Richard Stallman of Free Software Foundation (FSF) to develop a complete Unix-like operating system which is free software: the GNU operating system. Unix-like operating systems are built from a collection of libraries, applications and developer tools, plus a kernel to allocate resources and to talk to the hardware. GNU is often used with the Linux kernel. The combination of GNU and Linux is the GNU/Linux operating system, now used by millions.

Firefox - Firefox is a free and open source web browser produced by Mozilla Foundation. Firefox runs on various versions of GNU/Linux, Mac OS X, Microsoft Windows and many other Unix-like operating systems.

Open Office - Open Office is the leading open source office suite for word processing, spreadsheets, presentations, databases etc. It is available in many languages. It stores files in open document format (ODF) for data interchange that is its default file format.

NetBeans - NetBeans began in 1996 as verify, a Java Integrated Development Environment (IDE) student project, under the guidance of the Faculty of Mathematics and Physics at Charles University in Prague. In 1999 it was bought by Sun Microsystem which open-sourced the NetBeans IDE in June of the following year. The NetBeans community has since continued to grow, thanks to individuals and companies using and contributing to the project. NetBeans refers to both a platform framework for Java desktop applications, and an IDE for developing applications with Java, JavaScript, PHP, Python, Ruby, C, C++, and others.

BOSS(Bharat Operating Systems Solutions) - BOSS is a free Indian Operating System based on GNU/LINUX developed by C-DAC(Center for Development of Advance Computing). BOSS makes it easier for a Microsoft Windows user to shift to GNU/LINUX platform which is there with variant features. Currently BOSS GNU/Linux Desktop is available in many Indian Languages which also enables the non-English literate users in the country to be exposed to Information and communication technology and to use the computer more effectively.

Open Source Software Security

A commonly voiced concern about open source software:

"If anyone can contribute to open source software, doesn't it become a free-for-all full of loopholes?"

The Answer is while anyone can propose a contribution to an open source project, any actual change must go through a small core group of maintainers first. Getting a change incorporated into an open source project is thus rather like getting an article published in a scientific journal.

Open source software is more heavily tested than their commercial counterparts as it can be downloaded by anybody around the world and any one of them can discover a bug or security flaw and submit those reports back to the project.

Also since it's an Open Source, if any bug or flaw is found, project's core maintainers, while potentially embarrassed, have no further reason to cover up the flaw like their proprietary counterpart. Also the loop-hole or bug is available to every one and the customer can take the preventive measure accordingly and fix comes faster.

Common open standards

Open Document Format

The Open Document format (ODF) is a format for office documents, such as spreadsheets, databases, presentations and word-processing documents. Open Document is a free and open format. For governments, businesses, archivists and others, it's critical to store documents in a way that can be read for years to come. Proprietary digital file formats are typically changing with every new version of the software, so there should be some format which supports files created in any application. The data should be the center of importance not the application. Office suite applications, as always, need a file format that is designed to organize the data when it moves away from the application. People with different machines in different places should be able to open and edit the data in a file. ODF offers an open alternative to the formats used by all of the existing Office application versions for text, spreadsheet, presentation, and other kinds of documents. Open Document's main file extensions are .odt (for text documents), .ods (for spreadsheets), and .odp (for presentations). These will be more commonly recognized when more people and organizations adopt Open Document-ready software.

Ogg Vobris

Ogg Vorbis is a new audio compression format developed by Xiph.org. It is an open, patent-free, professional audio encoding and streaming technology with all the benefits of Open source. It is comparable to other formats such as MP3, VQF, AAC etc. used to store and play digital music.

Character Encoding

A character encoding system consists of a code that associates each character from a given system with something else, such as a sequence of natural numbers, binary numbers or electrical pulses, in order to facilitate the transmission of data (generally numbers and/or text) through telecommunication networks or for storage of text in

computers. ASCII, EBCDIC, and UNICODE are the most widely used character encoding systems for computers.

Indian Language Computing

Let us recall from Annexure I of Class XI that in order to communicate with the computers some kind of a binary code is required. In this regard, a detailed study was made on BCD code which is a 4 bit code. However, BCD code is not sufficient enough to accommodate even all the characters of English alphabet and digits. To accommodate all these, a higher bit code is required. Among such codes, the most popular is the ASCII code (American Standard Code for Information Interchange). It is a 7 bit code that can store $2^7 = 128$ characters. In the earlier days most computers were using an 8 bit system. This extra bit gave computer developers lot of empty spaces which was used for different purposes. In order to work with Indian languages, these unspecified spaces were used for Indic characters. The efforts were very creative and gave good results despite the fact that there was no or very little support from the operating systems. Since no general rules and methodologies were adopted in developing Indic characters, different developers developed these characters in their own ways. This created compatibility issues across different programs and across different operating systems. For example browsers like Internet Explorer did not support all Indic characters and displayed distorted texts.

On one hand, more and more work had been undertaken to facilitate the use of Indian languages on computers but on the other hand it was becoming difficult to maintain consistency across different programs developed for the same set of Indic characters. In such a scenario, it was important to have a common standard for coding Indian scripts. In 1991, the Bureau of Indian Standards (BIS) adopted the Indian Script Code for Information Interchange (ISCII), the ISCII standard that was evolved by a standardization committee, under Department of Electronics during 1986-88.

ISCII is an 8 bit encoding system as compared with the 7 bit ASCII. The lower 128 code points are plain ASCII while the upper 128 code points are ISCII specific containing the basic alphabets required for the 10 Indian scripts (Assamese, Bengali, Devanagari, Gujarat, Gurmukhi, Kannada, Malayalam, Oriya, Tamil and Telugu) which have originated from the Brahmi script.

Know More!

The **Indic scripts** are a family of abugida (alphabetic-syllabary) writing systems. They are used throughout South Asia, Southeast Asia, and parts of Central and East Asia, and are descended from the Brahmi script of the ancient Indian subcontinent. They are used by languages of several linguistic families: Indo-European, Dravidian, Tibeto-Burman, Mongolic, Austro-Asiatic, Austronesian, Tai, and possibly Korean (hangul). They were also the source of the dictionary order of Japanese kana.

UNICODE

For a long time ASCII has been the standard code used worldwide. ISCII was the Indian contribution to work with Indian languages. Similarly, to work with other languages of the world, people were making efforts to use ASCII along with the extra 8th bit. Efforts were continued to conceive a single standard code which could cater to all world's languages and it was UNICODE.

Unicode was developed with the aim to conceive a single standard code which could manage to represent all languages of the world.

Unicode characters are represented in one of the three encoding forms: a 32-bit form (UTF-32), a 16-bit form (UTF-16), and an 8 bit form (UTF-8). The 8-bit, byte oriented form, UTF-8, has been designed for ease of use with existing ASCII-based systems.

Advantages of Unicode

- With the usage of Unicode, single versions of software were developed instead of language-specific versions that reduced the complexity.
- UNICODE is supported by most OS and application vendors. This ensures platform, vendor and application independence.
- Incorporating Unicode into applications and websites offers significant cost savings than proprietary solutions.
- It allows data to be transported through several different systems without distortion.

- Since every number and character combination is unique, the representation results in a true standard.

Different Types of Fonts

Post Script

This font format was developed by Adobe in 1980's. This font consists of two parts which are both necessary for the font to be properly printed and displayed on screen. With most operating systems, these fonts can be installed simply by being placed in the system's folder. However, for Micro Soft operating systems predating windows 2000, they need to be installed using ATM (Adobe Type Manager) utility.

True Type

This format was jointly developed by Apple and Microsoft in the late 80s. These fonts contain both the screen and printer font data in a single component, making the fonts easier to install. And that is why these are a good choice for those who find the installation of fonts difficult.

Open Type

This is the latest font format which is a joint effort by Apple and Microsoft. Like True Type fonts, this contains both the screen and printer font data in a single component. However, open type fonts support multiple platforms and expanded character sets. Additionally, open type format allows the storage of upto 65,000 characters. This additional space provides freedom to include add-ons such as small caps, old style figures, alternate characters and other extras that previously needed to be distributed as separate fonts.

However, not all open type fonts contain additional characters. Many fonts have been converted from either PostScript or TrueType formats without expanded character sets to take advantage of the cross-platform functionality benefits of Open Type. OpenType fonts that do contain expanded character sets are referred to informally as OpenType Pro fonts. Support for OpenType Pro fonts is increasing, but the format is yet to be fully supported by all applications.

Static and Dynamic fonts

Static fonts - In these types of fonts the characters are designed and digitized and then stored in font files. Every time printing takes place, same character will appear with same shape e.g. Times New Roman, Arial etc.

Dynamic fonts - Dynamic font is a web browser technology used when visiting any website that uses fonts which are not installed on client's machine. The web browser would not be able to display the page properly, but will select one of fonts available on client machine. In this the characters are redefined at each occurrence (everytime they are displayed or printed). All hand written fonts such as handwritten alphabets, calligraphic letters, graffiti etc are dynamic fonts because of individual variations.

This technology is helpful in displaying multilingual websites.

Entering Indian language text

Many tools have been developed to facilitate the typing of Indian language text. These tools broadly support two types of text entries:

- Phonetic Text Entry
- Keymap based Text Entry

Phonetic Text Entry (Transliteration)

In this type of text entry, traditional keyboards with English keys are used. But while typing, the Indian alphabets are written phonetically (i.e., the way they sound as per the pronunciation) in English Script and then converted to corresponding language word. For e.g. we will type "mera desh mahaan" from English keyboard and the relevent phonetic key entry software will transliterate it in the language selected eg. Hindi. ("मेरा देश महान")

Keymap based Text Entry

In this method the keyboard keys are mapped to specific characters using a keymap. The whole arrangement of mapping the keyboard keys to specific language characters is known as keymap. A keymap is internally stored as a table. Multiple keymaps are used to store complete keymapping e.g. a table to represent the keymappings without any additional key press(such as shift) ; a table representing keymappings along with Shift key press; along with Ctrl key press and so on.

Indian language keymaps are known as INSCRIPT keymaps or Indian Script Keymaps.

ENGLISH KEYBOARD WITH INSCRIPT OVERLAY

The ASCII characters of a standard QWERTY keyboard are on the left half of a key. The INSCRIPT (Indian Script) overlay characters are shown on the right half of a key. CAPS LOCK is used to select the INSCRIPT overlay.

NUKTA CHARACTERS IN INSCRIPT OVERLAY

When Nukta "-" is typed after character, the character shown to its left on the key, is obtained.

INSCRIPT OVERLAY FOR ASSAMESE

Know More!

You can get more information on Indian script keymaps from http://tdil.mit.gov.in/isciichart.pdf.

Future Trends

At present very few open source endeavors which involve few people are there and are largely unstructured and undisciplined. But as this concept matures in future more OSS projects will come up and will adopt more structured methodologies to code, control and coordinate. In future this concept will be adopted at a big level by private and government sector. This will force companies to embrace open source and will force the open source community to innovate in line with more complex need of government and business.

Summary

- Free and open source software, also OSS, FOSS, or FLOSS (free/libre open source software) is software that is liberally licensed to grant the user right to use, study, change, and improve its design through the availability of its source code.
- NRCFOSS - National Resource Centre for Free and Open Source Software.
- ODF (Open Document Format) offers an open alternative to the formats used by all of the existing software.
- Ogg Vorbis is open audio compression format.
- A character encoding system consists of a code that associates each character from a given system with something else, such as a sequence of natural numbers, binary numbers or electrical pulses, in order to facilitate the transmission of data (generally numbers and/or text) through telecommunication networks or for storage of text in computers
- ISCII - Indian Script Code for Information Interchange is common standard for coding Indian Scripts.
- UNICODE - Universal Code is a single standard code which can cater to all the world's languages.
- Post Script font consists of two parts which are both necessary for the font to be properly printed and displayed on screen.
- True type fonts contain both the screen and printer font data in a single component.
- Open type fonts support multiple platforms and expanded character sets.
- On the basis of their configuration, fonts are categorized as static fonts and dynamic fonts.
- For entering text in Indian script keymap based or phonetic text based entry can be done.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1) Open Document's main file extensions are

   (a) .odt (b) .ods

   (c) .odp (d) All of these

2) Technology no longer protected by copyright, available to everyone, is considered to be

   (a) Proprietary (b) Open

   (c) Experimental (d) in the public domain

3) Which of the following codes uses 7 bits to represent a character

   (a) ISCII (b) ASCII

   (c) UNICODE (d) All of the above

4) How many bits are used by UTF-8 to represent a character

   (a) 7 (b) 8

   (c) 16 (d) 32

5) ASCII stands for

   (a) American System Code for Information Interchange

   (b) American Standard Code for Information Interchange

   (c) American Standard Code for Interchange Information

   (d) American System Code for Interchange Information

6) ISCII stands for

   (a) Indian System Code for Information Interchange

   (b) Indian Standard Code for Information Interchange

   (c) Indian Script Code for Interchange Information

   (d) International Standard Code for Interchange Information

7) ISCII is

(a) An 8 bit character code (b) A 16 bit character code

(c) A 7 bit character code (d) Same as ASCII

8) Which of the following is open source operating system

(a) DOS (b) Windows 2010

(c) BOSS (d) Mac

9) Following operations are possible with OSS

(a) Free download

(b) Source code is available

(c) You can redistribute the modified version

(d) All of the above

10) Platform independence is provided by

(a) ASCII (b) ISCII

(c) UNICODE (d) All of the above

ANSWER THE FOLLOWING QUESTIONS

1) Define Open source software. Give at least two examples of OSS. Justify that free software is not open source software.

2) Give examples of two OSS in each of the following categories :

(a) OS (b) Graphics and animation

(c) Office software (d) Internet related software

(e) Programming related software.

3) What is a character encoding system?

4) Differentiate between true type font and open type font.

5) What do you understand by ODF?

6) What is Ogg Vorbis?

7) Expand the following terms:

OSS, FLOSS, ISCII, ASCII, BOSS, NRCFOSS

8) What is UNICODE?

9) Write short notes on GNU, BOSS, ISCII

10) What is the difference between static and dynamic fonts?

LAB EXERCISES

1) Find out which software in your school lab are open source.

2) Note down the category of software (system software or application software) to which they belong to.

3) If any of them is application software then specify its area of application.

4) Search on internet about the features of mySQL and Netbeans.

TIME BOUND TEAM BASED EXERCISE

(Team size recommended: 3 students each team)

1) Download open office on your computer. Write down its components, their areas of applications, their file extensions.

2) Procure BOSS and install it (with the permission of your teacher) on your computer and write down the features which are similar to windows and also which are not similar to windows.

GUI PROGRAMMING - A REVIEW

Learning Objectives

After studying this lesson the students will be able to:

- Identify, name and state the usage of the different components of the NetBeans IDE.
- Identify and name the various methods and properties associated with the various form controls
- Create simple applications in Java using NetBeans IDE.
- Create GUI applications using the concepts of variables and control structures.

NetBeans IDE allows us to develop applications by dragging and positioning GUI components from a palette onto a container. The GUI builder automatically takes care of the correct spacing and alignment of the different components relative to each other. The JFrame acts as a container for the elements like the JLabel, JButton, JTextArea and allows direct editing of their associated properties at the design time and run time. The related methods are used in the code to develop applications for a specific task. The concept of variables and control structures are used to simplify the code of the applications.

Puzzle [3]

This group of phrases has something amazing hidden in it. Try and find out what is so unusual about this group of phrases? Is there any specific pattern in each line? *

Bar crab

Borrow or rob

Straw warts

Live evil

A man, a plan, a canal--Panama!

Delia failed.

Evil olive

Pull up if I pull up.

Step on no pets.

Ten animals I slam in a net.

Was it a bat I saw?

Was it a car or a cat I saw?

We found an interesting pattern in the puzzle and we will use this concept to develop an application in netbeans.

Introduction

Let us take a journey back in time and think as to what the world was like 15 years ago. Amazon was a large river in South America. Yahoo was a term from Gulliver's Travels. A googol was a very large number (one followed by a hundred zeroes) and to get our tickets booked we had to go to shops called 'travel agents'. In case we fell sick and went to a hospital, our records were maintained on paper. If the doctor wanted to refer to a particular patient's record, he had to instruct a assistant to hunt for information from the pile of files.

Nowadays hospitals use computers to keep the records of patients - medical history, details on what medication to give to a patient, the prescribed dosage and also personal details that can be accessed at the click of a button. The entire information is entered into a computer using a front end that accepts the different patient details. We learnt how to design valid front end forms in class XI in Java using the NetBeans IDE. This chapter will help us recapitulate all the concepts learnt in class XI.

NetBeans IDE

It can be used to create java applications very easily using the efficient GUI builder. Let us quickly recap the different components of the NetBeans IDE:

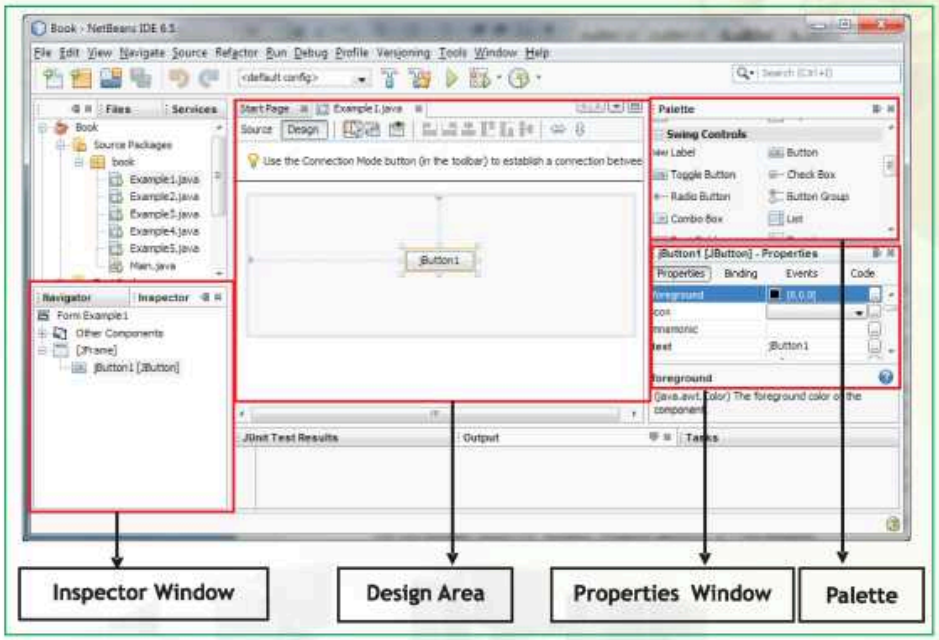


Figure 3.1 NetBeans IDE

1. Title Bar
2. Menu Bar with pull down menus
3. Toolbars
4. GUI builder: It is an area to place components on the form visually. There are two views of the GUI builder- the Design View and the Source View. We can switch over from one view to another by simply clicking on the source and design tabs directly above the Design Area.
5. Palette: Palette contains controls or components used to create GUI applications.

6. Inspector Window: This window is used to display a hierarchy of all the components or controls placed on the current form.
7. Properties Window: Using this window we can make changes in the properties of currently selected control on the form.
8. Code Editor Window: - It is the area where we write code for our java application.

Components

Components (also known as "widgets") are the basic interface elements the user interacts with: jlabels, jbuttons, jtextfields etc. Components are placed on a container (like the jFrame). There are two types of controls :

- **Parent or container controls:** They act as a background for other controls. For example-Frame. When we delete a parent control, all its child controls get deleted. When we move a parent control all its child controls also move along with it.
- **Child controls:** controls placed inside a container control are called child controls. For example-Text Field, Label, Button etc.

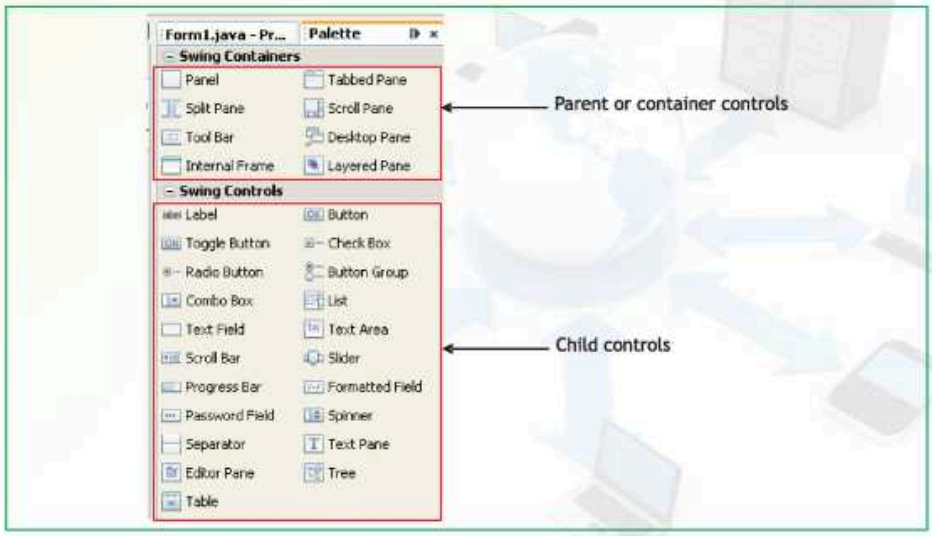


Figure 3.2 Parent and Child controls

Creating a New Project

The steps to create a new project are:

1. Select New Project from the File menu. You can also click the New Project button in the IDE toolbar.
2. In the Categories pane, select the General node. In the Projects pane, choose the Java Application type. Click the Next button.
3. Enter the name of the project in the Project Name field and specify the project location. Do not create a Main class here.
4. Click the Finish button.

Let us recap the relation between a Project, Form and Components. Each application is treated as a Project in NetBeans and each project can have one or multiple forms and this fact is clear from the Projects window as shown in Figure 3.3.

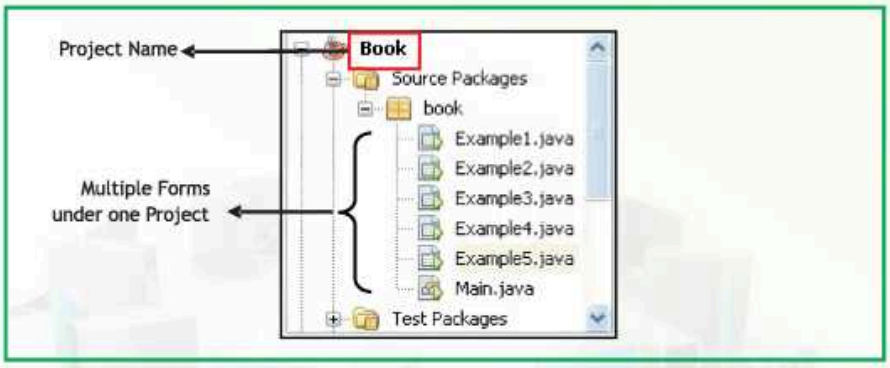


Figure 3.3 Project Window Showing Multiple Forms

Further each form can have one or more elements - some of which may be visible and some invisible. The visible components are all shown under the Frame Component and the non-visible components are part of Other components.

We use the drag and drop feature of NetBeans to place components on the form to design an effective interface for our applications. The first step that we undertook while designing our applications was adding a new jFrame form. The jFrame is a window with

title, border, (optional) menu bar and is used to contain all other components placed by the user on the form. Some of the properties of the jFrame form are defaultCloseOperation and Title.

Property	Description
defaultCloseOperation	Sets action to be performed when the user attempts to close the form.
Title	Sets the text to be displayed in the Title bar of the form window.

Figure 3.4 Properties of the jFrame Form

Any component of GUI front-end (the form itself and the swing containers and controls placed in the form) of an application is an object. Each of these objects belongs to its corresponding class predefined in Java. For example, a form is an object of JFrame class, all the textfields are objects of JTextField class, and so on. Each object has some properties, methods, and events associated with it using which you can control the object's appearance and behaviour.

Properties of an object are used to specify its appearance on the form. For example to set the **background** colour of a textfield you change its background property; to set its **font** you change its font property; and so on.

Methods are used to perform some action on the object. For example to display something in a textfield you can use its **setText()** method, to extract the contents of a textfield you can use its **getText()** method. Methods can be divided into two categories- *getters* and *setters*.

- Getters are the methods which extract some information from the object and return it to the program. Getters start with the word get. Examples of getters **are: getText(), getForeground(), getModel(), isEditable** etc.
- Setters are the methods which set some properties of the object so that the object's appearance changes. Setters start with the word set. Examples of setters are: **setText(), setForground(), setModel()** etc.

Events are the actions which are performed on controls. Examples of events are: **mouseClick, mouseMoved,keyPressed** etc. When the user performs any action on a

control, an event happens and that event invokes (sends a call to) the corresponding part of the code and the application behaves accordingly.

After setting the properties of the jFrame we can start placing components like jButton on the jFrame form. A button is a component that the user presses or pushes to trigger a specific action. When the user clicks on the button at runtime, the code associated with the click action gets executed. The various methods and properties associated with the jButton are summarized in Figure 3.5.

Property	Description
Background	Sets the background color.
Enabled	Contains enabled state of component - true if enabled else false.
Font	Sets the font.
Foreground	Sets the foreground color.
horizontal alignment	Sets the horizontal alignment of text displayed on the button.
Label	Sets the display text.
Text	Sets the display text

Method	Description
getText()	Retrieves the text typed in jButton. `String result=<button-name>.getText( );`
setEnabled	Enables or disables the button. `<button-name>.setEnabled(boolean b);`
setText()	Changes the display text at runtime. `<button-name>.setText(String text);`
setVisible	Makes the component visible or invisible - true to make the component visible; false to make it invisible. `<button-name>.setVisible(boolean aFlag);`

Figure 3.5 Properties and Methods of the jButton

We developed simple real life applications wherein on the click of the button we accepted the data from the user in the jTextField and after processing the data the result was displayed in a jTextField or a jLabel. jTextField allows editing/displaying of a single line of text. jTextField is an input area where the user can type in characters whereas a jLabel provides text instructions or information. It displays a single line of read-only text, an image or both text and image. The various methods and properties associated with the jTextField and jLabel are summarized in Figure 3.6 and 3.7 respectively.

Property	Description
Background	Sets the background color.
Border	Sets the type of border that will surround the text field.
editable	If set true user can edit textfield. Default is true.
enabled	Contains enabled state of component- True if enabled else false.
font	Sets the font.
foreground	Sets the foreground color.
horizontalAlignment	Sets the horizontal alignment of text displayed in the textField.
text	Sets the display text
toolTipText	Sets the text that will appear when cursor moves over the component.

Method	Description
getText()	Retrieves the text in typed in jTextField. `String result=<textfield-name>.getText( );`
isEditable()	Returns true if the component is editable else returns false. `boolean b=<textfield-name>.isEditable( );`

isEnabled()	Returns true if the component is enabled,else returns false. `boolean b =<textfield-name>.isEnabled( );`
setEditable	Sets whether the user can edit the text in the textField. true if editable else false. `<textfield-name>.setEditable(boolean b);`
setText()	Changes the display text at runtime. `<textfield-name>.setText(String t);`
setVisible()	Makes the component visible or invisible - true to make the component visible; false to make it invisible. `<textfield-name>.setVisible(boolean b);`

Figure 3.6 Properties and Methods of the jTextField

Property	Description
background	Sets the background color.
enabled	Contains enabled state of component- true if enabled else false.
font	Sets the font.
foreground	Sets the foreground color.
horizontalAlignment	Sets the horizontal alignment of text displayed in the component.
text	Sets the display text

Method	Description
getText()	Retrieves the text in typed in jLabel. `String result=<label-name>.getText();`
isEnabled()	Returns true if the component is enabled,else returns false. `boolean b=<label-name>.isEnabled();`
setText()	Changes the display text at runtime. `<label-name>.setText(String t);`
setVisible()	Makes the component visible or invisible - true to make the component visible; false to make it invisible. `<label-name>.setVisible(boolean b);`

Figure 3.7 Properties and Methods of the jLabel

When we had become familiar with the usage of jTextField and jLabel controls then we developed an application in which we wanted to accept multiline input and display multiline output. Well can you recall the name of the component. Exactly the component is Text Area component. This component allows us to accept multiline input from the user or display multiple lines of information. This component automatically adds vertical or horizontal scroll bars as and when required during run time. The various methods and properties associated with the jTextArea are summarized in Figure 3.8.

Property	Description
background	Sets the background color.
columns	Sets number of columns preferred for display.
editable	If set true user can edit textfield. Default is true.
enabled	Contains enabled state of component- true if enabled else false.
font	Sets the font.

foreground	Sets the foreground color.
lineWrap	Indicates whether line of text should wrap in case it exceeds allocated width. (Default is false)
rows	Sets number of rows preferred for display.
text	Sets the display text
wrapStyleWord	Sends word to next line in case lineWrap is true and it results in breaking of a word, when lines are wrapped.

Method	Description
append()	Adds data at the end. `<textarea-name>.append(String str);`
getText()	Retrieves the text in typed in jTextArea. `String str = <textarea-name>.getText();`
isEditable()	Returns true if the component is editable else returns false. `boolean b = <textarea-name>.isEditable();`
isEnabled()	Returns true if the component is enabled, else returns false. `boolean b = <textarea-name>.isEnabled();`
setText()	Changes the display text at runtime. `<textarea-name>.setText(String t);`

Figure 3.8 Properties and Methods of the jTextArea

Let us try and recollect the name of the component which can be used to enter confidential input like passwords which are single line. That's right the component is jPassword. We can suppress the display of input as this component allows us to input confidential information like passwords. Each character entered can be replaced by an echo character. By default, the echo character is the asterisk, *. The properties of jPassword are summarized below:

Property	Description
background	Sets the background color.
font	Sets the font.
foreground	Sets the foreground color.
text	Sets the display text
echoChar	Sets the character that will be displayed instead of text.

Figure 3.9 Properties of jPassword

Well we used radio buttons when we wanted to provide the user several choices and allowed him to select one of the choices (the radio buttons belong to a group allowing the user to select single option). But radio buttons occupy a lot of space. Thus, in case of too many options we used Combo boxes as they help save space and are less cumbersome to design as compared to radio button. We used check box and list when we wanted to display multiple options like selecting favourite sports or ordering multiple food items in a restaurant. The list is a preferred option over check box in situations wherever multiple options are required to be selected from a large number of known set of options as they help save space and are less cumbersome to design as compared to check boxes. The properties and methods of jRadioButton are summarized below:

Property	Description
background	Sets the background color.
buttonGroup	Specifies the name of the group of button to which the jRadioButton belongs.
enabled	Contains enabled state of component -true if enabled else false.
font	Sets the font.
foreground	Sets the foreground color.
label	Sets the display text.
text	Sets the display text.
Selected	Sets the button as selected, if set to true, default is false.

Method	Description
getText()	Retrieves the text displayed by radio button. `String str = <radiobutton-name>.getText();`
isSelected()	Returns true if the component is checked else returns false. `boolean b = <radiobutton-name>.isSelected();`
setText()	Changes the display text at runtime. `<radiobutton-name>.setText(String t);`
setSelected()	Checks(true) or unchecks the radio button. `<radiobutton-name>.setSelected(boolean b);`

Figure 3.10 Properties and methods of the jRadioButton

jCheckBox is a small box like component that is either marked or unmarked. When you click on it, it changes from checked to unchecked or vice versa automatically. The properties and methods of jCheckBox are summarized below:

Property	Description
background	Sets the background color.
buttonGroup	Specifies the name of the group of button to which the jCheckBox belongs.
font	Sets the font.
foreground	Sets the foreground color.
label	Sets the display text.
text	Sets the display text
selected	Sets the check box as selected if set to true, default is false.

Method	Description
getText()	Retrieves the text typed in `String str = <checkbox-name>.getText();`
isSelected()	Returns true if the component is checked else returns false. `boolean b = <checkbox-name>.isSelected();`
setText()	Changes the display text at runtime. `<checkbox-name>.setText(String t);`
setSelected()	Checks(true) or unchecks the checkbox. `<checkbox-name>.setSelected(boolean b);`

Figure 3.11 Properties and methods of the jCheckBox

jComboBox is like a drop down box - you can click a drop-down arrow and select an option from a list whereas jList provides a scrollable set of items from which one or more may be selected. The properties and methods of jComboBox and jList are summarized below:

Property	Description
background	Sets the background color.
buttongroup	Specifies the name of the group of button to which the jComboBox belongs.
editable	If set true user can edit ComboBox. Default is true.
enabled	Contains enabled state of component- True if enabled else false.
font	Sets the font.
foreground	Sets the foreground color.
model	Contains the values to be displayed in the combobox.
text	Sets the display text
selectedIndex	Sets the index number of the element which should be selected by default.
selectedItem	Sets the selected item in the combobox. selectedItem and selectedIndex are in synchronization with each other.

Method	Description

getSelectedItem()	Retrieves the selected item.
	`Object result =`
	`<combobox-name>.getSelectedItem();`

getSelectedIndex()	Retrieves the index of the selected item.
	`int result =`
	`<combobox-name>.getSelectedIndex();`

setModel()	Sets the data model that the combo box uses to get its list of elements.
	`<combobox-name>.setModel`
	`(ComboBoxModel aModel);`

Figure 3.12 Properties and methods of the jComboBox

Property	Description
background	Sets the background color.
enabled	Contains enabled state of component- true if enabled else false.
font	Sets the font.
foreground	Sets the foreground color.
model	Contains the values to be displayed in the list.
selectedIndex	Contains the index value of selected option of the control.
selectionMode	Describes the mode for selecting values. - SINGLE (List box allows single selection only) - SINGLE_INTERVAL (List box allows single continuous selection of options using shift key of keyboard) - MULTIPLE_INTERVAL (List box allows multiple selections of options using ctrl key of keyboard)

Method	Description
getSelectedValue()	Returns the selected value when only a single item is selected, if multiple items are selected then returns first selected value. Returns null in case no item selected `Object result= <list-name>.getSelectedValue();`
isSelectedIndex()	Returns true if specified index is selected. `boolean b = <list-name>.isSelectedIndex(int index);`

Figure 3.13 Properties and methods of the jList

We used JOptionPane when we wanted to request information from the user, display information to the user or a combination of both. It required an import statement at the top of the program. Well can you recollect the import statement? That's right it is:

```
import javax.swing.JOptionPane;
```

OR

```
import javax.swing.*;
```

Either of them is acceptable. The difference is that the latter will import the entire library as denoted by the star whereas the first statement will just import the JOptionPane library.

Method	Description
showMessageDialog()	Shows a one-button, modal dialog box that gives the user some information. Example : `JOptionPane.showMessageDialog(this,"Java and NetBeans");`

showConfirmDialog()	Shows a three-button modal dialog that asks the user a question. User can respond by pressing any of the suitable buttons.
	Example:
	`Confirm=`
	`JOptionPane.showConfirmDialog(null,"quit?")`

showInputDialog()	Shows a modal dialog that prompts the user for input. It prompts the user with a text box in which the user can enter the relevant input.
	Example :
	`name=`
	`JOptionPane.showInputDialog(this,"Name:");`

Figure 3.14 Properties and methods of the JOptionPane

Variables

Let us try and recollect why the need for variables arose. Well, we used variables when we required containers to store the values for some input, intermediate result or the final result of an operation. The characteristics of a variable are:

- It has a name.
- It is capable of storing values.
- It provides temporary storage.
- It is capable of changing its value during program execution.

Variables help us to hold value for some input coming from the user or to hold intermediate result of some calculation or the final result of an operation. In other words, variables are like containers that can be used to store whatever values are needed for a specific computation. However, as different materials require different containers, and so we used different data types to hold different values.

Java programming language requires that all variables must first be declared before they can be used.

When programming, we store the variables in our computer's memory, but the computer has to know what kind of data we want to store in them, since it is not going to occupy the same amount of memory to store a simple number or to store a single letter or a large number, and they are not going to be interpreted the same way so variables were used along with datatypes. The data types supported by java are summarized as follows:

Data type states the way the values of that type are stored, the operations that can be done on that type, and the range for that type.

Numeric Data Types :

These data types are used to store integer values only i.e. whole numbers only. The storage size and range is listed below :

Name	Size	Range	Example
byte	1 byte(8 bits)	-128 to 127(-2^7 to $+(2^7-1)$)	byte rollno;
short	2 bytes(16 bits)	-32768 to 32767(-2^{15} to $+(2^{15}-1)$)	short rate;
int	4 bytes(32 bits)	-2^{31} to $+(2^{31}-1)$	int num1;
long	8 bytes (64 bits)	-2^{63} to $+(2^{63}-1)$	long amount;

Figure 3.15 Storage size and range of numeric data types

Floating Data Types:

These data types are used to store numbers having decimal points i.e. they can store numbers having fractional values.

Name	Description	Size	Range	Example
float	Single precision floating point	4 bytes (32 bits)	(3.4×10^{-38}) to $+(3.4\times10^{38})$	float average;
double	Double precision floating point	8 bytes (64 bits)	(1.8×10^{-308}) to $+(1.8\times10^{308})$	double principal;

Figure 3.16 Storage size and range of floating data types

The decision about which numeric data type to use should be based on the range of values that a variable can take.

Character Data Types:

These data types are used to store characters. Character data types can store any type of values - numbers, characters and special characters. When we want to store a single character, we use char data type and when we want to store a group of characters we use string data type. For example to store grades (A, B, C, D, E) of a student we will use char type but to store name of a student, we will use string type. The char data type value is always enclosed inside '' (single quotes), whereas a string data type value is enclosed in "" (double quotes).

Operators are symbols that manipulate, combine or compare variables.

Operators

With the introduction of variables and constants there arose a need to perform certain operations on them. We performed operations on variables and constants using operators. The operators available in java are summarized below:

Assignment Operator :

One of the most common operator is the assignment operator "=" which is used to assign a value to a variable. We assign the value given on the right hand side to the variable specified on the left hand side. The value on the right hand side can be a number or an arithmetic expression. For example:

```
int sum = 0;
int prime = 4*5;
```

Arithmetic Operators :

These operators perform addition, subtraction, multiplication, and division. These symbols are similar to mathematical symbols. The only symbol that is different is "%", which divides one operand by another and returns the remainder as its result.

```
+ additive operator
- subtraction operator
```

```
* multiplication operator
/ division operator
% remainder operator
```

Relational Operator :

A relational operator is used to test for some kind of relation between two entities. A mathematical expression created using a relational operator forms a relational expression or a condition. The following table lists the various relational operators and their usage:

Operator	Meaning	Usage
==	equal to	Tests whether two values are equal.
!=	not equal to	Tests whether two values are unequal.
>	greater than	Tests if the value of the left expression is greater than that of the right.
<	less than	Tests if the value of the left expression is less than that of the right.
>=	greater than or equal to	Tests if the value of the left expression is greater than or equal to that of the right.
<=	less than or equal to	Tests if the value of the left expression is less than or equal to that of the right.

Figure 3.17 Relational Operators

Logical Operator :

A logical operator denotes a logical operation. Logical operators and relational operators are used together to form a complex condition. Logical operators are:

Operator	Use	Meaning
&&	a>10 && b<8	a and b are both true
||	a>10 || b<8	Either a or b is true
!	! a	a is false

Figure 3.18 Logical Operators

Unary Operators :

The unary operators perform different kind of operations on a single operand .The operations performed are increasing/decreasing a value, negating a value/ expression, or inverting a boolean value.

Symbol	Name of the Operator	Operation	Example
+	Unary plus operator	indicates positive value	`num = +1;`
-	Unary minus operator	negates an expression	`num = - num;`
++	Increment operator	increments a value by 1	`num = ++ num;`
--	Decrement operator	decrements a value by 1	`num = -- num;`

Figure 3.19 Unary Operators

Increment/Decrement Operators :

The increment/decrement (++,--) operators can be a prefix or a postfix. In a pre increment/decrement expression (++ x or -- x), an operator is applied before an operand while in a post increment/decrement expression (x++ or x --) an operator is applied after an operand. In both conditions 1 is added to the value of the variable and the result is stored back to the variable. However, in a prefix expression, value is incremented first then this new value is restored back to the variable. In postfix expression the current value is assigned to a variable then it is incremented by 1 and restored back to the original variable.

Let us now try and recollect the conversion methods that we have used in java. When a Java program receives input data from a user, it must often convert it from one form (e.g., String) into another (e.g., double or int) for processing.

Conversion methods

- To convert a string value to a number (For example, to convert the String value in a text field(jTextField1) to an int, long, float or double), we can use parse methods. Assume the following declarations:

```
String num1=jTextField1.getText();
int sum; long product;
float amount;  double simple_int;
```

Type	Example statement
int	`sum=Integer.parseInt(num1);`
int	`sum=Integer.parseInt(jTextField1.getText());`
long	`product=Long.parseLong(num1);`
long	`product=Long.parseLong(jTextField1.getText());`
float	`amount=Float.parseFloat(num1);`
float	`amount=Float.parseFloat(jTextField1.getText());`

double	`simple_int=Double.parseDouble(num1);`
	OR
	`simple_int=Double.parseDouble(jTextField1.getText());`

Figure 3.20 Usage of Parse Methods

- To convert a number to string we used **valueOf** method. Assume the following code:

```
int i    = 100;
float f  = (float) 200.0;
double d = 400.0;
long l   = 100000;
string str1 = String.valueOf(i);
string str2 = String.valueOf(f);
string str3 = String.valueOf(d);
string str4 = String.valueOf(l);
jTextField1.setText("Values of int, float, double and long
are "+str1+", "+str2+", "+str3+" and "+str4);
```

Figure 3.21 Usage of valueOf Method

- To convert a number to string, we also used **toString** method. Assume the following code :

```
int i    = 100;
float f  = (float) 200.0;
double d = 400.0;
long l   = 100000;
string str1 = Integer.toString(i);
string str2 = Float.toString(f);
string str3 = Double.toString(d);
string str4 = Long.toString(l);
jTextField1.setText("Values of int, float, double and long
are "+str1+", "+str2+", "+str3+" and "+str4);
```

Figure 3.22 Usage of toString Method

- To convert a number to string, we also used concatenation operator(+). If either operand of a concatenation is a string, the other operand is converted to string. Assume the following code :

```
int sum = 103;
String convertsum;
convertsum = "" + sum;
    // Converts int 103 to String "103"
convertsum = sum + " is a number";
    // Assigns "103 is a number" to convertsum.
convertsum = "" + 10.9;
    // Assigns "10.9" to convertsum
convertsum = "" + 10.0/3.0;
    // Assigns "3.3333333333333333" to convertsum
```

Figure 3.23 Concatenation operator

Control Structures

We used control structures when we wanted to control the flow of the program. We learnt two types of control structures in class XI namely, Selection statements and Iteration statements.

Control structures allow us to control the flow of our program's execution. If left unchecked by control-flow statements, a program's logic will flow through statements from top to bottom. We can have some control on the flow of a program by using operators to regulate precedence of operations, but control structures provide the power to change statement order and govern the flow of control in a program.

Selection Statements:

A selection statement selects among a set of statements depending on the value of a controlling expression. The selection statements are the if statement and the switch statement, which are discussed below:

Simple if Statement - The if statement allows selection (decision making) depending upon the outcome of a condition. If the condition evaluates to true then the statement immediately following if will be executed and otherwise if the condition evaluates to false then the statements following the else clause will be executed. The selection statements are also called conditional statements or decision statements.

The syntax of if statement is as shown below:

Syntax:

```
if (conditional expression)
{
    Statement Block;
}
else
{
    Statement Block;
}
```

Points to remember about if statement :

- The conditional expression is always enclosed in parenthesis.
- The conditional expression may be a simple expression or a compound expression.
- Each statement block may have a single or multiple statements to be executed. In case there is a single statement to be executed then it is not mandatory to enclose it in curly braces ({}) but if there are multiple statements then they must be enclosed in curly braces ({}).
- The else clause is optional and needs to be included only when some action is to be taken if the test condition evaluates to false.

Nested if . . . else - These control structures are used to test for multiple conditions as against the simple if statement which can be used to test a single condition. The syntax of nested if else is as follows:

Syntax:

```
if (conditional expression1)
{
    statements1;
}
else if (conditional expression2)
{
    statements2;
}
else if (conditional expression3)
{
    statements3;
}
else
{
    statements4;
}
```

Now let us design a form as shown in figure 3.24

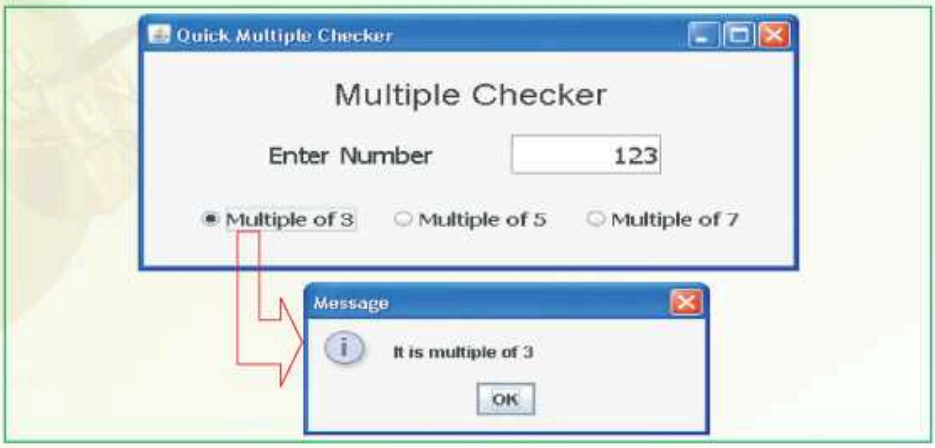


Figure 3.24 To check whether the given number is a multiple of 3,5 or 7

Follow the steps enumerated below to design the form:

1. Add a new JFrame Form and change its title property to Quick Multiple Checker.
2. Add a label set its Text as Enter Number
3. Add a jTextField and set its initial Text as ""
4. Add three radio buttons on the form - Set the text of each of them as "Multiple of 3", "Multiple of 5" and "Multiple of 7".Group the radio buttons so that only one can be selected.

In this application we ask the user to enter a number and then the user will select one of the radio buttons and depending upon the button selected the multiple check for that number will be performed. Let us now write the code for the above mentioned application. Code to check for multiple of 3 is given. Try and write the code to perform a similar check for multiple of 5 and multiple of 7.

```
private void
jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
  double Number=Double.parseDouble(jTextField1.getText());
  if (Number%3==0)
    JOptionPane.showMessageDialog(this,"It is multiple of 3");
  else
    JOptionPane.showMessageDialog
                  (this,"It is NOT multiple of 3");
}
```

Figure 3.25 Code for Multiple Checker

Let us now understand this code in detail.

`Double.parseDouble(jTextField1.getText())`

- retrieves the value entered by the user in the text field using getText(). This value by default is treated as a string and not as a number so it needs to be converted to a double type and this is achieved using the parseDouble() method. The value is then stored in the variable Number.

`if (Number%3=0)`

- check whether the number when divided by 3 gives 0 as the remainder. If the remainder is zero then the number is divisible by 3 else it is not.

Switch Statement - This selection statement allows us to test the value of an expression with a series of character or integer values. On finding a matching value the control jumps to the statement pertaining to that value and the statement is executed, till the break statement is encountered or the end of switch is reached. The expression must either evaluate to an integer value or a character value. It cannot be a string or a real number. The syntax of the switch statement is as follows:

```
switch (Variable/Expression)
{
    case Value1:statements1 ;
                break ;
    case Value2:statements2 ;
                break ;
                .
                .
    default:statements3 ;
}
```

Comparing Switch and If..else Statements - Switch is used to select sections of code depending on specific integer or character values. If we are handling specific coded values (eg, the number of the button that was clicked in a JOptionPane), or processing characters(whose codes are treated like numbers), then switch is useful. The limitations of switch are as follows:

- It doesn't allow ranges, eg case 90-100.
- It requires either integers or characters and doesn't allow useful types like String.

```
String comment; // The generated result.
int choice = Integer.ParseInt(jTextField.getText);
                //Enter 0, 1, or 2.
switch (choice)
{
    case 0: comment = "You look so much better than usual.";
            break;
    case 1: comment = "Your work is up to its usual standards.";
            break;
```

```
        case 2: comment =
            "You're quite competent for so little experience.";
                break;
        default: comment =
            "Oops -- something is wrong with this code.";
    }
```

Equivalent if statement

```
String comment; // The generated result.
int choice= Integer.ParseInt(jTextField.getText);
          //Enter is 0, 1,or 2.
if (choice == 0)
     comment = "You look so much better than usual.";
else if (choice == 1)
     comment = "Your work is up to its usual standards.";
else if (choice == 2)
  comment="You're quite competent for so little experience";
else
  comment = "Oops -- something is wrong with this code.";
```

A switch statement can often be rewritten as an if statement. Let us look at the example given above, when a selection is to be made based on a single value, the switch statement is generally easier to read. The switch is useful when you need to manage a lot of **if /else if / else.** It has a shorter syntax and is more appropriate in this case.

Let us now design an application in which we will calculate the selling price depending upon the profit percent selected by the user. Design the application as shown in figure 3.26. Set the relevant properties of the components.

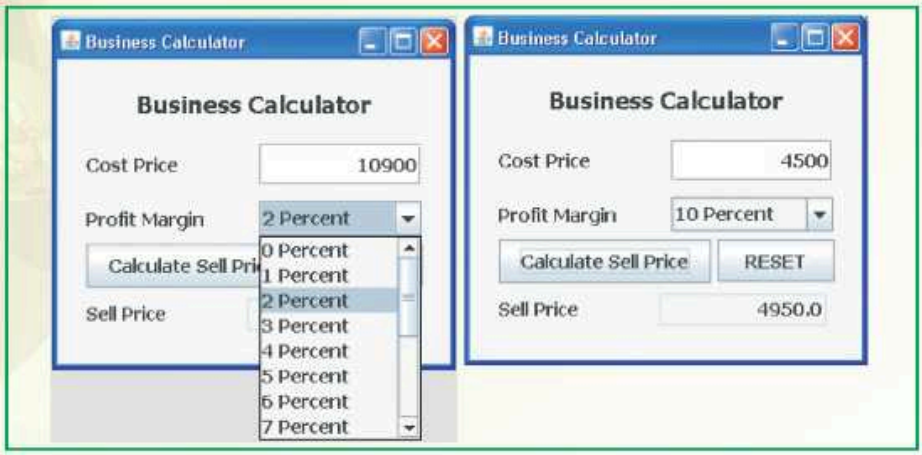


Figure 3.26 To calculate selling price depending upon profit margin

In this application depending upon the profit percent selected by the user we will calculate the selling price .Let us now write the code.

```
//Business Calculator
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    double CP,Profit=0,SP;
    CP=Double.parseDouble(jTextField1.getText());
    Profit=jComboBox1.getSelectedIndex();
    SP=CP+CP*(Profit/100);
    jTextField2.setText(Double.toString(SP));
}
```

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    jTextField1.setText("");
    jComboBox1.setSelectedIndex(0);
    jTextField2.setText("");
}
```

Figure 3.27 Code to calculate selling price

Let us now understand the code in detail.

`Double.parseDouble(jTextField1.getText())`

- retrieves the value entered by the user in the text field using getText().This value by default is treated as a string and not as a number so it needs to be converted to a double type and this is achieved using the parseDouble() method. The value is then stored in the variable CP.

`jComboBox.getSelectedIndex()`

- retrieves the index of the selected item so if the user selects 1st item profit is 0 and if the user selects 3rd, item profit is 2.This value is saved in a variable profit.

`SP=CP+CP*(Profit/100)`

- Formula to calculate selling Price.

`jTextField2.setText(Double.toString(SP)`

- The variable SP is a numeric value so it is converted to a string using the toString() method and then the value is displayed in the text field using the setText() method.

In the above application the profit margin is obtained from the Index value of the selected item of the combo box but what happens if the values in the combobox are as shown in the figure 3.28.

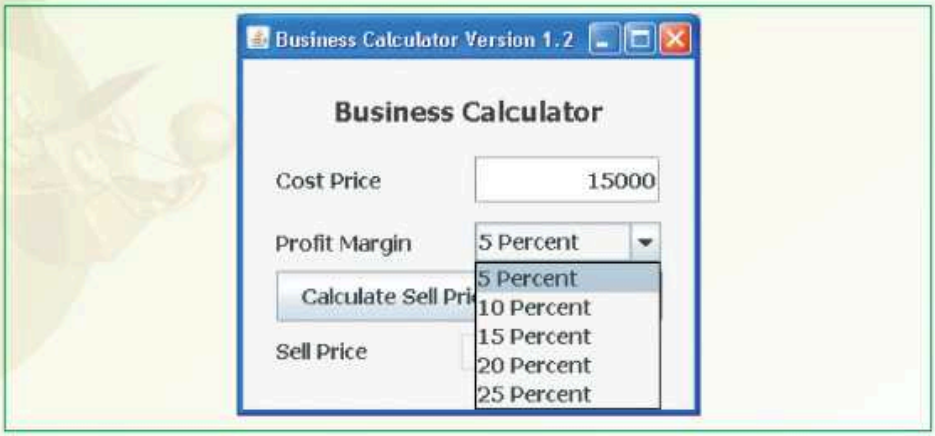


Figure 3.28 Business Calculator version 1.2

We will use switch case to write the code. Only the code using switch case is given below but it is recommended that you think and try writing the code on your own.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    double CP,Profit=0,SP;
    CP=Double.parseDouble(jTextField1.getText());
    switch (jComboBox1.getSelectedIndex())
    {
   case 0:Profit=5;break;//if first item selected profit is 5
   case 1:Profit=10;break;//if second item selected profit is 10
   case 2:Profit=15;break;
   case 3:Profit=20;break;
   case 4:Profit=25;break;
   default:Profit=0;break;
}
```

Figure 3.29 Code for Business Calculator version 1.2

Iteration Statements :

These statements are used to perform a set of instructions repeatedly while the condition is true. Iteration statements are also called looping statements.

for loop - The loop has four different elements that have different purposes. These elements are:

a) **Initialization expression:** Before entering in a loop, its variables must be initialized.

b) **Test Expression:** The test expression decides whether the loop body will be executed or not. If the test condition is true, the loop body gets executed otherwise the loop is terminated.

c) **Increment/Decrement Expression:** The Increment/Decrement expression changes the value of the loop variable.

d) **The Body of the loop:** The statements, which are executed repeatedly while the test expression evaluates to true form the body of the loop.

The syntax of the for loop is:

Syntax

```
for( initialization; test exp; increment/decrement exp)
{
statements;
}
```

The three expressions inside the round braces of for loop are optional. Using this fact an infinite loop can be created as follows:

```
for (int I = 1 ;I <= 5 ;I++ )
{
    jTextArea1.setText (jTextArea1.getText()+I);
    //It will display 12345 in jTextArea1
}
```

While Loop - The while loop is an entry-controlled loop. It means that the loop condition is tested before executing the loop body. If the loop condition is initially false, for the first iteration, then loop may not execute even once. The main characteristic of the while loop is that it can be used in both cases i.e. when the number of iterations is known as well as when it is unknown. The syntax of the while loop is as follows:

Syntax

```
while(test expression)
{
    loop body
}
```

Remember that in while loop, a loop control variable should be initialized before the loop begins and the loop variable should be updated inside the body of the while loop (else it will become an endless loop).

Do..While Loop - Do..While loop is an exit-controlled loop. In the do..while loop, the test occurs at the end of the loop. This ensures that the do..while loop executes the statements included in the loop body at least once. After the first execution of the statement, it evaluates the test expression. If the expression evaluates to true, then it executes the statements of the loop body again. Like if and while statements, the condition being checked must be included between parenthesis. The while statement must end with a semicolon. The syntax of the loop is as follows:

Syntax

```
do
{
    loop body
}
while (test expression);
```

Comparing Do...While and While - The difference between do-while and while is that do-while evaluates its expression at the end of the loop instead of at the beginning. Therefore, the statements within the do block are always executed at least once. Do-while is an exit controlled loop and while is an entry controlled loop.

```
int ctr;
ctr=0;
while (ctr < 5)    // Test and Loop
{
   JOptionPane.showMessageDialog(this,ctr);
                   // Display Number
   ctr = ctr + 1;
                   // Increment LoopCounter
}
```

The same code is written using do while. In do while counter is checked at the end of loop body, therefore loop body is executed at least once.

```
int ctr = 0;
do                 // Loop Begins
{
    JOptionPane.showMessageDialog(this,ctr);
                   // Display Number
    ctr = ctr + 1;
                   // Increment LoopCounter
}
while (ctr<5);     // Testing the value of counter
```

Let us now develop a few applications which will help us to strengthen our programming concepts learnt in class XI.

Let us now design an application in which we will generate the series and its sum. Design the application as shown in figure 3.30. Set the relevant properties of the components.

Series and its Sum

1+2+3+4

2+4+6+8+10

2+7+12+17+22

Figure 3.30 Sum of Series Calculator

Let us now write the code as follows:

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    int Start,Step,Terms,Sum=0;
    Start=Integer.parseInt(jTextField1.getText());
                //Start point of series
```

```
        Step =jComboBox1.getSelectedIndex()+1;
                    //step in the series
        Terms=Integer.parseInt(jTextField2.getText());
                    // no of terms
        for (int I=1;I<=Terms;I++)
        {
            Sum+=Start;
            Start+=Step;
        }
        jTextField3.setText(Integer.toString(Sum));
    }
    private void
    jButton2ActionPerformed(java.awt.event.ActionEvent evt)
    {
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
                //Index -1 indicates no selection
        jComboBox1.setSelectedIndex(-1);
    }
```

Figure 3.31 Code to generate the series and the Sum of Series

Let us now understand the code in detail.

`Integer.parseInt(jTextField1.getText())`

- retrieves the value entered by the user in the text field using getText(). This value by default is treated as a string and not as a number so it needs to be converted to an integer type and this is achieved using the parseInt() method. The value is then stored in the variable Start.

```
jComboBox.getSelectedIndex()+1
```

- retrieves the index of the selected item so if the user selects 1st item as Step then 1 is added to the index value of the 1st item. This value is saved in a variable Step.

```
Integer.parseInt(jTextField1.getText())
```

- retrieves the value entered by the user in the text field using getText(). This value by default is treated as a string and not as a number so it needs to be converted to an integer type and this is achieved using the parseInt() method. The value is then stored in the variable Terms.

Now we have the value from where the user wants to start the series, the terms in the series and the step between numbers of the series.

```
for (int I=1;I<=Terms;I++)
{
        Sum+=Start;
        Start+=Step;
}
jTextField3.setText(Integer.toString(Sum));
```

- the for loop is executed as many times as the Terms and the Step is added to Start which in turn is added to Sum and finally the Sum of the numbers is displayed.

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jComboBox1.setSelectedIndex(-1);
```

- On clicking the reset button all the three textfields are set to "" and the setselectedIndex value of combobox is set to -1. Index -1 indicates that no item is selected.

Let us now develop an application to reverse the given number and also find out whether the number is a palindrome or not.(The pattern that we found in the phrases was that each line had a palindrome hidden in it.)

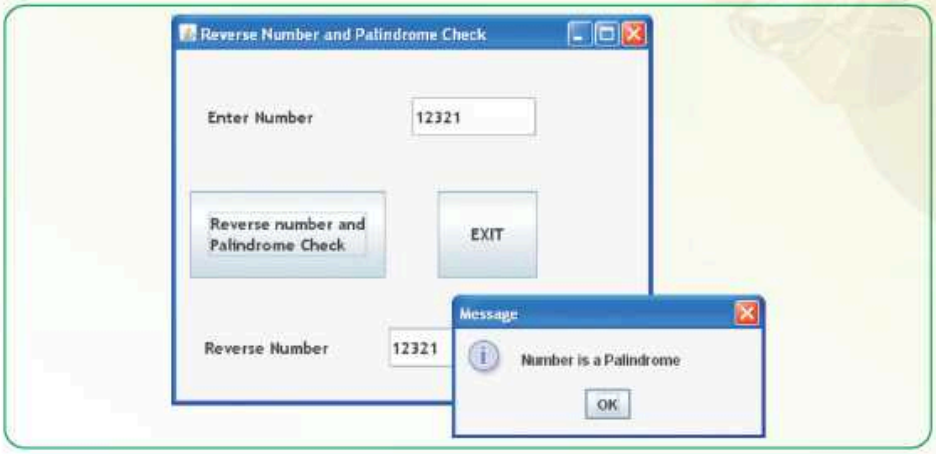


Figure 3.32 Reverse Number and Palindrome Check

Let us now write the code for reversing a given number and check whether the given number is a palindrome or not.

```
 //Reverse Number Program with Palindrome Check
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    long Number,Temp,RevNumber=0;
    Number=Long.parseLong(jTextField1.getText());
    Temp=Number;
```

```
    while (Temp>0) // till temp > 0 continue to perform the loop
    {
        RevNumber=(RevNumber*10)+(Temp%10);
        // RevNumber is multiplied by 10 and added to the
        // remainder of temp divided by 100
        Temp=Temp/10;
    }
    jTextField2.setText(Long.toString(RevNumber));
    if (Number==RevNumber)
        JOptionPane.showMessageDialog
                    (this,"Number is Palindrome");
    else
        JOptionPane.showMessageDialog
                    (this,"Number is not a Palindrome");
}
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    jTextField1.setText("");
    jTextField2.setText("");
}
```

Figure 3.33 Code for Reverse Number and Palindrome Check

Let us now understand the code in detail :

```
Long.parseLong(jTextField1.getText())
```

- retrieves the value entered by the user in the text field using getText().This value by default is treated as a string and not as a number so it needs to be converted to long type and this is achieved using the parsetLog() method. The value is then stored in the variable Number.

```
Temp=Number
```

- the number is stored in another variable temp.

```
while (Temp>0)
{
    RevNumber=(RevNumber*10)+(Temp%10);
    Temp=Temp/10;
}
```

- while loop will continue as long as temp is greater than zero. In this loop we divide temp by 10 and obtain the remainder. The remainder is then added to the value in the variable RevNumber multiplied by 10.Initially RevNumber=0.The value of variable temp is divided by 10 every time the loop is executed. The loop is executed as long as temp is greater than 0.

```
if (Number==RevNumber)
    JOptionPane.showMessageDialog
                        (this,"Number is Palindrome");
else
    JOptionPane.showMessageDialog
                        (this,"Number is not Palindrome");
```

- If the reverse of the number, which is stored in variable RevNumber and the number entered by the user which is stored in Number are the same then the number is a palindrome otherwise it is not a palindrome.

We have revised the concepts learnt in class XI. Now in the next few chapters we will enhance our knowledge of NetBeans and delve further into programming concepts.

Summary

- NetBeans is an IDE using which we can develop GUI applications in Java.
- NetBeans provides various components used to create a GUI front-end interface.
- GUI components' appearance and behaviour is controlled by their properties and methods.
- We should use meaningful names for controls on the form and variables in the code. It makes programming convenient.
- Some useful Data Types supported in Java are: int, double, char and boolean.
- String is an Object (reference) type supported in Java.
- A variable must be declared before it can be used.
- Different types of operators are available in Java. Operators are used to perform various operations on data.
- Control Statements available in java are: if..else, switch..case, for, while, do..while.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. What will be the output of the program given below. Which number is printed twice?

```
int sum1 = 3;
sum1++;
jTextField1.setText(""+sum1);
++sum1;
jTextField2.setText(""+sum1);
jTextField3.setText(""+(++sum1));
```

```
jTextField4.setText(""+sum1++);
jTextField5.setText(""+sum1);
```

a. 5 b. 6

c. 4 d. 7

2. Consider the following code snippet :

```
int anumber=14;
    if (anumber >=10){
    if (anumber == 10)
    jLabel1.setText("first string");
    else jLabel1.setText("second string");
    jLabel2.setText("third string");
```

What will be the output when anumber=14

a. first string b. second string

c. first string d. second string
third string third string

3. What value for X above will print EXACTLY 20 lines to standard output :

```
int count = 0;
while(count++ < X ) {
jTextField1.setText("Line " + count);
}
```

a. 0 b. 5

c. 10 d. 11

4. What will be the value of total after the loop finishes execution.

```
int total = 0; // must be initialized before the for (int count =
5; count <=10; count++ )
{
total += count;
}
jTextField1.setText("The total is " + total);
```

a. 10 b. 16

c. 45 d. 36

5. What's wrong with the while statement?

```
while( (ctr < 5) && (ctr > 30))
```

a the logical operator && cannot be used in a test condition.

b the while loop is an exit-condition loop.

c the test condition is always false.

d the test condition is always true.

6. If there is more than one statement in the block of a for loop, which of the following must be placed at the beginning and the ending of the loop block?

a parentheses () b French curly braces { }

c brackets [ ] d arrows < >

7. Given the following information:

int a = 11;

int b = 22;

int c = 33;

int d = 11;

Which of the following statements are true :

i) a == b ii) b != d iii) c <= b

iv) a < c v) a == d vi) c > a

vii) a >= c

a i),iv) & vii) b ii),iv), v) & vi)

c ii),iv), vi) & vii) d iii),v),vi) & vii)

8. The statement i++; is equivalent to

a i = i + i; b i = i + 1;

c i = i - 1; d i - - ;

ANSWER THE FOLLOWING QUESTIONS

1. Explain the following terms:
   a) IDE
   b) Inspector Window
   c) Form
2. Differentiate between :
   a) TextField and TextArea
   b) ComboBox and ListBox
   c) getText() and setText()
3. What is the significance of the following properties in TextArea ?
   LineWrap WrapStyleWord
4. What are list type controls used for ?
5. How would you determine whether a combo box is editable or not ?
6. List different selection modes of a list.
7. What is a button group? Which control is generally used with a buttongroup.
8. Write and explain two methods each of check box and radio button.

LAB EXERCISES

1. Design a GUI application in which the user enters a number in the text field and on clicking the button the sum of the digits of the number should be displayed in a label.

   Hint : Suppose user enters 123 the output should be 6(1+2+3).

2. Design a GUI application to accept a String from the user in a text field and print using option pane whether it is a palindrome or not.

   Hint ABBA is a palindrome.

3. Design a GUI application to accept the cost price and selling price form the user in two text fields then calculate the profit or loss incurred.

4. Design a GUI application to accept a character in a text field and print in a label if that character is a vowel: a, e, i, o, or u. The application should be case sensitive.

5. Design a GUI application that repeatedly accepts numbers in a option pane and once the typed number is 0 the maximum and minimum of all numbers typed are displayed.

6. Design a GUI application in java to convert temperature from Celsius to Fahrenheit or vice versa using radio buttons and two text fields

7. Design a GUI application in java to convert kilograms into grams, litres into milliliters, rupees into paisa using combobox and text fields.

8. A book publishing house decided to go in for computerization. The database will be maintained at the back end but you have to design the front end for the company. You have to accept book code, Title, Author and Quantity sold from the user. The Price will be generated depending upon the book code. Net price should be calculated on the basis of the discount given.

   Bookseller - 25%

   School - 20%

   Customer - 5%

9. A networking company decided to computerize its employee salary . Develop an application to store employee's personal data which will be saved in the back end. The front end should accept Name, Father's Name, Mother's Name, Address, Gender, Basic Salary, Medical and Conveyance. Calculate gross and net salary.

Basic	DA	HRA
>=40000	35%	37%
>=20000	25%	32%
>=10000	25%	30%

TEAM BASED TIME BOUND EXERCISE

(Team size recommended : 3 students in each team)

1. Students will visualize the details that have to be provided while creating a member registration for an e-reservation site. Each team has to design a layout for the form. The form has to be designed using NetBeans IDE. The team has to specify the following:
   - Controls that will be used to develop the application.
   - Data types of variables to be used.
   - Validations to be performed while accepting the data.
2. Students will visualize the details that have to be provided while developing a domicile certificate for a student of class XII. Each team has to design the form using NetBeans IDE. The team has to specify the following:
   - Controls that will be used to develop the application
   - Data types of variables to be used.
   - Validations to be performed while accepting the data.

BASICS OF OBJECT ORIENTED PROGRAMMING

Learning Objectives

After studying this lesson the students will be able to:

- Understand the need of object oriented programming
- Define the various terms related to object oriented programming
- Identify the features of an object oriented programming language
- Use features of object oriented programming language to develop simple applications

Over the last lesson, we have reviewed the core Java language. We have learnt how to work with variables and data, perform operations on that data, make decisions based on the data and also loop repeatedly over the same section of code. Now we will move on to learn a concept that is central to Java, namely Object Oriented Programming. Object Oriented Programming is a very user friendly yet a powerful approach to solve basic to difficult problems. The idea was created for developing a language that could be used for system description (for people) and system prescription (as a computer program through a compiler). There are several object-oriented languages. The three most common ones are, Smalltalk, C++ and Java.

Puzzle [1]

A student and a lady are travelling in a train. They get around talking and the lady decides to give a puzzle to the student. She tells him that she has 3 children whose product of ages is equal to the maximum number of runs possible to score in an over without any illegitimate ball being bowled (i.e. NB, Wide, etc). Also, the sum of their ages is equal to her berth number. However, the student isn't able to answer. The lady then gives him a further hint that the eldest of her children has only one eye. At this information, the

student knows the ages of the three children. Without knowing the lady's berth number, can you guess the ages of her children?

Introduction

James Alexander is a resident of a developed nation and works as a freelance consultant. He is hired by one of the corporate houses of a developing nation to plan a strategy to improve production in one of their factories which is located in a remote village named Khabri. The consultant decides to submit a quick action plan and so starts searching for information about the remote village. He has never visited any of the remote locations and so tries to simply imagine the problems faced by remote people. Mohan Swamy is a resident of one of the developing countries and he also is a freelance consultant. He completed his studies from a top notch university and to actually put his theoretical knowledge to practice, he started staying in the remote village Khabri. He wanted to actually experience the hardships faced by people residing in remote areas. To sustain himself he decides to pick up a job in the only factory situated in Khabri. The HR manager impressed with his in-depth knowledge and qualifications requests him to also plan a strategy to improve production of their factory. Who do you think will be able to provide a more viable solution? The obvious answer for most of us would be that the person sitting in the remote village and literate enough to solve the problem will be able to provide a better strategy because he closely understands the real problems of the residents as compared to a person sitting far away. But what does this teach us about programming? This teaches us that in programming also functions/methods/programs written for specific situations are able to manipulate data of their respective entities more efficiently. Now, let us understand a little about the various programming paradigms.

Introduction to Programming

Computer programming is a process of designing, writing, testing, debugging and maintaining the source code of computer programs written in a particular programming language. The purpose of programming is to organize instructions that are capable of solving real life problems. The process of writing source code of programs requires expertise in subject, knowledge of desired application domain, a formal logic and knowledge of syntax of the relevant programming language. Writing instructions in the desired order gives the required results from the program but when these instructions

increase in number, it becomes extremely difficult for the programmer to debug, maintain or troubleshoot codes. For this reason, technology experts kept developing and introducing different programming paradigms and accordingly kept developing languages to support these paradigms. Procedural programming paradigm was one of the major stepping stone for these experts which focused on breaking down a programming task into a collection of small modules known as sub routines or procedures. This paradigm helped the programmers to debug, maintain or troubleshoot codes in a more effective manner. The experts did not stop their research in improving this paradigm and introduced a new paradigm known as object oriented programming paradigm where a programming task was broken into objects. Here each object was capable of encapsulating its own data and methods (subroutines / procedures). The most important distinction is that where procedural programming uses procedures to operate on data, object oriented programming bundles data and methods together and operates as a independent entity of the program. Some languages support one particular programming paradigm while some are developed to support multiple programming paradigms. C++, C sharp, Object Pascal etc. are the languages which support procedural as well as object oriented paradigm. Java supports only object oriented programming. Basic, COBOL support only procedural programming.

Object Oriented Programming

Object Oriented Programming follows bottom up approach in program design and emphasizes on safety and security of data. It helps in wrapping up of data and methods together in a single unit which is known as data encapsulation. Object Oriented Programming allows some special features such as polymorphism and inheritance. Polymorphism allows the programmer to give a generic name to various methods or operators to minimize his memorizing of multiple names. Inheritance enables the programmer to effectively utilize already established characteristics of a class in new classes and applications.

The major components of Object Oriented Programming are as follows:

1. Class
2. Object

3. Data Members & Methods

4. Access Specifier and Visibility Modes

Classes and Objects :

A class is used to encapsulate data and methods together in a single unit. It helps the programmer to keep the data members in various visibility modes depending upon what kind of access needs to be provided in the remaining part of the application. These visibility modes are classified as private, public and protected. Usually, data members of a class are kept in private or protected visibility modes and methods are kept in the public visibility mode.

An object is an instance of a class that is capable of holding actual data in memory locations.

Class and objects are related to each other in the same way as data type and variables. For example, when we declare float variable named marks, the variable marks can be thought of as an object of type float which can be assumed as the class. If we take another hypothetical case in which Human is a class, Mr. Arun Shah, Mr. Aneek Ram will be the objects of this Human class.

Data Members and Methods :

We have already learnt that a class contains data members and methods. As discussed in the above example, Mr.Arun Shah is an object of class Human. The phone numbers retained by Mr.Arun Shah in his brain (memory) will be the data. His eyes, ears, nose and mouth can be considered as various methods which allow Mr.Arun Shah to collect, modify and delete data from his memory.

In real java programming, this data will be required to conform to a specific data type as in char, int, float or double whereas the methods will be a sequence of steps written together to perform a specific task on the data. Carefully observe the illustration given in Figure 4.1 to reinstate the theoretical concepts learnt above.

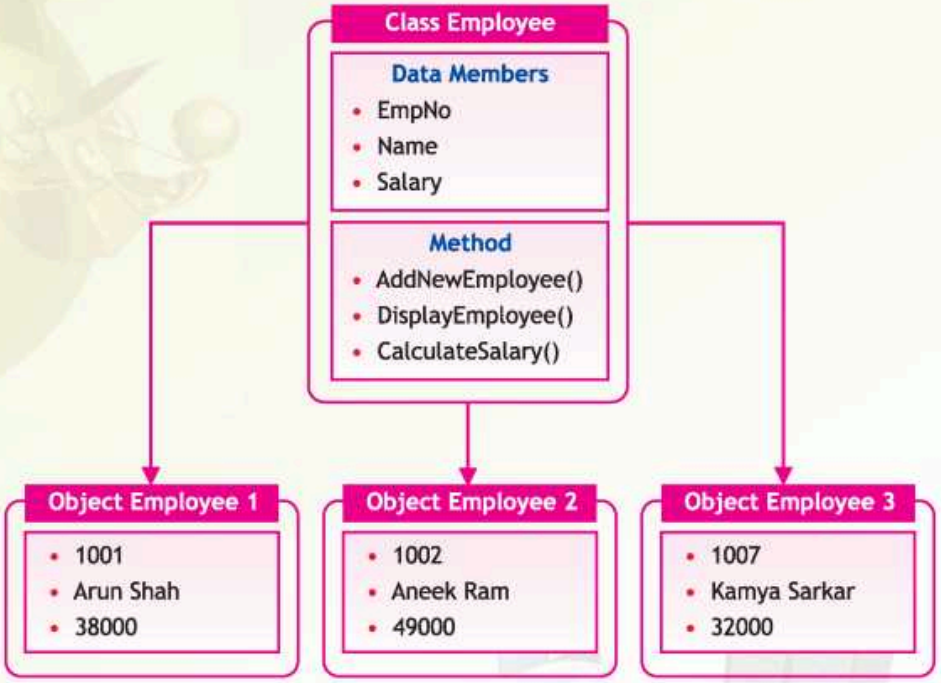


Figure 4.1 Illustration Showing the Class, Object, Members and Methods

Referring to the situation presented in the introduction of the chapter, just like the person residing in the village can efficiently solve problems pertaining to his village, similarly the methods of specific classes are able to manipulate data of their respective classes efficiently resulting in better security of data in an Object Oriented Programming paradigm.

Now that you are clear about the concept of a class and an object, you will be able to appreciate and identify classes and methods that we have already been using throughout our class XI. Do you know JTextField, JLabel, JTextArea, JButton, JCheckBox and JRadioButton are all classes and the jTextField1, jLabel1, jTextArea1, jButton1, jCheckBox1 and jRadioButton1 components are all objects. The setText(), setEnabled(), pow(), substring() are all methods of different classes. This concept is illustrated in Figure 4.2.

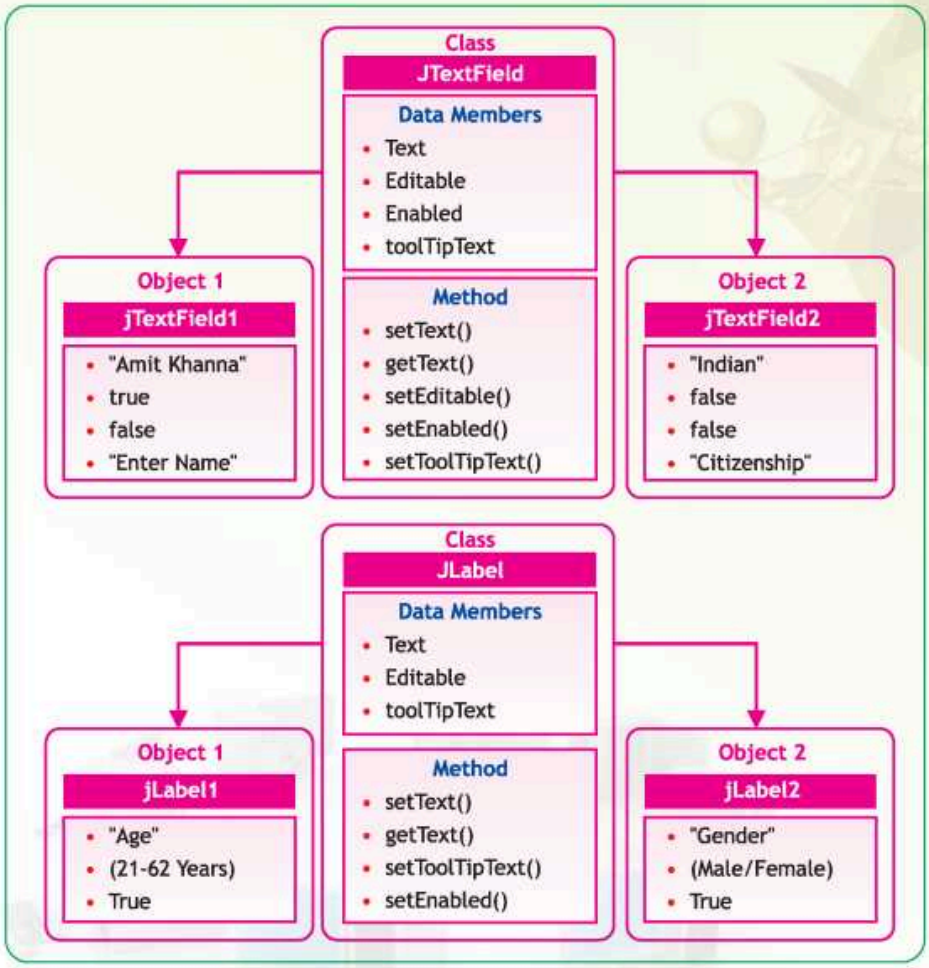


Figure 4.2 JTextField and JLabel Classes

Notice that the properties like Text, Enabled, Editable are actually the data members in the class because they store specific values as data. For example the property Text of jTextField1object contains the actual text to be displayed in the text field.

The following two tables summarize the data members (properties) and methods of all classes learnt during the course till now:

Class	Objects	Data Members (Properties)	Methods
JTextField	*jTextField1*	• *Text* • *Editable* • *Enabled* • *ToolTipText*	• *setText()* • *getText()* • *setEditable()* • *setEnabled()* • *setToolTipText()*
JLabel	*jLabel1*	• *Text* • *Enabled* • *ToolTipText*	• *setText()* • *getText()* • *setEnabled()* • *setToolTipText()*
JTextArea	*jTextArea1*	• *Columns* • *Editable* • *Font* • *lineWrap* • *Rows* • *wrapStyleWord* • *toolTipText*	• *isEditable()* • *isEnabled()* • *getText()* • *setText()*
JButton	*jButton1*	• *Background* • *Enabled* • *Font* • *Foreground* • *Text* • *Label*	• *getText()* • *setText()*

JCheckBox	jCheckBox1	• Button Group • Font • Foreground • Label • Selected • Text	• getText() • setText() • isSelected() • setSelected()
JRadioButton	jRadioButton1	• Background • Button Group • Enabled • Font • Foreground • Label • Selected • Text	• getText() • setText() • isSelected() • setSelected()
JPasswordField	jPasswordField1	• Editable • Font • Foreground • Text • Columns • toolTipText	• setEnabled() • setText() • getText() • isEnabled()
JComboBox	jComboBox1	• Background • ButtonGroup • Editable • Enabled	• getSelectedItem() • getSelectedIndex() • setModel()

		• *Font* • *Foreground* • *Model* • *SelectedIndex* • *SelectedItem* • *Text*	
JList	*jList1*	• *Background* • *Enabled* • *Font* • *Foreground* • *Model* • *SelectedIndex* • *SelectedItem* • *SelectionMode* • *Text*	• *getSelectedValue()*
JTable	*jTable1*	• *model*	• *addRow()* • *getModel()*

Polymorphism :

It is the ability of a method to execute in many forms. In object oriented programming there is a provision by which an operator or a method exhibits different characteristics depending upon different sets of input provided to it. This feature in Object Oriented Programming is known as polymorphism. Two examples of polymorphism are method overloading and operator overloading. Method overloading is where a method name can be associated with different set of arguments/parameters and method bodies in the same class. The round() method of the Math class and the substring() method of the String class are good examples of method overloading. The following examples explain how round() and substring() methods are overloaded.

If the argument passed to the round() method is of type double then it rounds off the double value and returns the closest long value. On the other hand, if the argument passed to the round() method is of type float then it rounds off the float value and returns the closest integer value. This simply means that the round() method is overloaded on the basis of the type of arguments supplied to it. Figure 4.3 illustrates the concept of method overloading as demonstrated by the round()method. For example:

```
float f = 12.5;
double d = 123.6543;
int num1 = Math.round(f);          //num1 will store 13
float num2 = Math.round(d);        //num2 will store 124.0
```

Notice that if the argument is of type double and we try to round it to an integer, then it results in an error because no such method is pre-defined which takes a double argument and rounds it to integer type.

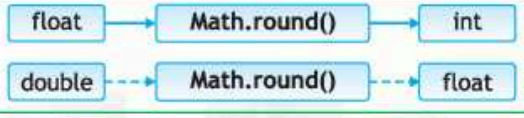


Figure 4.3 Polymorphism demonstrated by round() method

If a single argument is passed to the substring() method then it returns all characters from the start position to the end of the string whereas if two arguments are passed to the substring() method then it returns a substring of the characters between the two specified positions of the string. This simply means that the substring() method is overloaded on the basis of the number of arguments supplied to it. For example:

```
String Message = "Good Morning", Message1, Message2;
Message1 = Message.substring(5);//Message1 will store Morning
Message2 = Message.substring(0,4); //Message2 will store Good
```

Operator Overloading is another example of polymorphism. Just as methods can be overloaded, operators can also be overloaded. A very good example of operator overloading is the + operator which returns the sum of two numbers if the operands are numbers but returns the concatenated string if the operands are characters or strings. For example:

```
String A = "Hello", B = "There";
String C = A + B;              //C will store HelloThere
int num1 = 10, num2 = 20;
int num3 = num1 + num2;        //num3 will store 30
```

Abstract Class :

The classes for which objects are required to be declared are known as concrete classes like JLabel, JTextField, JComboBox etc. The classes for which it is not essential to declare objects to use them are known as abstract classes like JOptionPane. Abstract classes are normally used as base class in inheritance for which no direct object is required to be created. Also abstract classes are used for defining generic methods where there is no requirement of storing results.

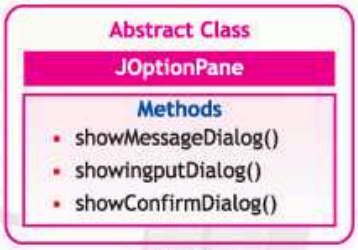


Figure 4.4
Example of an Abstract Class

Notice that the JOptionPane Class which is an abstract class has no data members and so it will not be able to store any values. The reason behind this is that because we cannot create objects of an abstract class, so we will not be able to provide any data to this class. Therefore, there is no point of having a data member.

Inheritance :

Inheritance is the most powerful feature of Object Oriented Programming, after classes themselves. Inheritance is a process of creating new class (derived class or sub class or child class) from existing class (base class or super class or parent class). The derived classes not only inherit capabilities of the base class but also can add new features of

their own. The process of Inheritance does not affect the base class. Observe the base and the derived class shown in Figure 4.5 carefully. Notice that the derived class inherits the data members namely Colour and Height from the base class and has a new data member defined in itself namely the Type. Similarly the derived class inherits the methods namely getColour() and getHeight() from the base class and has a new method defined in itself namely the setType().

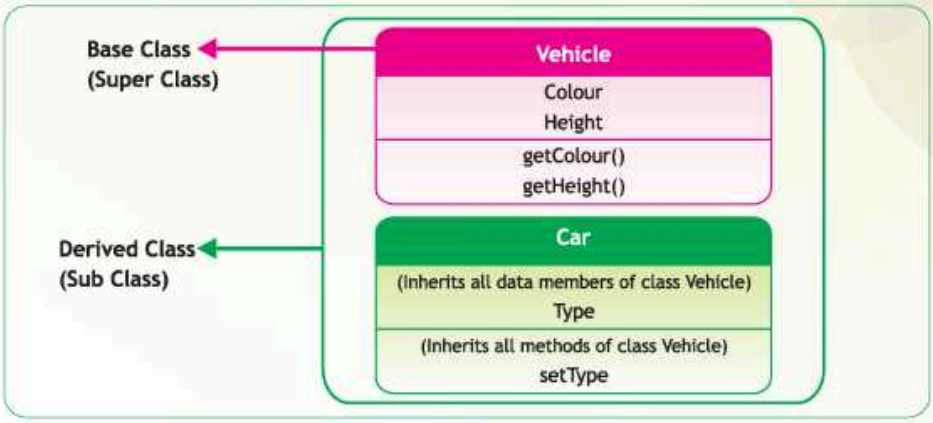


Figure 4.5 Concept of Base Class and Derived Class

The most important aspect of inheritance is that it allows reusability of code, and also a debugged class can be adapted to work in different situations. Reusability of code saves money as well as time and increases program reliability. Inheritance is very useful in original conceptualization and design of a programming problem. The idea of inheritance helps you to include features of already existing class (debugged) in a new class. Properties (data members) and methods included in super class can be invoked/accessed from the subclass. A subclass inherits all the members (data members and methods) from its superclass.

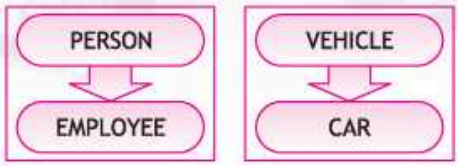


Figure 4.6 Common Examples of Inheritance

The new class can, in turn, can serve as the basis for another class definition. It is important to know that all Java objects use inheritance and every Java object can trace back up the inheritance tree to the generic class Object.

If you look at the pre-fabricated code (code automatically added by the compiler) in any of your application (JFrame Form) created in NetBeans, you will observe that javax.swing.JFrame acts as a base class (super class) and the name given by you for the JFrame Form acts as the name for the derived class (sub class). Each new JFrame Form added to the application inherits all the features of the javax.swing.JFrame class. The screen shot from one such application is shown below to illustrate the same. Here, javax.swing.JFrame is the base class and Example is the derived class.

```
public class Example extends javax.swing.JFrame
{
        /** Creates new form named Example */
        public Example()
    {
        initComponents();
    }
}
```

Figure 4.7 Sample Code Illustrating the Concept of Inheritance in Java

Notice that the extends keyword is used to inherit data members and methods of the JFrame base class for the Example class which is the derived class.

Summary

- Procedural programming paradigm focuses on breaking down a programming task into a collection of small modules known as sub routines or procedures.
- The most important distinction between Procedural and Object Oriented Programming is that where Procedural Programming paradigm uses procedures to operate on data, Object Oriented Programming paradigm bundles data and methods together and operates as an independent entity of the program.
- Class, object, data members and methods are the major components of an Object Oriented Programming paradigm.
- A class is used to encapsulate data and methods together in a single unit and an object is an instance of a class.
- Polymorphism is the ability of a method to execute in many forms.
- Method overloading and operator overloading are two examples of Polymorphism.
- The round() method of the Math class and the substring() method of the String class are good examples of method overloading.
- The + operator is a good example of operator overloading in Java.
- The classes for which it is not essential to declare objects to use them are known as abstract classes. JOptionPane is one of the examples of an abstract class.
- The concept of Inheritance allows reusability of code by including features of already existing class (called the base class) in a new class (called the derived class).
- The extends keyword is used to inherit data members and methods of the base class and allows the derived class to use these methods.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. Which of the following is not a feature of Object Oriented Programming?
   a. Inheritance
   b. Data Overloading
   c. Polymorphism
   d. Objects
2. Identify which of the following are not valid examples of method overloading?
   a. round()
   b. subString()
   c. subStr()
   d. getText()
3. Which of the following statements about the Java language is true?
   a. Both procedural and OOP are supported in Java
   b. Java supports only procedural approach
   c. Java supports only OOP approach
   d. None of the above
4. Which of the following is an Abstract class?
   a. JTextArea
   b. String
   c. Math
   d. JFrame
5. Which of the following statements is false about objects?
   a. Object is an instance of a class
   b. Object is capable of storing data
   c. Each object has its own copy of methods
   d. None of the above
6. A class can have many methods with the same name as long as the number of parameters or type of parameters is different. This OOP concept is known as
   a. Method Overriding
   b. Method Overloading
   c. Method Invocating
   d. Method Labelling

7. Which of the following statement is true?

   a. A super class is a subset of a sub class

   b. class Second extends First means First is a sub class

   c. class Second extends First means Second is a super class

   d. None of the above

8. Which feature(s) of Object Oriented Programming is illustrated in the following code snippet:

```
String A = "Hello", B = "There", C;
String C = A + B;
int num1 = 10, num2 = 20;
int num3 = num1 + num2;
C = B.substring(1);
B = A.substring(3);
```

   a. Method Overloading b. Inheritance

   c. Operator Overloading d. None of the above

ANSWER THE FOLLOWING QUESTIONS

1. Define the term Polymorphism. What are the two ways polymorphism is demonstrated in Java?
2. What is the importance of abstract classes in programming?
3. Write a short note on Operator overloading and Method Overloading citing examples for both.
4. Carefully study the code given below. It is giving an error whenever it is compiled:

```
float f = 12.5;
double d = 123.6543;
```

```
int num1 = Math.round(f);          //Statement 1
float num2 = Math.round(d);        //Statement 2
int num2 = Math.round(d);          // Statement 3
```

Identify the statement that will result in an error. Justify.

5. Carefully study the code given below:

```
String Message = "Hello! How are you?", Msg1, Msg2;
Msg1 = Message.substring(7);
Msg2 = Message.substring(0,5);
```

What will be the contents of the variables Msg1 and Msg2 after the above statements are executed?

6. Study the following code and answer the questions that follow:

```
String SMS=jTextArea1.getText();
int L=SMS.length(),Balance;
Balance=160-L;
jTextField2.setText(Integer.toString(L));
jTextField3.setText(Integer.toString(Balance));
```

a. Name any one native class of Java used in the above code.

b. Name the object created of the above mentioned native class.

c. Identify and name two methods of the native class.

d. Name the method used to convert one type of data to another and also mention the data type before and after conversion.

7. Study the following code and answer the questions that follow:

```
public class Example extends javax.swing.JFrame
{
    /** Creates new form named Example */
```

```
        public Example()
        {
            initComponents();
        }
}
```

a. Which feature of object oriented programming is depicted above?

b. Name the base class and the derived class.

c. Name the keyword used for passing on characteristics of the base class to derived class.

8. Compare and contrast the Procedural Programming paradigm and the Object Oriented Programming paradigm by writing a simple program of a mathematical calculator using both approaches. (Note: The teacher may illustrate the Procedural Programming Paradigm using any other simple programming language but should not test the students on it).

LAB EXERCISES†

1. Create a GUI application to accept a string and display it in reverse order using the substring() method.

2. Create a GUI application to create random whole numbers between 2 float numbers input by the user.

3. Create a GUI application to accept 3 numbers in separate text fields and display their sum, average, maximum or minimum after rounding the results on the click of appropriate buttons (There are four separate buttons - one for sum, one for average, one for maximum and one for minimum). The result should be displayed in the fourth text field.

4. Create a GUI application to accept the date (as 1), month (as a number like 3 for March) and year (as 2010) in separate text fields and display the date in the format: dd/mm/yy. Take care of the following points while creating the application:

† The students should be encouraged to design appropriate forms for the applications themselves.

- Verify the date input before displaying it in the suggested format and display error messages wherever applicable
- The date is accepted as 1 (for the first of any month) but should be displayed as 01 in the final format.
- The year is accepted as 2010 but displayed as 10 in the final format.

TEAM BASED TIME BOUND EXERCISES

(Team size recommended: 3 students each team)

1. Divide the class into 6 groups. Assign one class (out of JTextField, JTextArea, JLabel, JCheckbox, JRadioButton and JComboBox) to each of the groups and instruct each group to create applications demonstrating the usage of all methods learnt of that particular class. The groups may also enhance the forms using different properties of the assigned classes.
2. Divide the class into three groups and tell each group to create presentations on one of the following topics (The topics may be allocated using a draw system):
   a) Programming Paradigms
   b) The Philosophy of Object Oriented Programming
   c) Future Trends in Programming

ADVANCED PROGRAMMING CONCEPTS

Learning Objectives

After studying this lesson the students will be able to:

- Define objects and their usage
- Appreciate the usage of native classes Math and String
- Understand the need and use of methods pow() and round() of Math class
- Understand the need and use of methods toUpperCase(), toLowerCase(),substring(), concat(), length() and trim() of String class
- Develop small applications involving the methods learnt of Math and String classes.

In the last lesson, we introduced the concept of Object Oriented Programming and learnt about the different elements of an Object Oriented Programming Language. Now we will move on to learn about two important classes we commonly use in Java - namely Math and String. The lesson focuses on how to use some of the popular methods of these classes appropriately and appreciate how they simplify many programming tasks for us.

Puzzle [5]

Find a 9-digit number, which you will gradually round off starting with the number placed at units, then tens, hundreds etc., until you get to the last numeral, which you do not round off. The rounding alternates (down, up, down ...) which means that the first number from the right is rounded down while the second number from the right is rounded up and so on. After rounding off 8 times, the final number is 500000000. The original number is commensurable by 6 and 7, all the numbers from 1 to 9 are used, and after rounding four times the sum of the not rounded numerals equals 24.

Classes and Objects

As studied in the previous lesson, a class is used to encapsulate data and methods together in a single unit. An object is an instance of a class that is capable of holding actual data in memory locations. Class and objects are related to each other in the same way as data type and variables. If we take a hypothetical case in which human is a class, Mr. Shah and Mr. Kumar will be the objects of this Human class.

Math Class

The Math class contains built-in methods for performing basic numeric operations such as the elementary exponential, rounding of a number, square root, and trigonometric functions. These functions can be used directly by the user in the program. These methods are highly reliable and can tremendously reduce the amount of coding required for an application.

Some of the most commonly used Math class methods are as follows:

Method	Description
pow(double a, double b)	Returns the value of the first argument raised to the second argument.
round(double a)	Returns the closest long to the double argument.
round(float a)	Returns the closest int to the float argument.

It is not necessary to import any package for the Math class because this is already in the java.lang package. Therefore in-built methods of the Math class can be used directly in the application just like the other methods, as we have learnt in the previous class. Let us learn the usage of these methods by building some simple applications. First let us create an application that calculates the power of a number. Observe the following form carefully.

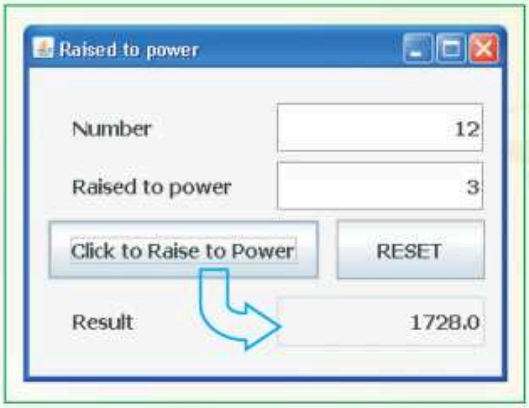


Figure 5.1 Sample Run of the Raised to Power Application

Let us first design the form as shown in Figure 5.1. First add a new JFrame form and set its title property to "Raised to Power". Now, add the following components on the form:

- Two editable text fields to accept the number and the power
- One non-editable text field to display the calculated result
- Two buttons - one to calculate & display the result and one to reset the form components
- Three appropriate labels - one against each of the text field to direct the user.

Change the properties of the components as learnt earlier so that the form looks exactly like the one displayed in Figure 5.1. The next step is to associate code with both the buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 5.2.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // Calculate the value of NumberRaise using the pow()
    function
```

```
        double Number,Raise,Result;
        Number=Double.parseDouble(jTextField1.getText());
        Raise=Double.parseDouble(jTextField2.getText());
        Result=Math.pow(Number, Raise);
        jTextField3.setText(Double.toString(Result));
    }
```

```
    private void
    jButton2ActionPerformed(java.awt.event.ActionEvent evt)
    {
        // Clear all text fields by initializing them with blank
        spaces
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
    }
```

Figure 5.2 Code for the Raised to Power Application

Let us now understand the code in detail line by line:

`double Number,Raise,Result;`

- Declare three variables named Number, Raise and Result of type double.

`Number=Double.parseDouble(jTextField1.getText()); and`

`Raise=Double.parseDouble(jTextField2.getText());`

- Retrieve the values input by the user from the text fields using the method getText() and store these values in the variables Number and Raise respectively.

```
Result=Math.pow(Number, Raise);
```

- Calculate the number (value stored in variable Number) raised to the value stored in the variable Raise using the pow() method and store the final value in the variable Result.

```
jTextField3.setText(Double.toString(Result));
```

- Display the final result in the third text field using the setText() method after converting it to string type using the toString() method.

Next, let us give a quick look to the coding of the RESET button:

```
jTextField1.setText(""); and

jTextField2.setText(""); and

jTextField3.setText("");
```

- The display text of all the three buttons is set to an empty string (i.e. blank) using the setText() method.

Next let us learn the usage of another method of the Math class namely round(). Observe the following form carefully.

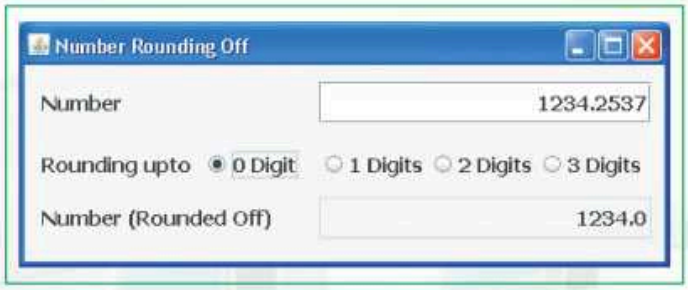


Figure 5.3 Design of the Number Rounding Off Application

Let us first design the form as shown in Figure 5.3. First add a new JFrame form and set its title property to "Number Rounding Off". Now, add the following components on the form:

- One editable text field to accept the number to be rounded.
- One non-editable text field to display the rounded off number

- Four radio buttons -to give a choice to the user for rounding off the number upto 0, 1, 2 or 3 digits
- Three labels - one against each of the text field and one against the radio button group to appropriately direct the user.

Change the properties of the components so that the form looks exactly like the one displayed in Figure 5.3. The next step is to associate code with the radio buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 5.5. A sample run of the application is shown in Figure 5.4.

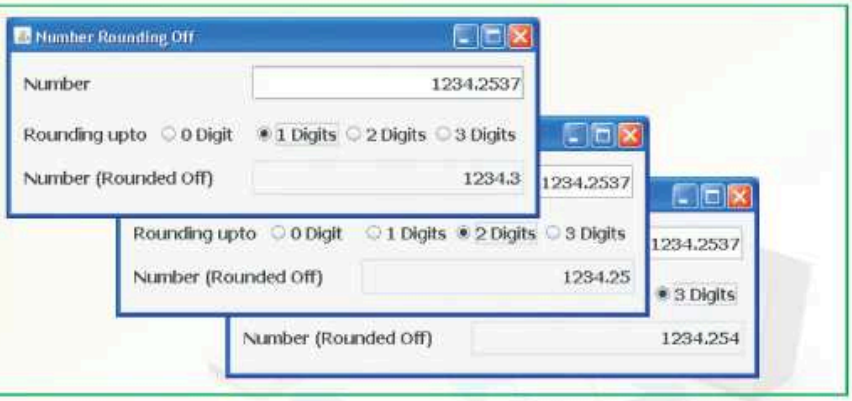


Figure 5.4 Sample Run of the Number Rounding Application

```
private void
jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ //to round off the input number
    double Number,RoundedNumber;
    Number=Double.parseDouble(jTextField1.getText());
    RoundedNumber=Math.round(Number);
    jTextField2.setText(Double.toString(RoundedNumber));
}
```

```
private void
jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ //to round off the number to 1 digit
    double Number,RoundedNumber;
    Number=Double.parseDouble(jTextField1.getText());
    //You need to divide by a real number and so 10.0 and
    not 10
    RoundedNumber=Math.round(Number*10)/10.0;
    jTextField2.setText(Double.toString(RoundedNumber));
}
```

```
private void
jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt)
{ //to round off the number to 2 digits
    double Number,RoundedNumber;
    Number=Double.parseDouble(jTextField1.getText());
    RoundedNumber=Math.round(Number*100)/100.0;
    jTextField2.setText(Double.toString(RoundedNumber));
}
```

```
private void
jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt)
{ //to round off the number to 3 digits
    double Number,RoundedNumber;
    Number=Double.parseDouble(jTextField1.getText());
    RoundedNumber=Math.round(Number*1000)/1000.0;
    jTextField2.setText(Double.toString(RoundedNumber));
}
```

Figure 5.5 Code for the Number Rounding Application

Let us now understand the code in detail line by line:

```
double Number,RoundedNumber;
```

- Declare two variables named Number and RoundedNumber of type double.

```
Number=Double.parseDouble(jTextField1.getText());
```

- Retrieve the value input by the user in the first text field using the getText() method and store it in the variable Number after converting it to type double (using the parseDouble() method)

```
RoundedNumber=Math.round(Number*10)/10.0;
```

- Calculate the rounded number using the round method and store it in the variable RoundedNumber. Since the round method does not allow the user to specify the precision digits, so we first multiply the number by 10 (for rounding off to 1 digit and similarly multiply by 100 for rounding off to 2 digits and so on) and then divide the result by 10 to get the closest double number.

```
jTextField2.setText(Double.toString(RoundedNumber));
```

- Display the calculated result in the second text field using the setText() method after converting it to type String using the toString() method.

The coding for the other radio buttons is similar.

Next let us learn a few methods of another important class, namely String class, of Java.

String Class :

The String class includes methods for examining individual characters of a string sequence, for converting strings to uppercase or lowercase, for extracting substrings, for joining two strings together, for calculating the length of a string and also for creating a new string by omitting the leading and trailing whitespaces.

When using most of the String class's methods, it should be kept in mind that a string is just a series of individual characters and that each character has a position or index, a little like a queue. Remember that in strings the first position, or index, is labelled 0 and not 1. So, if we create a string "HAPPY" then the characters will be stored as shown below:

Character Index	0	1	2	3	4
Character stored	H	A	P	P	Y

Some of the String class methods are as follows:

Method	Description
concat(String str)	Concatenates the specified string to the end of this string.
length()	Returns the length of the string.
substring (int beginpos [, int endpos])	Returns a substring of the characters between the two specified positions of the string. The second parameter is optional; if not included then all characters from the start position to the end of the string are included. The character at the ending position (n2) is not included.
toLowerCase()	Returns the string converted to lower case.
toString()	Returns the object as a string.
toUpperCase()	Returns the string converted to upper case.
trim()	Returns the string, after removing the leading and the trailing whitespaces

The most important methods are case conversion methods toUpperCase() and toLowerCase(), so let us first develop a "Case Changer" application. Observe the following form carefully:

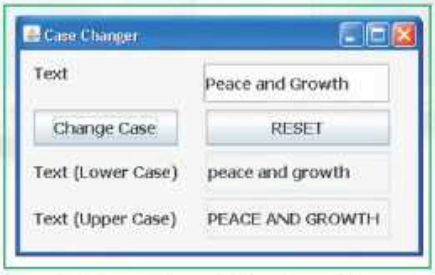


Figure 5.6 Sample Execution of the Case Changer Application

Let us first design the form as shown in Figure 5.6. First add a new JFrame form and set its title property to "Case Changer". Now, add the following components on the form:

- One editable text field to accept the string to be converted to uppercase and lowercase
- Two non-editable text fields - one to display the string converted to uppercase and the other to display the string converted to lowercase
- Two buttons - one to convert & display the converted strings and one to reset the form components
- Three appropriate labels - one against each of the text field to direct the user.

Change the properties of the components so that the form looks exactly like the one displayed in Figure 5.6. The next step is to associate code with both the buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 5.7.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
//Convert an input string to lower case and upper case
    String Str=jTextField1.getText();
    jTextField2.setText(Str.toLowerCase());
    jTextField3.setText(Str.toUpperCase());
}
```

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ // Clear all text fields by initializing them with blank
spaces
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
}
```

Figure 5.7 Code for the Case Changer Application

Let us now understand the code in detail line by line:

```
String Str=jTextField1.getText();
```

- Declare a variable named Str of type String and initialize it with the value input by the user in the first text field. The value entered in the text field is retrieved using method getText().

```
jTextField2.setText(Str.toLowerCase());
```

- Convert the string named Str to lower case using the toLowerCase() method and then display the converted string in the second text field using the setText() method.

```
jTextField3.setText(Str.toUpperCase());
```

- Convert the string named Str to upper case using the toUpperCase() method and then display the converted string in the third text field using the setText() method.

The coding of the RESET button is exactly the same as learnt in the earlier examples.

Now that we know how to use the methods of the String class, let us next learn two more new methods - one to extract specified number of characters from a string and next to join two strings together. To learn these two methods, let us design an application called

Short Name. The aim of the application is to accept the First Name, Middle Name and the Last Name from the user and display his short name (i.e Last Name followed by his initials). Observe the following form carefully.

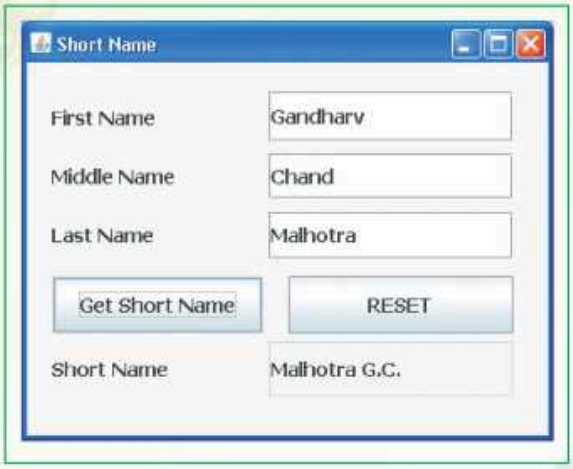


Figure 5.8 Sample Run of the Short Name Application

Let us first design the form as shown in Figure 5.8. First add a new JFrame form and set its title property to "Short Name". Now, add the following components on the form:

- Three editable text fields to accept the first, middle and last name from the user
- One non-editable text field to display the short name
- Two buttons - one to convert & display the short name and one to reset the form components
- Four labels - one against each of the text field to appropriately direct the user.

Change the properties of the components so that the form looks exactly like the one displayed in Figure 5.8. The next step is to associate code with both the buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 5.9.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // To convert a full name to a short name
    String FirstName,MiddleName,LastName,ShortName="";
    FirstName=jTextField1.getText().substring(0,1);
    MiddleName=jTextField2.getText().substring(0,1);
    LastName=jTextField3.getText();
    ShortName=ShortName.concat(LastName);
    ShortName=ShortName.concat(" "); // to add a blank space
    ShortName=ShortName.concat(FirstName);
    ShortName=ShortName.concat("."); // to add a dot to
    separate FN and MN
    ShortName=ShortName.concat(MiddleName);
    ShortName=ShortName.concat(".");
    jTextField4.setText(ShortName);
}
```

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    // Clear all text fields by initializing them with blank
    spaces
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
}
```

Figure 5.9 Code of the Short Name Application

Let us now understand the code in detail line by line:

```
String FirstName,MiddleName,LastName,ShortName="";
```

- Declare four variables of type String and initialize them with blanks (to ensure that it is an empty string).

```
FirstName=jTextField1.getText().substring(0,1); and
MiddleName=jTextField2.getText().substring(0,1);
```

- First retrieve the values of First Name and Middle Name entered in the first and second text fields using the getText() method and then extract the first character of both the First Name and Middle Name using the substring() method and store them in the variables FirstName and MiddleName respectively.

```
LastName=jTextField3.getText();
```

- Retrieve the value of the Last Name entered by the user in the third text field using getText() method and store it in the variable named LastName.

```
ShortName=ShortName.concat(LastName); and
ShortName=ShortName.concat(" "); and
ShortName=ShortName.concat(FirstName); and
ShortName=ShortName.concat("."); and
ShortName=ShortName.concat(MiddleName); and
ShortName=ShortName.concat(".");
```

- Initially the variable ShortName is blank. It is first joined together with LastName and then with a blank space after which it is joined together with variable FirstName, dot, variable Middlename and a dot again respectively using the concat() method. Finally, the content of the ShortName will be the LastName followed by the initials of the name entered by the user.

```
jTextField4.setText(ShortName);
```

- The ShortName is then displayed in the fourth text field using the setText() method.

Let us first design the form as shown in Figure 5.10. First add a new JFrame form and set its title property to "SMS Testing". Now, add the following components on the form:

- One editable text field to accept the SMS string
- Two non-editable text fields - one to display the number of characters entered by the user and the other to display how many more characters can be entered (to reach the maximum allowed length)
- Three labels - one against each of the text field to appropriately direct the user.

Change the properties of the components so that the form looks exactly like the one displayed in Figure 5.10. The next step is to associate code with the text field. Double click on the text field in the design window to reach the point in the source window where the code needs to be written. Add the code as given in Figure 5.11.

Figure 5.10 Sample Run of the SMS Testing Application

```
private void
jTextField1ActionPerformed(java.awt.event.ActionEvent evt)
{ //Display the total number of characters entered and
remaining to be
    //entered in a SMS
```

```
        String SMS=jTextField1.getText();
        int L=SMS.length();  // calculate the current length of
        the input message
        int Balance;
        Balance=160-L; // calculate the remaining no. of
        characters
        jTextField2.setText(Integer.toString(L));
        jTextField3.setText(Integer.toString(Balance));
}
```

Figure 5.11 Code for the SMS Testing Application

Let us now understand the code in detail line by line:

`String SMS=jTextField1.getText();`

- Declare a variable named SMS of type String and initialize it with the value input by the user in the first text field. The value entered in the text field is retrieved using method getText().

`int L=SMS.length(),Balance;`

- Declare two variables - Balance and L of integer type. Initialize the variable named L with the length of the SMS entered by the user. The length is calculated using the in-built method length() of the String class.

`Balance=160-L;`

- Calculate the balance number of characters by subtracting the length of the SMS entered from 160 (the maximum number of characters that can be entered)

`jTextField2.setText(Integer.toString(L));` and

`jTextField3.setText(Integer.toString(Balance));`

- Convert the variables L and Balance to type String using the toString() method and then display these values in the relevant text fields using the setText() method.

The SMS sample application developed above can further be modified with the help of a Text Area to give a better view of the SMS. Let us first design the form as shown in Figure 5.12. First add a new JFrame form and set its title property to "SMS Typist". Now, add the following components on the form:

- One editable text area to accept the SMS message
- Two non-editable text fields - one to display the number of characters entered by the user and the other to display how many more characters can be entered (to reach the maximum allowed length)
- Three labels - one against each of the two text fields and the text area to appropriately direct the user.
- Two buttons - one to reset the form components and the second to exit from the application

Change the properties of the components so that the form looks exactly like the one displayed in Figure 5.12. The next step is to associate code with the Text Area and the buttons. Double click on the text area and the two buttons, one by one, in the design window to reach the point in the source window where the code needs to be written. Add the code for the each of the three components as given in Figure 5.13.

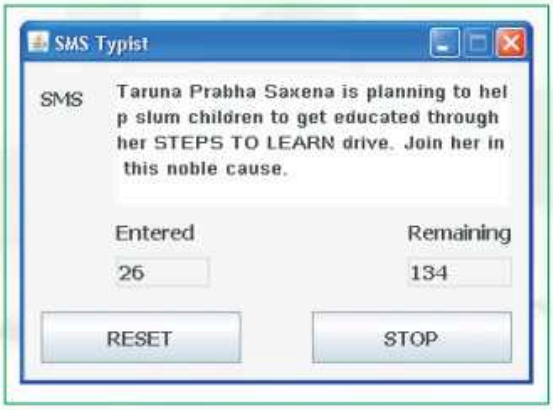


Figure 5.12 Sample Run of the SMS Typist Application

```
private void
jTextArea1CaretUpdate(javax.swing.event.CaretEvent evt)
{
    // Display the total number of characters input and
    remaining for an SMS
    String SMS=jTextArea1.getText();
    int L=SMS.length(),Balance;
    Balance=160-L;
    jTextField2.setText(Integer.toString(L));
    jTextField3.setText(Integer.toString(Balance));
}
```

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ // Clear all text fields by initializing them with blank
spaces
    jTextArea1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
}
```

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ //Exit the application
    System.exit(0);  //The parameter 0 indicates a normal
    termination
}
```

Figure 5.13 Code for the SMS Typist Application

The code for the text area is exactly same as the one written in the previous example with just a minor difference. In this case, the SMS is retrieved from the text area instead of the text field. The rest of the code is exactly the same. The code for the two buttons is also similar to all previous applications having a RESET and a STOP button.

Extra Reading

Observe that the event jTextField1ActionPerformed will display the number of characters input and left only after the user presses Enter marking the end of the input. If the user is interested in seeing the status of the number of words input and number of words that can be accepted on the spot while inputting the text, then instead of using the event jTextField1ActionPerformed(java.awt.event.ActionEvent evt), we can use the jTextField1CaretUpdate(javax.swing.event.CaretEvent evt) as shown below:

```
private void
jTextField1CaretUpdate(javax.swing.event.CaretEvent evt)
{
        String SMS=jTextField1.getText();
        int L=SMS.length(),Balance;
        Balance=160-L;
        jTextField2.setText(Integer.toString(L));
        jTextField3.setText(Integer.toString(Balance));
}
```

Note: For writing the code for jTextField1CaretUpdate Event, we need to use the Steps as shown in the following figure:

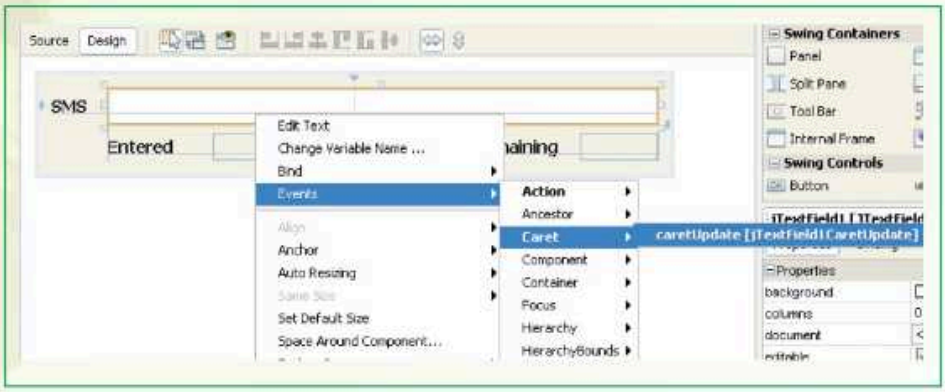


Figure 5.14 Steps for writing the code for jTextField1CaretUpdate Event

What will happen if the user enters more number of characters than the permissible range or enters unnecessary blank spaces before the beginning of the string? The SMS example can further be modified to solve the above problem. We will now develop an application which will remove the unwanted leading and trailing blanks and also truncate the SMS to the permissible range using the trim() and the substring() methods of the String class. Let us first design the form as shown in Figure 5.15. First add a new JFrame form and set its title property to "SMS Generator". Now, add the following components on the form:

- Two editable text areas - one to accept the SMS message and the second to display the final truncated SMS.
- Two non-editable text fields - one to display the number of characters entered by the user and the other to display how many more characters can be entered (to reach the maximum allowed length).
- Four labels - one against each of the two text areas and the two text fields to appropriately direct the user.
- Two buttons - one to truncate and display the SMS in the text area and the other to reset the form components.

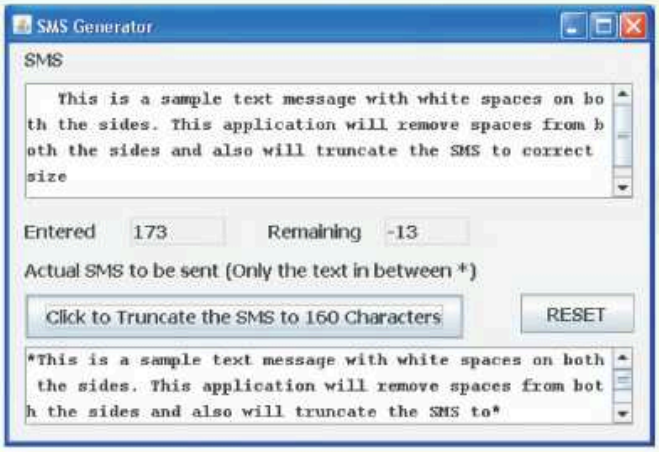


Figure 5.15 Sample Run of the SMS Generator Application

Change the properties of the components so that the form looks exactly like the one displayed in Figure 5.15. The next step is to associate code with both the buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 5.16.

Let us now understand the code in detail line by line:

```
String SMS,FinalSMS;
```

- Declare two variables named SMS and FinalSMS of type String.

```
SMS=jTextArea1.getText();
```

- Retrieve the SMS entered by the user in the first text area using the getText() method and store it in the variable SMS.

```
if (SMS.length()>160)
{
    SMS=SMS.substring(0,159);
}
```

- Check if the length of the SMS entered by the user is greater than 160 (using the length() method), then extract only first 160 characters (using substring() method) and store it in the variable SMS.

```
FinalSMS=SMS.trim();
```

- Remove White Spaces from both sides of the SMS input by the user and store the resultant SMS in the variable named Final SMS.

```
FinalSMS="*"+FinalSMS+"*";
```

- Add the character "*" on both sides of the message

```
jTextArea2.setText(FinalSMS);
```

- Display the final sms in the second text area using the setText() method.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ // Truncate the SMS to 160 characters and display it
    String SMS,FinalSMS;
    SMS=jTextArea1.getText();
    if (SMS.length()>160)
    {
        SMS=SMS.substring(0,159); // Pick up the first 160
        characters
    }
    FinalSMS=SMS.trim(); //Removes White Space from both sides
    FinalSMS="*"+FinalSMS+"*"; //Add a '*' on both sides of
    the final SMS
    jTextArea2.setText(FinalSMS);
}
```

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ //Clear all the text fields by initializing them with
blank space
    jTextArea1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextArea2.setText("");
}
```

```
private void
jTextArea1CaretUpdate(javax.swing.event.CaretEvent evt)
{ // Display the number of characters input and no. of
remaining characters
    String SMS=jTextArea1.getText();
    int L=SMS.length(),Balance;
    Balance=160-L;
    jTextField2.setText(Integer.toString(L));
    jTextField3.setText(Integer.toString(Balance));
}
```

Figure 5.16 Steps for writing the code for jTextField1CaretUpdate Event

Summary

- Java supports native classes like Math and String with predefined methods to make the programming process simpler.
- The pow(d1,d2) method of the Math class returns the value d1raised to d2.
- The round(d) method of the Math class returns the closest integer/float to d.
- The concat(str) method of the String class concatenates the string named str to the string calling the method.
- The length() method of the String class returns the number of characters in the string.
- The toLowerCase() and toUpperCase() methods of the String class return the string converted to lower case and upper case respectively.
- The substring(n1, n2) method of the String Class returns the extracted characters starting from the position n1 till n2.
- The trim() method of the String class returns the string after removing the leading and the trailing blanks.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. The _____ method is used to extract specified number of characters from a string.

   A. subStr() B. subString()

   C. takeString() D. extract()

2. Which of the following method(s) does not accept any arguments?

   A. pow() B. toLowerCase()

   C. round() D. All of the above

3. Which of the following statement is true?

   A. The trim() method is used to trim a string by removing spaces from one side

B. The trim() method is used to trim a number by removing zeroes

C. The trim() method is used to trim a string by removing spaces from both sides.

D. The trim() method is used to trim a string by removing extra characters.

4. Which of the following is not a valid method of the String class?

A. length() B. concat()

C. trim() D. round()

5. _____ is a class which contains methods for basic numerical operations like rounding off the number.

A. Number B. String

C. Math D. Integer

6. What will be the effect of executing the following code:

```
String F1="INTER",F2="MEDIATE";
F1=F1.concat(F2);
```

A. F2 will be concatenated at end of F1 and stored in F1.

B. F1 will be concatenated at end of F2 and stored in F1.

C. F1 will be concatenated at end of F2 and stored in F2.

D. F2 will be concatenated at end of F1 and stored in F2.

7. What will be the contents of L after executing the following code?

```
String F1="I#N#T#E#R",F2;   //# represents a space
F2=F1.trim();
int L=F2.length();
```

A. 5 B. 9

C. 10 D. 6

8. Which of the following is not a primitive data type in Java?

A. float B. int

C. string D. void

ANSWER THE FOLLOWING QUESTIONS

1. What is a class?
2. Name any four native classes of Java and briefly explain the use of each.
3. Give two appropriate advantages of using in-built methods over creating user defined methods.
4. Explain the general syntax of using any method of the String class. Is it in anyway different from using a method of the Math class?
5. What will be the contents of F1 and F2 after the following code is executed?

```
String F1="Hello",F2="Friend";
F1=F1.concat(F2);
```

6. Tanyaman is creating a simple application in java called "Password Checker" in which she needs to convert the characters input by the user in a particular case. Name two methods of the String class that she can use for this purpose.
7. Aryamani is creating a simple application in java called "Name Concatenator" in which he needs to concatenate the first name, middle name and last name input by the user in separate text fields. Name a method and an equivalent operator for this purpose.
8. Study the following code and answer the questions that follow:

```
double Number,RoundedNumber;
Number=Double.parseDouble(jTextField1.getText());
RoundedNumber=Math.round(Number*1000)/1000.0;
jTextField2.setText(Double.toString(RoundedNumber));
```

   a) How many variables have been declared in the above code? Identify and name them.

   b) How many objects have been declared in the above code?

   c) Name any one native class of Java used in the above code.

   d) Identify and name a method of this native class.

e) Name the method used to convert one type of data to another and also mention the data type before and after conversion.

LAB EXERCISES†

1. Create an application to accept two strings - First Name and Last name from the user and display the message Welcome with the complete name of the user.
2. Create an application to accept the radius of a circle, calculate the area and circumference and display the results in a message box after rounding off the area and circumference to an integer number.
3. Modify application to make sure that the user has not input the complete name in the first name text field.
4. Modify the Case Changer application developed in the lesson to display the input text in Title case using the substring(), toLowerCase() and toUpperCase() methods.

TEAM BASED TIME BOUND EXERCISES

(Team size recommended: 3 students each team)

1. Divide the class into groups of 4 students each and tell them to create a GUI application that allows a user to change his password according to the following guidelines:
   a. The form should accept the name and password from the user
   b. Only if the password matches with the pre-input password, the user should be allowed to proceed. (Hint - need to check the length and the contents and also convert the string to a particular case)
   c. Hide the initial text fields and now create form elements to accept input of first name, middle name and last name along with the age.
   d. Ensure that the age input is an integer and the first name, middle name and the last name are not more than 12 characters long.
2. Divide the students into groups of 6 and then give each group an algorithm which is predesigned by the teacher, and tell them to execute it. This algorithm should draw a figure with the children; the first team to find the figure and name it wins.

† The students should be encouraged to design appropriate forms for the applications themselves.

For example :

child[1].AtLeftOf(child[2])

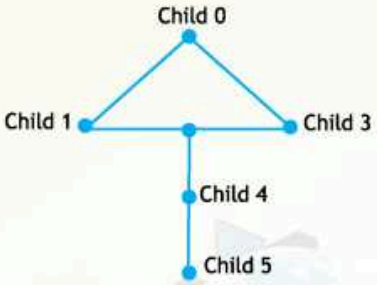
child[0].InFrontOf(child[2])

child[2].InFrontOf(child[4])

child[3].AtRightOf(child[2])

child[4]. InFrontOf(child[5])

The algorithm given above is to form the shape of an arrow as depicted in the following figure:



DATABASE CONNECTIVITY

Learning Objectives

After studying this lesson the students will be able to:

- State the need of saving data in the database
- Connect a GUI interface with data stored in database.
- Use various methods to retrieve data from a database.
- Write a complete GUI application with database connectivity
- State the advantages of developing applications in local languages

In the earlier lessons, you have learnt how to develop a GUI Interface using Netbeans and creating & manipulating data in a table of a database as two independent concepts. This lesson will help you to combine both these concepts together and help you to develop complete applications in which the GUI Interface will act as Front-End or Client Side application and Database will act as Back-End or Server-Side application.

Puzzle [6]

A person wanted to withdraw X rupees and Y paise from the bank. But the cashier made a mistake and gave him Y rupees and X paise. Neither the person nor the cashier noticed that. After spending 20 paise, the person counts the money. And to his surprise, he has double the amount he wanted to withdraw. Find X and Y.

(1 Rupee=100 paise)

Introduction

Imagine a swanky air-conditioned car showroom in prime location of the city with all modern car accessories available in an outlet within the showroom. The showroom has a

wonderful ambience, employs skilled salesperson with excellent communication skills to convince prospective buyers. They have a tie-up with various banks and financial agencies to provide lucrative payment plans to lure customers. What do you think he will sell? Do you think this showroom will be able to cover cost without selling cars? Do you think the skilled salesperson with excellent communication skills will be able to manufacture cars?

On the other hand there is a factory, as usual in a remote location, with abundant raw material, latest machinery, skilled workforce that is technically sound with latest automobile expertise. The factory keeps manufacturing cars but does not have any dealers. Factories are generally set up in remote industrial towns so do you think a remote area in Gujrat will be able to sell cars directly from their factory? Do you think the factory will be able to cover their costs incurred on raw materials without delivering cars? Do you think a skilled workforce can convince people to buy cars?

GUI Application (The Front-End)

As you know that the programmers develop applications for solving real life problems to simplify life of others. That is the reason, they try to develop applications, which require minimum inputs from the user and provide meaningful information in the form of output through their applications. The programmer develops these applications keeping in mind that the user will have a very basic understanding of handling mouse, keyboard and desktop applications. In the earlier lessons of GUI programs, you must have got a good idea of developing simple GUI Applications, in which user was typing in or clicking on certain inputs to produce meaningful results (output). If you see all these applications carefully, you will realize that none of them are capable of retaining the inputs or outputs for future reference or uses. So, typing in any number of inputs in the application carries a life span of one execution only and when the application is executed second time, it requires a new set of inputs from the user. Do you realize why it is like this? It is because; the data entered by you in a front end application was getting stored in temporary memory (RAM) and not getting saved on a hard disk (or any other secondary storage device).

Database (The Back-End)

The database helps you to save data permanently in secondary storage devices and keeps data ready for future reference, modification and addition. In the earlier lessons about database concepts, you have learnt about how to create a table, modify content of a table, add new content in the table, delete content from a table and retrieve content from a table in various forms. But you must have seen that to perform these activities, you were required to learn SQL commands like CREATE, UPDATE, INSERT, DELETE, SELECT and so on and each of these commands had their own syntax structures. Let us review this with the help of the Employee table shown below:

Table: Employee

Empno	Name	Salary
1001	Jasmine Taneja	45000
1009	Ravi Mathwad	34000
1015	Desai Ramakrishnan	58000
1004	Jigyasa Burman	49000

The SQL commands

- to add a new row in the table:

```
INSERT INTO Employee VALUES (1013, "Punya Sarthi", 51000);
```

- to retrieve the entire content from the table:

```
SELECT * FROM Employee;
```

- to increase salary by 10% for the employee whose Empno is 1009:

```
UPDATE Employee SET Salary=Salary*1.1 WHERE Empno=1009;
```

Obviously, you do not expect the user to learn all these commands to perform these activities. User always likes to access the information using the computer with some clicks of mouse or by typing in little textual input as per instructions provided preferably in his/her local language. So, there is a need to provide him/her with an interface to access or modify the information available in the database.

Database Connectivity (Front-End + Back-End)

After looking at the role of the Front-End and the Back-End, it is clear that the Car showroom in our introduction is actually our front end and the factory is nothing but the back end which supplies cars to the showroom. The showroom owner will be able to cover the costs incurred on running and maintaining the showroom only if the cars are delivered to the showroom by the factory. Similarly, the factory needs to send the cars to the showroom set up in an urban area for selling. Setting up an effective link between the two ends to facilitate the transfer of cars is the only solution of all the stated problems.

After going through the above paragraphs, you can understand that there is a requirement of creating an interface (the form), which a user may use for accessing information and also there is a requirement of keeping the information saved in the database (table) in an organized manner for future references and uses. So, we need to combine both these concepts together to develop a complete application for various domain specific problems. Now after learning about the front end and the back end, the next step is to learn how to set up a database connection that allows the front end to communicate with the back end. The two components essential to establish this connectivity are enumerated below and demonstrated in Figure 6.1:

- The JDBC API - software used to provide RDBMS access and execute SQL statements within java code.
- The JDBC Driver for MySQL - software component enabling a java application to interact with a MySQL database.

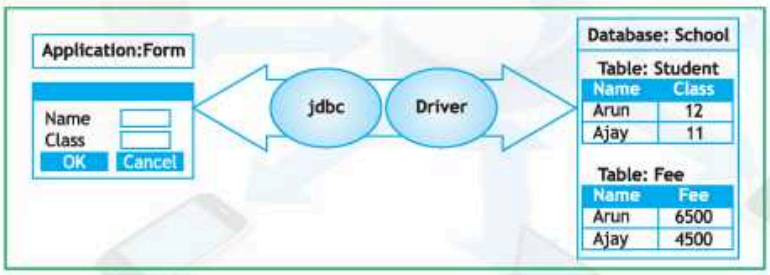


Figure 6.1 Communicating With a Database Using JDBC API and Driver

Know more

An application programming interface (API) is an interface implemented by a software program which enables it to interact with other software. It facilitates interaction between different software programs similar to the way the user interface facilitates interaction between humans and computers.

Adding [MySQL JDBC Driver] Library in NetBeans

The first step for establishing data connectivity is to add the MySQL JDBC Driver Library for use by your project. Remember, that this process has to be repeated every time you start a new project but not for new forms or applications added to an existing project. To add the MySQL JDBC Driver Library follow the given steps.

Step 1: Right click on the Project name and select the Properties option as shown in Figure 6.2.

Step 2: In the Properties dialog box,

i. choose the Libraries option from the Categories pane

ii. click on the Add Library button as shown in Figure 6.3.

iii. From the Add Library dialog box choose the MySQL JDBC Driver

iv. Click on the Create button as shown in Figure 6.3.

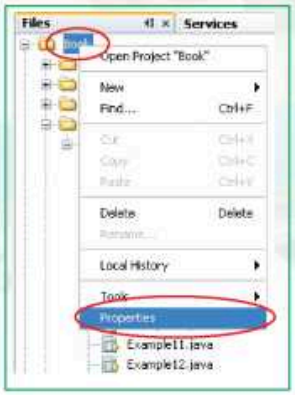


Figure 6.2 Opening Project Properties Dialog Box

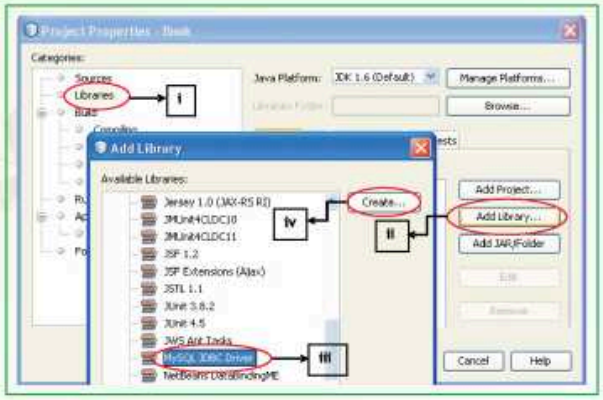


Figure 6.3 Adding the MySQL JDBC Driver From Available Libraries

Know more

The JDBC (Java Database Connectivity) API is a software for executing SQL statements. It provides RDBMS access by allowing you to embed SQL inside Java code. We can create a table, insert values into it, query the table, retrieve results, and update the table with the help of JDBC.

The driver is now added to the compile time libraries and this can be verified by clicking on the Compile Time Libraries tab in the Project Properties dialog box as shown in Figure 6.4. Click on OK to close the Project Properties dialog box.

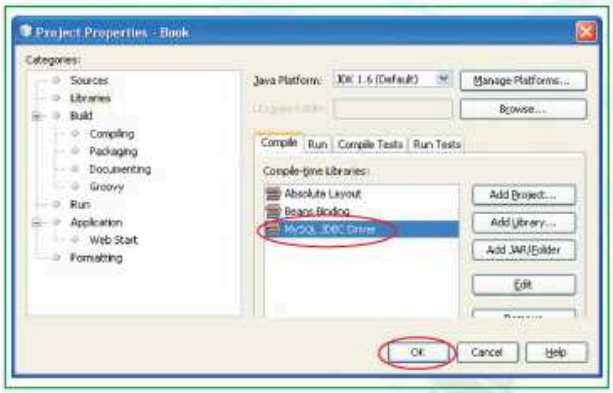


Figure 6.4 Compile-Time Libraries

Basic Libraries Required in the Application for Data Connectivity

Apart from the MySQL JDBC driver, we also need to import some basic class libraries which are essential for setting up the connection with the database and retrieve data from the database - namely DriverManager Class, Connection Class and Statement Class. These class libraries can be added using the following commands in our application code:

```
import java.sql.DriverManager;

import com.mysql.jdbc.Connection;

import com.mysql.jdbc.Statement;
```

Let us quickly understand the basic purpose for each of these class libraries:

- The JDBC DriverManager class defines objects which can connect Java applications to a JDBC driver. DriverManager is considered the backbone of JDBC architecture. DriverManager class manages the JDBC drivers that are installed on the system. Its getConnection() method is used to establish a connection to a database. It uses a username, password, and a jdbc url to establish a connection to the database and returns a connection object.
- A JDBC Connection represents a session/connection with a specific database. Connection interface defines methods for interacting with the database via the established connection. An application can have one or more connections with a single database, or it can have many connections with different databases.
- Once a connection is obtained we can interact with the database. To execute SQL statements, you need to instantiate (create) a Statement object using the connection object. A Statement object is used to send and execute SQL statements to a database.

In case any of these libraries are missing, then the execution results in an error as shown in Figure 6.5.

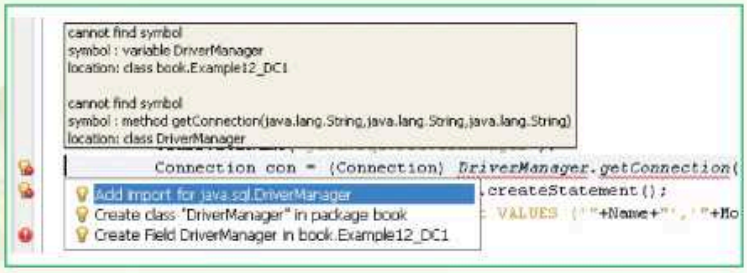


Figure 6.5 Error Window Indicating Missing DriverManager Library

Data Connectivity Application 1: Add Records in a Table Input Through a Form

Now that we have learnt the basics about establishing a connection, let us start developing an application that uses database connectivity. The aim of the first application that we will develop is to add data into a table.

Using MySQL command prompt, create a Table named "Contact" in the Database CBSE, with the specified table structure. The relevant command is shown in Figure 6.6.

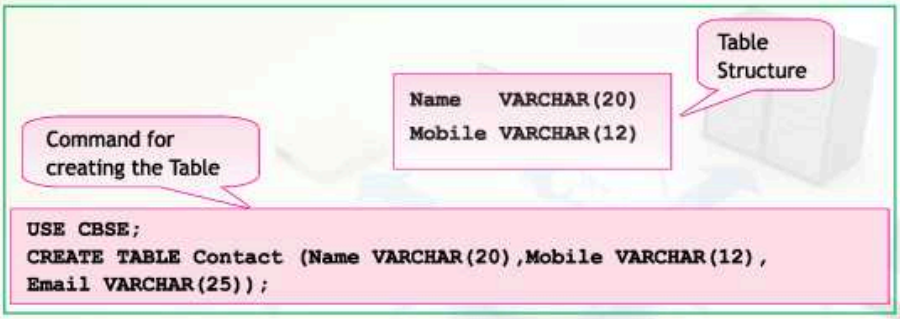


Figure 6.6 Command for creating the back end Table with a specified structure

Now let us design the form as shown in Figure 6.7. First add a new JFrame form and set its title property to "Contact List". Now, add the following components on the form:

- Three editable text fields to accept the name, mobile number and email address from the user.
- Three appropriate labels - one against each of the text field to direct the user.

- Two buttons - one to add the data supplied by the user in the database and one to exit from the application.

Change the properties of the components as learnt earlier so that the form looks exactly like the one displayed in Figure 6.7. The next step is to associate code with both the buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 6.8.

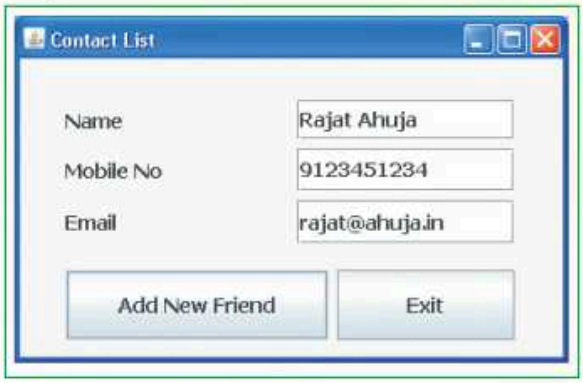


Figure 6.7 Front-end Form for the Contact List Application

```
//NetBeans Application 1 - To add new row(s) in the Contact Table
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
//Declare variables and store values retrieved from the
front end
    String Name=jTextField1.getText();
    String Mobile=jTextField2.getText();
    String Email=jTextField3.getText();
    try
    // this block is executed in case of normal execution
```

```
    {
        Class.forName("java.sql.DriverManager");
        Connection con = (Connection)
        DriverManager.getConnection
        ("jdbc:mysql://localhost:3306/cbse",
        "root", "abcd1234");
        Statement stmt = (Statement) con.createStatement();
        String query="INSERT INTO Contact VALUES
        ('"+Name+"','"+Mobile+"','"+Email+"');";
        stmt.executeUpdate(query);
    }
    catch(Exception e)
    //this block is executed in case of an exception
    {
    //Display an error message in the dialog box for an exception
        JOptionPane.showMessageDialog (this, e.getMessage());
    }
}
// This part of the code is used to exit from the application
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    System.exit(0);
}
```

Figure 6.8 Code for the Contact List Application

Know more

To catch an exception in Java, you write a try block with one or more catch clauses. Each catch clause specifies one exception type that it is prepared to handle. The try block delimits a bit of code that is under the watchful eye of the associated catchers. If the bit of code delimited by the try block throws an exception, the associated catch clauses will be examined by the Java virtual machine. If the virtual machine finds a catch clause that is prepared to handle the thrown exception, the program continues execution starting with the first statement of that catch clause. For example, in the above code if any error occurs while establishing a connection or executing the SQL query then the error message will be displayed in a message box due to the catch block immediately following the try block.

Let us now understand the code in detail line by line:

```
String Name=jTextField1.getText(); and
String Mobile=jTextField2.getText(); and
String Email=jTextField3.getText();
```

- Declare three variables named Name, Mobile and Email of type String and initialize them with the values input by the user in the text fields. The values are retrieved from the text fields using the getText() method.

```
Class.forName("java.sql.DriverManager");
```

- In this step of the jdbc connection process, we load the driver class by calling Class.forName() method with the Driver class name as an argument. Once loaded, the Driver class creates an instance of itself. A client can then connect to the Database Server through JDBC Driver.

```
Connection con = (Connection)
DriverManager.getConnection
    ("jdbc:mysql://localhost:3306/cbse", "root", "abcd1234");
```

- The getConnection() method of the DriverManager class is used to establish a connection to the cbse database. The method uses a username, password, and

a jdbc url to establish a connection to the database and returns a connection object named con.

- In the above command:

  jdbc:mysql - is the Database Driver Connection

  3306 - is the Default Port no on which MySQL runs

  cbse - is the Database Name

  root - is the User Name

  abcd1234 - is the Password

```
Statement stmt = (Statement) con.createStatement();
```

- Instantiate a Statement object called stmt from the connection object (named con created in the previous statement) by using the createStatement() method. A statement object is used to send and execute SQL statements to a database.

```
String query="INSERT INTO Contact VALUES
        ('"+Name+"','"+Mobile+"','"+Email+"');";
```

- Create a variable called query of type String and initialize it with the SQL statement to be executed (in this case the INSERT INTO statement).

```
stmt.executeUpdate(query);
```

- Execute the SQL statement stored in the variable query, using the executeUpdate() method of the Statement class. This results in adding the values stored in the three variables (Name, Mobile & Email) in the table Contact of the cbse database.

```
catch (Exception e)
{
    JOptionPane.showMessageDialog (this, e.getMessage());
}
```

- In case of an exception, retrieve the error message string using the getMessage() method and display it in a dialog box using the showMessageDialog() method.

Note that the forName() method is used to load the class specified as its argument at runtime.

Figure 6.9 shows a sample run of the above application.

Figure 6.9 Sample Run of the Contact List Application Showing Error Message

The above application can be modified to handle basic data validations as shown in Figure 6.10.

Note that Data validation is the process of ensuring that a program operates on clean, correct and useful data. It uses routines, often called "validation rules" or "check routines", that check for correctness, meaningfulness, and security of data that are input to the system. For example, telephone numbers should include the digits and possibly the characters + & -. A more sophisticated data validation routine may check to see the user has entered a valid country code, i.e., that the number of digits entered match the convention for the country or area specified.

```
//Netbeans Application 1 (with Data Validation)
//- To add new row(s) in the Contact Table
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{  String Name=jTextField1.getText();
   String Mobile=jTextField2.getText();
   String Email=jTextField3.getText();
   /* Check if any of the variable is empty and accordingly
      display
      an appropriate error message */
   if (Name.isEmpty())
      JOptionPane.showMessageDialog(this,"Name not Entered");
   else if (Mobile.isEmpty())
    JOptionPane.showMessageDialog(this,"Mobile not Entered");
  else if (Email.isEmpty())
    JOptionPane.showMessageDialog(this,"Email not Entered");
  else
   { try
     {
         Class.forName("java.sql.DriverManager");
         Connection con = (Connection)
         DriverManager.getConnection
         ("jdbc:mysql://localhost:3306/cbse", "root",
         "abcd1234");
         Statement stmt = (Statement) con.createStatement();
```

```
            String query="INSERT INTO Contact VALUES
                ('"+Name+"','"+Mobile+"','"+Email+"');";
            stmt.executeUpdate(query);
            jTextField1.setText("");
            jTextField2.setText("");
            jTextField3.setText("");
        }
        catch (Exception e)
        {
        JOptionPane.showMessageDialog  (this,  e.getMessage());
        }
}
```

Figure 6.10 Code for the Contact List Application with Data Validations

Let us now understand the purpose of the additional commands added to the above code:

```
if (Name.isEmpty())
    JOptionPane.showMessageDialog(this,"Name not Entered");
else if (Mobile.isEmpty())
    JOptionPane.showMessageDialog(this,"Mobile not Entered");
else if (Email.isEmpty())
    JOptionPane.showMessageDialog(this,"Email not Entered");
else   { ….. }
```

- Check if the value of any of the three variables (Name, Mobile, Email) is empty using the isEmpty() method with the help of the if..else conditional statement. In case any of the values is empty then display a error message in a dialog box using the showMessageDialog() method.

```
catch (Exception e)
    { JOptionPane.showMessageDialog (this, e.getMessage()); }
```

- In case of an exception, retrieve the error message string using the getMessage() method and display it in a dialog box using the showMessageDialog() method.

The code for the exit button is the same as all our previous applications.

Note that an exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

Running SQL Commands in Netbeans

After executing the Contact List Application, if you need to verify whether the row has been added to the table or not, then you need not go back to the MySQL prompt. Netbeans allows one to directly run SQL commands from the GUI interface.

Therefore, the content of the table can directly be tested by running SQL in Netbeans. To execute an SQL command in Netbeans perform the following steps:

Step 1: In the Services tab, right click on the Databases and select the New Connection option from the drop down menu.

Step 2: The New Database Connection dialog window opens up as shown in the Figure 6.11. Provide the values as shown in the figure and click OK. Clicking on OK in the above dialog window adds the connection to the existing list as shown in the Figure 6.12. Note that this step is required to be done only once - the first time when creating a connection.

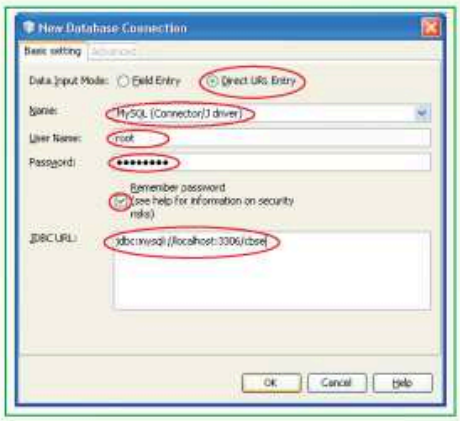


Figure 6.11 Setting Up a New Database Connection

Step 3: Right click on the newly added Connection and select the Connect option from the drop down menu. This step is to be repeated every time you start Netbeans. In the Connect dialog box that opens up, enter the user name and password and select the Remember Password checkbox (Selecting this checkbox ensures that this step is not to be repeated every time you start Netbeans) as shown in Figure 6.12.

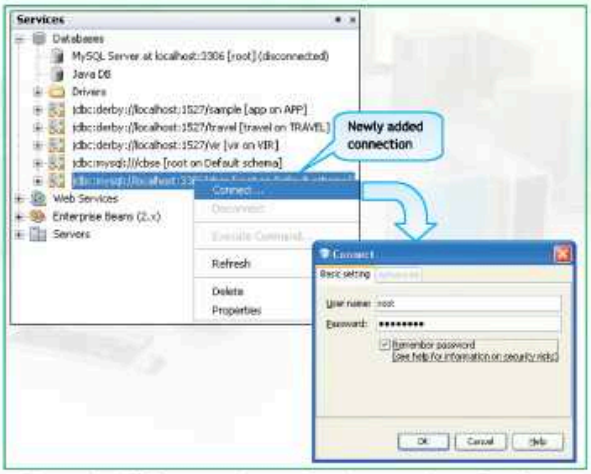


Figure 6.12 Connecting to the Appropriate Database

After establishing the connection, right click on the connection driver and select the Execute Command option from the drop down menu. The screen as shown in Figure 6.13 is displayed. Now, type in any SQL to be executed on the database and execute the command. The result is displayed in the bottom half of the window as shown in Figure 6.13.

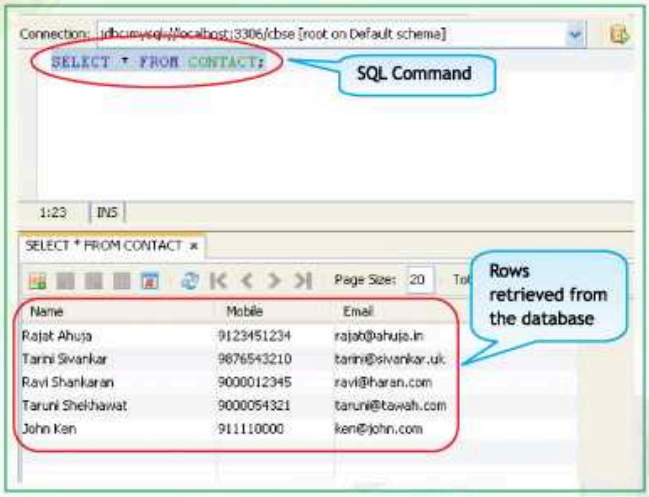


Figure 6.13 Executing SQL Command Directly in Netbeans

Data Connectivity Application 2: To Display Records Retrieved from a Database (Back End) in a Form (Front End) one by one

The next application that we will develop can be used to display data retrieved from a database in a form, one record at a time. Add a few records to the table Contact created above. Now design the form as shown in Figure 6.14. First add a new JFrame form and set its title property to "Search From Contacts". Now, add the following components on the form:

- One editable text field to accept the mobile number from the user.
- Two non-editable text fields to display the details - Name and E-mail of the contact being searched.

- Three appropriate labels - one against each of the text field to direct the user.
- Two buttons - one to search for the details on the basis of a mobile number supplied by the user and one to exit from the application.

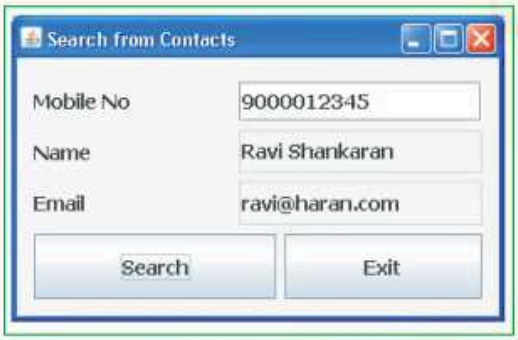


Figure 6.14 Sample Run of Search From Contacts Application

Change the properties of the components as learnt earlier so that the form looks exactly like the one displayed in Figure 6.14. The next step is to associate code with both the buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 6.15.

```
/*Netbeans Application 2 (Basic Part) - To search for a
matching row and display corresponding information from the
table Contact */
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
   // Search the table to find a record matching
   // the input Mobile no.
   String Mobile=jTextField1.getText();
   if (Mobile.isEmpty())
   //Execute this part if text field is blank
```

```
{
   jTextField2.setText("");
   jTextField3.setText("");
   JOptionPane.showMessageDialog
                              (this,"Enter the Mobile No");
}
// This part is executed if a Mobile No is
// input in the text field
else
{
   try
   {
       Class.forName("java.sql.DriverManager");
       Connection con = (Connection)
       DriverManager.getConnection
       ("jdbc:mysql://localhost:3306/cbse",
       "root", "abcd1234");
       Statement stmt = (Statement) con.createStatement();
       String query="SELECT NAME,EMAIL FROM CONTACT
                     WHERE MOBILE='"+Mobile+"';";
       ResultSet rs=stmt.executeQuery(query);
       if (rs.next())
       {
          String Name  = rs.getString("Name");
       //Retrieve the name
```

```
                String Email = rs.getString("Email");
                //Retrieve the email
                jTextField2.setText(Name);
                jTextField3.setText(Email);
            }
            // This part is executed if no matching record is found
            else
            JOptionPane.showMessageDialog
                                (this,"Sorry!No such Mobile No");
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }
}
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt)
{
  System.exit(0);
}
```

Figure 6.15 Code for the Search From Contacts Application

Let us now understand the code in detail line by line:

```
String Mobile=jTextField1.getText();
```

- Declare a variable called mobile and initialize it with the value input by the user in the first text field which has been retrieved using the getText() method.

```
if (Mobile.isEmpty())
{
    jTextField2.setText("");
    jTextField3.setText("");
    JOptionPane.showMessageDialog(this,"Enter the Mobile No");
}
```

- Check if the mobile variable is empty i.e. in case the user has not input any mobile number then reinitialize both the text fields with blank spaces using the setText() method and show an error message using the showMessageDialog() method.

```
else
{
    try
    {
        Class.forName("java.sql.DriverManager");
        Connection con = (Connection)
        DriverManager.getConnection
        ("jdbc:mysql://localhost:3306/cbse",root",
        "abcd1234");
        Statement stmt = (Statement) con.createStatement();
        String query="SELECT NAME,EMAIL FROM CONTACT
                WHERE MOBILE='"+Mobile+"';";
```

- If the user has entered a valid mobile number then perform all the steps for setting up a connection and initializing a query as learnt in the first database application.

```
ResultSet rs=stmt.executeQuery(query);
```

- Instantiate an object named rs of the ResultSet class and initialize it with the records returned by executing the SQL statement stored in the query variable. The executeQuery() method is used when we simply want to retrieve data from a table without modifying the contents of the table.

```
if (rs.next())
```

- Check if there is a matching record in the result set using the if construct. The next() method of the result set is used to move to the next record of the database.

```
String Name  = rs.getString("Name");
String Email = rs.getString("Email");
jTextField2.setText(Name);
jTextField3.setText(Email);
```

- Create two variables named Name and Email and initialize them with the values retrieved from the table using the getString() method. Next display these values retrieved from the table in the relevant text fields using the setText() method.

```
else
{ JOptionPane.showMessageDialog
    (this,"Sorry!No such Mobile No"); }
```

- If there is no matching record found in the table then display an error message using the showMessageDialog() method.

```
catch (Exception e)
{
    JOptionPane.showMessageDialog(this, e.getMessage());
}
```

- In case of an exception, retrieve the error message string using the getMessage() method and display it in a dialog box using the showMessageDialog() method.

A sample run of the validation check applied in the above application is shown in Figure 6.16

Figure 6.16 Validation Message in Search From Contacts Application

Just like we need to add the DriverManager, Connection and Statement class libraries, similarly we need to import the ResultSet class library when we are retrieving data from a database. In case the above mentioned library is not imported then the error window is displayed as shown in Figure 6.17.

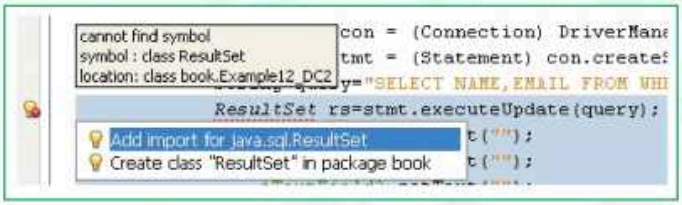


Figure 6.17 Error Message Indicating Missing ResultSet Class Library

Data Connectivity Application 3: To Display the Details Stored in the Back-end Based on the Name Input by the User in a Form.

Observe the sample run of our next application as shown in Figure 6.18 carefully. Can you observe the difference in this application as compared to our previous one?

In the previous application, we are searching on mobile number which will be unique i.e. we will always have a single matching record. What will happen if we perform a search on Name? In this case there is a possibility that we may have more than one matching record. So the major difference between the two applications is that in the new application we will aim at performing a search in which multiple records may be returned.

Figure 6.18 Sample Run of the Search For Contacts (For Given Name)

The form design and most of the coding is similar with minor differences. Observe the code given in Figure 6.19 carefully and try to point out the differences.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
   String name=jTextField1.getText();
   if (name.isEmpty())
      JOptionPane.showMessageDialog(this,"Name not Entered");
   else
   {
```

```
try
{
    Class.forName("java.sql.DriverManager");
    Connection con = (Connection)
    DriverManager.getConnection
    ("dbc:mysql://localhost:3306/cbse", "root",
    "abcd1234");
    Statement stmt = (Statement) con.createStatement();
    String query="SELECT Mobile,Email FROM Contact
                  WHERE Name='"+name+"';";
    ResultSet rs=stmt.executeQuery(query);
    int Found=0;
    while(rs.next())
    // Till there are records in the result set
    {
        String mobile = rs.getString("Mobile");
        String email  = rs.getString("Email");
        jTextField2.setText(mobile);
        jTextField3.setText(email);
        JOptionPane.showMessageDialog
                  (null,"Click OK to continue!!!");
        Found++;
        //Increment the variable to indicate a matching
        //record has been found
    }
```

```
            if (Found==0)
                JOptionPane.showMessageDialog
                (this,"Sorry! No such Name in Contact List");
    }
      catch (Exception e)
      {
         JOptionPane.showMessageDialog(this, e.getMessage());
      }
      }
}
```

Figure 6.19 Code for the Search for Contacts (For Given Name) Application

Let us now discuss the changes (additional commands) made to the previous code one by one:

i) The first change is declaration of a new integer variable named Found. This variable is initialized to 0 and is used to keep a track of how many matching records have been found.

ii) The second change is that we have used while loop to traverse through all the records in the table to find all possible matching records.

iii) The third change is the statement used to increment the variable found each time a matching record is encountered.

Data Connectivity Application 4: To Display Records Retrieved From the Back-end in the Table Component

In all the previous applications we have aimed at displaying only one record at a time in a form. What if we want to see all the records in one go? The next application is aimed at solving this problem. The aim of this application is to display all the records in a tabular form.

First, create a new table named Official with the same structure as that of table Contact. Add few records in this table.

Now let us design the form. First add a new JFrame form and set its title property to "Contact List". Now, add the following components on the form:

- One table to display the records retrieved from the table.
- Two radio buttons to select whether the records of the Contact table are displayed or the records of the Official table are displayed.
- Two appropriate labels - one against each of the radio button to direct the user.
- Three buttons - one to refresh data displayed in the table, one to reset the table and one to exit from the application.

Change the properties of the radio buttons, labels and button components as learnt earlier so that the form looks exactly like the one displayed in Figure 6.20. Before associating the code with the buttons, we need to customize the table component.

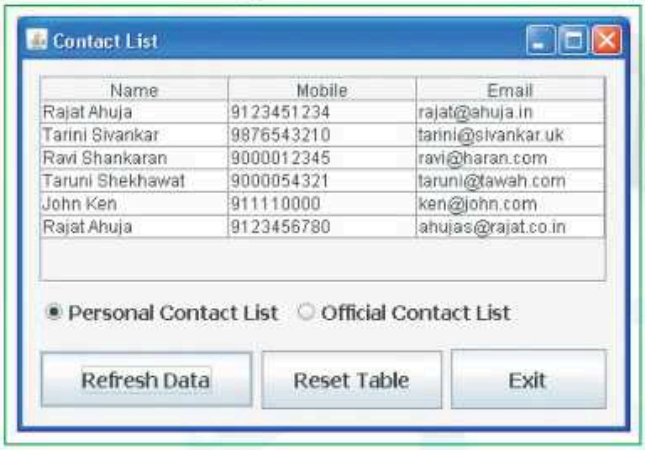


Figure 6.20 Form Design of the Modified Contact List Application

Right click on the table component and select Table Contents option to open the Customizer Dialog window. In this window choose the Columns tab and customize the column names by changing the Title property of each column as shown in Figure 6.21.

Next choose the Rows tab and delete all the four default rows. These rows have to be deleted otherwise they appear as blank rows in the final output and the retrieved data is displayed from fifth row onwards.

Once the form has been totally customized, the next step is to associate code with all the three buttons. Double click on the buttons one by one in the design window to reach the point in the source window where the code needs to be written. Add the code for each of the buttons as given in Figure 6.22.

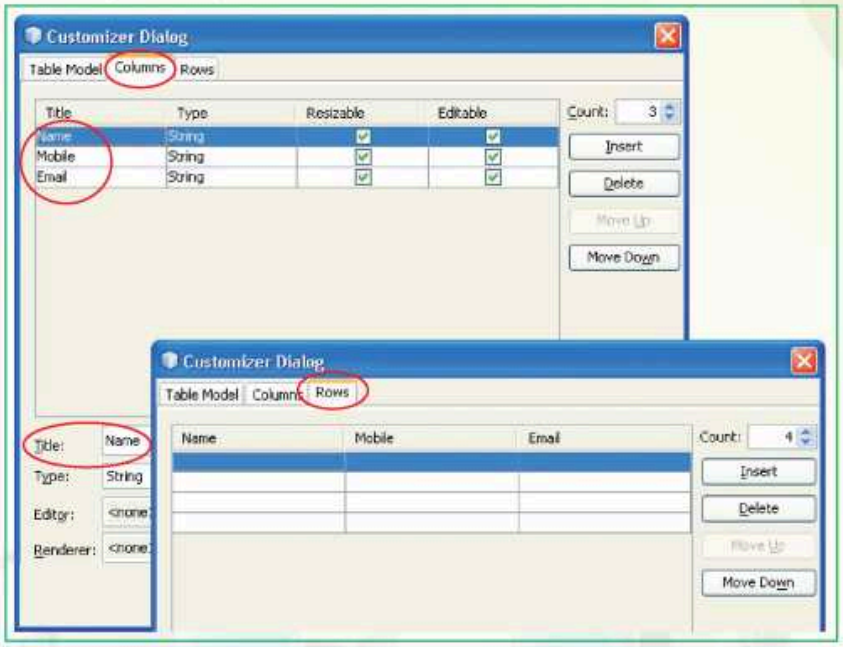


Figure 6.21 Customizing Columns and Rows of the Table Component of the Form

Before proceeding to writing the code, let us understand the purpose of the two buttons - Refresh Data and Reset Table. The purpose of the Refresh Data button is to display the data from the selected table without clearing the previous data. This means that the new data will be displayed in continuation of the previous displayed data. Whereas, if we press the Reset Table button, then it deletes all previous rows of the table component of the form.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
   DefaultTableModel model = (DefaultTableModel)
   jTable1.getModel();
   try
   {
     Class.forName("java.sql.DriverManager");
     Connection con = (Connection)
         DriverManager.getConnection
         ("jdbc:mysql://localhost:3306/cbse",
                       "root", "abcd1234");
     Statement stmt = (Statement) con.createStatement();
     String Tname;
     if (jRadioButton1.isSelected())
         Tname="Contact";
     else
         Tname="Official";
     /* The query is executed on different tables depending
     upon the Purpose */
     String query="SELECT * FROM "+Tname+";";
     ResultSet rs = stmt.executeQuery(query);
     while(rs.next())
     {
         String Name   = rs.getString("Name");
         String Mobile = rs.getString("Mobile");
```

```
            String Email  = rs.getString("Email");
            model.addRow (new Object[] {Name, Mobile,Email});
        }
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog (this, e.getMessage());
    }
}
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    DefaultTableModel model = (DefaultTableModel)
    jTable1.getModel(); int rows=model.getRowCount();
    if (rows>0)
   {
       for (int i=0; i<rows; i++)
            model.removeRow(0);  // To remove all rows from
            current model
    }
}
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
    System.exit(0);
}
```

Figure 6.22 Code for the Modified Contact List Application

Let us now understand the code in detail line by line:

```
DefaultTableModel model = (DefaultTableModel)
jTable1.getModel();
```

- To retrieve the Model of the newly created table and type cast it to the Default Model.

```
Class.forName("java.sql.DriverManager");
Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cbse",
                            "root", "abcd1234");
Statement stmt = (Statement) con.createStatement();
```

- Instantiate a Connection and a Statement object as before

```
String Tname;
```

- Declare a variable named Tname of type String. This variable will be sued to store the name of the table from which the data has to be retrieved.

```
if (jRadioButton1.isSelected())
    Tname="Contact";
else
    Tname="Official";
```

- Check which radio button has been selected and accordingly initialize the variable Tname.

```
String query="SELECT * FROM "+Tname+";";
```

- Create a variable called query of type String and initialize it with the SQL statement to retrieve all records from the specified table. The value of the specified table is stored in the variable Tname and so this variable is concatenated with the normal SQL statement.

```
ResultSet rs = stmt.executeQuery(query);
```

- Execute the SQL query using the executeQuery() method. Instantiate a variable named rs of class ResultSet to store the records returned by the SQL query.

```
while(rs.next())
{
    String Name   = rs.getString("Name");
    String Mobile = rs.getString("Mobile");
    String Email  = rs.getString("Email");
    model.addRow (new Object[] {Name, Mobile,Email});
}
```

- Till there are records in the database, keep retrieving the values of Name, Mobile and Email from the table one by one using the getString() method and store them in the three String variables using the assignment operator.
- After storing the values in the variables, display these values in the table component on the form using the addRow() method.

```
int rows=model.getRowCount();
```

- Declare a variable named rows of type integer and initialize it with the total number of rows in the table component of the form using the getRowCount() method.

```
if (rows>0)
{
    for (int i=0; i<rows; i++)
        model.removeRow(0);
}
```

- If the number of rows is greater than zero, then use a for loop to reset the table component i.e. delete all the rows using the removeRow() method. Note that the parameter passed to the removeRow() method is always zero as we are always deleting the topmost row and this process is repeated rows number of times to ensure that all the rows are deleted.

Data Connectivity Application 5: To Add Records in a Table Accepted Through a Form with Hindi Interface

Now that we are clear about data connectivity, let us learn a new feature - to add multilingual support facility to our Netbeans form and database and then display records in our native language as shown in Figure 6.23.

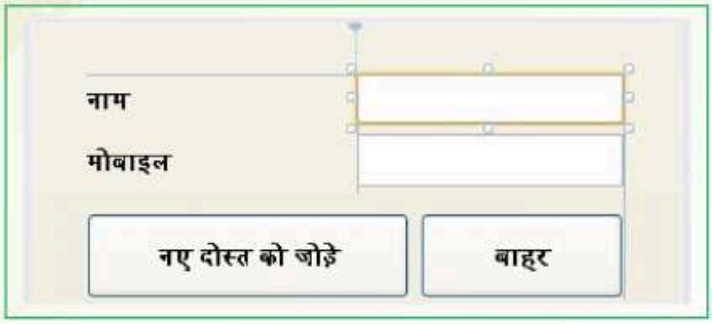


Figure 6.23 Form Design of the Contact List Application with a Hindi Interface

First create a database named *CBSEHINDI* using the following command:

```
CREATE DATABASE CBSEHINDI DEFAULT CHARACTER SET UTF8;
```

Note that the DEFAULT CHARACTER SET UTF8 has been added to the above command to have Unicode support for entering and processing data in HINDI language. Adding this command while creating the database ensures that all tables created within this database support Hindi language processing.

Next create a table named Contact in the CBSEHINDI database using the following command:

```
CREATE TABLE Contact (Name VARCHAR(20),Mobile VARCHAR(12),
                      Email VARCHAR(25));
```

Now while creating the form as shown above for the Netbeans application, remember to use unicode supported fonts for jTextFields, jLabels and jButtons (such as Arial Unicode MS) as shown in Figure 6.24.

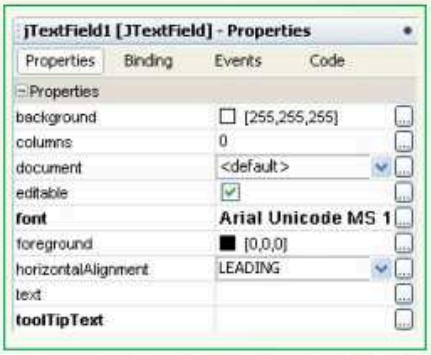


Figure 6.24 Setting the Font Properties for Multilingual Support

The code will be similar as the earlier example shown in Data Connectivity Application 1 (Only change the name of the database to cbsehindi as shown in the following code) and

```
Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cbsehindi","root",
"abcd1234");
```

A sample run of the Hindi interface application is shown in Figure 6.25.

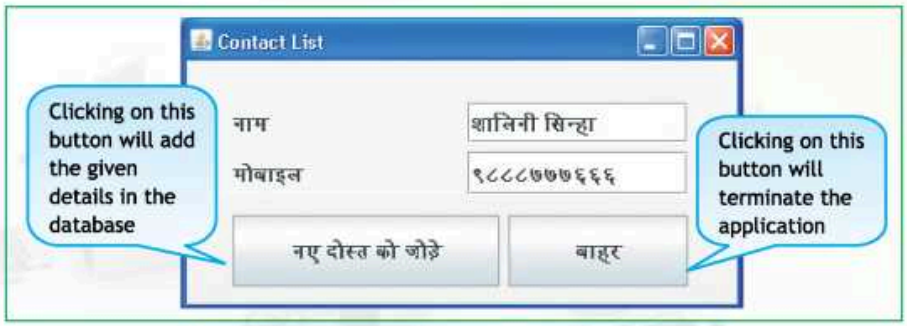


Figure 6.25 Sample Run of the Contact List Application with Hindi Interface

All the records in the cbsehindi database are stored in Hindi. Therefore, if we execute an SQL command to display all the records of the database then the records will be displayed in Hindi as shown in Figure 6.26.

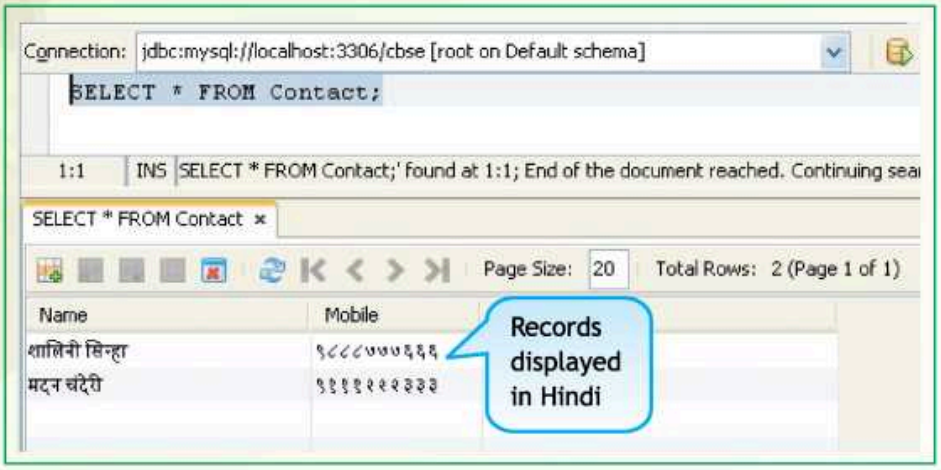


Figure 6.26 Running Simple SQL Statement in Netbeans for Data Display in Hindi

Future Trends

Cloud Computing

Cloud computing is a technology used to access services offered on the Internet cloud. Everything an Informatics System has to offer is provided as a service, so users can access these services available on the "Internet Cloud" without having any previous know-how. The cloud is just one example of a virtualized computing platform, and the next generation of developer tools must enable developers to build software that deploys and performs well in cloud and other virtual environments.

Summary

- A GUI (The Front-End) helps the user to design forms to accept data and provide instructions for retrieving information from the backend.
- A database (The Back-End) helps in storing data in organized manner for future reference and use

- A link needs to be established between the front end form and the back end table using jdbc driver (or an equivalent) to facilitate communication between the two.
- A complete application requires a combination of all three concepts i.e. it requires a user friendly interface, an efficient connection link and a database
- The JDBC DriverManager class defines objects which can connect Java applications to a JDBC driver.
- The JDBC Connection interface defines methods for interacting with the database via the established connection.
- To execute SQL statements, you need to instantiate (create) a Statement object using the connection object.
- The getConnection() method of the DriverManager class is used to establish a connection to the database.
- The getMessage() method is used to retrieve the error message string.
- The executeUpdate() method of the Statement class is used to update the database with the values given as an argument.
- The executeQuery() method is used when we simply want to retrieve data from a table without modifying the contents of the table.
- The next() method of the result set is used to move to the next record of the database.
- The getModel() method is used to retrieve the model of a table.
- The getRowCount() method is used to retrieve the total number of rows in a table.
- The removeRow() method is used to delete a row from a table.
- It is possible to add Multilingual support in MySQL database by setting the appropriate character set while creating the database.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. We may use _____ to develop the Front-End of an application.

   A. GUI B. Database

   C. Table D. None of the above

2. The _____ method is used when we simply want to retrieve data from a table without modifying the contents of the table

   A. execute() B. queryexecute()

   C. query() D. executeQuery()

3. Which of the following class libraries are essentially required for setting up a connection with the database and retrieve data?

   A. DriverManager B. Connection

   C. Statement D. All of the above

4. The _____ is required to establish connectivity between the Java code and the MySQL database.

   E. JDBC API F. JDBC Driver

   G. Both A and B

5. To execute SQL statements, you need to instantiate a _____ object using the _____ object.

   A. Connection , Statement B. Statement, Connection

   C. JDBC Driver, Statement D. JDBC DriverManager, Statement

6. Which of the following statements are false?

   A. The getConnection() method is used to establish a connection.

   B. An application can have more than one connection with a database.

   C. An application cannot have more than one connection with a database.

   D. An application can have many connections with different databases.

7. The _____ method is used to instantiate a statement object using the connection object

   A. getConnection() B. getStatement()

   C. createConnection() D. createStatement()

8. The _____ method is used to move to the next row of the result set.

   A. next() B. findNext()

   C. last() D. forward()

ANSWER THE FOLLOWING QUESTIONS

1. What is a GUI?
2. Explain the terms ResultSet with respect to a database.
3. Explain the usage of the following methods:

   a) next()

   b) removeRow()

4. Give two appropriate reasons of connecting GUI form with a Database.
5. What is JDBC? What is the importance of JDBC in establishing database connectivity.
6. Name the 3 essential class libraries that we need to import for setting up the connection with the database and retrieve data from the database.
7. Explain the importance of:

   a) JDBC b) forName() method

   c) getModel() method

8. Consider the following code and answer the questions that follow:

```
String query="SELECT NAME,EMAIL FROM CONTACT"
ResultSet rs=stmt.executeQuery(query); //Statement 1
if (rs.next())                          //Statement 2
{
```

```
        String Name  = rs.getString("Name");
        String Email = rs.getString("Email");
        jTextField2.setText(Name);
        jTextField3.setText(Email);
    }
```

a) Name the table from where the data is being retrieved

b) How many objects of the native String class have been declared in the code? Name the objects.

c) Identify the object name, class name and method name used in the statement labeled as Statement 1.

d) Explain the function of the statement labeled as Statement 2.

LAB EXERCISES†

1. Modify the SMS application developed in Chapter 5 to store the messages in a table along with the number of characters in a message.
2. Modify the password application developed in Chapter 5 to store the name and password of users in a table. The application should also update the password whenever the user modifies his password and should keep a count on how many times the password is being updated.
3. Design a GUI Interface for executing SQL queries for the above developed Password Application to accomplishing the following tasks:
   A. Retrieve a list of all the names and display it in a table form.
   B. Retrieve a list of names where name begins with a specific character input by the user and display it in a table form.
   C. Display all records sorted on the basis of the names.
4. Modify the password application developed above to handle processing in Hindi language.

TEAM BASED TIME BOUND EXERCISES

(Team size recommended: 3 students each team)

1. Divide the class in groups of four and tell each group to develop a phone book application to store and retrieve information about their friends on the basis of name, phone number or birthday. The information stored may include name, phone number, address, birthday, category etc.

2. Each group should next write SQL queries (to be executed on the above created table) for the following(and design appropriate interface element):

   A. Retrieve a list of all the users and display it in a table.

   B. Retrieve a list of users whose name begins with a specific character input by the user and display it one at a time in a form.

   C. Search and display records of the users whose birthday is in a particular month where month is input by the user.

WEB APPLICATIONS

Learning Objectives

After studying this lesson students will be able to:

- Define the term hypertext and state the purpose of HTML.
- Identify the main parts of an HTML document.
- Identify and state the purpose of different HTML elements.
- Differentiate between HTML elements, tags, and attributes.
- Create web pages using different basic and formatting tags.
- Differentiate between HTML and XML.

The World Wide Web (or simply the Web or WWW) is a system of sharing interlinked hypertext documents over the internet. These documents are stored on web-servers on the internet and contain text, images, videos and other multimedia. These documents also contain hyperlinks to navigate among them. HTML (Hyper Text Markup Language) is the basic language which is used to create Hypertext documents. In this lesson we are going to learn how to create hyper text documents using HTML.

Puzzle [7]

You have 5 jars of pills. Each pill weighs 10 gram, except for contaminated pills contained in one jar, where each pill weighs 9 gm. Given a scale, how could you tell which jar has the contaminated pills in just one measurement?

Well as we can infer from the puzzle given above that assimilation of information and then using the information to solve a problem are of utmost importance. Similarly, information retrieval from the World Wide Web is also of prime importance and this information retrieval is based on protocols.

Introduction

Internet is such a wonder box that whenever we need some information on any topic, we refer to internet. Information is available on almost all the topics - from Ayurveda to Advaced Medical Surgeries, from basic colors to advanced designs, from fundamentals of computers to latest developments in the field of supercomputers - you name it and internet has it. This information is actually stored on some computers on the net. These computers are called the servers. The information is stored in the form of some documents called hyper text documents. All the related documents on a server are linked together using hyperlinks. Therefore using hyperlinks we can move from one document to another. This is formally called navigation. There are a number of ways to create hyper text documents. There are many specialized software packages like Dreamweaver, CoffeCup etc. which are used to create web documents. The simplest way to create a web document is to use a text editor like notepad, notepad2 etc. and write code in HTML. A hyper text document on the web is also called a web page.

The information over the web is shared using a protocol called Hyper Text Transfer Protocol (HTTP)

History of World Wide Web

The concept of WWW was designed in 1989 by Tim Berners-Lee and scientists at CERN (Geneva), the European centre for High Energy Physics. Their purpose was to make sharing and retrieval of research material simpler. A year later they had developed a 'browser/editor' program and had named the program World Wide Web. The World Wide Web grew rapidly and attained its present form. Its further development is guided by the WWW Consortium (W3C) based at the Massachusetts Institute of Technology in Cambridge, Massachusetts

Fig.7.1 Tim Berners-Lee

Uniform Resource Locator

The uniform resource locator (URL) is the unique identifier of a web page. The address or URL of the current page you are on appears in the "Address Bar" of the web browser. You can go directly to a web page if you know its URL by simply typing the URL in the address bar. You can click in the address bar at any time and overwrite the current address with another URL to jump to a different web page.

The most general form of a URL syntax is as follows:

Protocol://domain name/<directory path>/<object name>

For example:

`http://www.openoffice.org/dev_docs/features/3.2/rc2.html`

The elements of this syntax are as follows:

Part	Description	Example
Protocol	represents name of the protocol which is used to transfer the data/web page	http, ftp
Domain Name	represents name of the web server where the web page resides	www.openoffice.org
Directory Path	Represents location of the web page on the web server	dev docs/features/3.2
Object Name	name of file	rc2.html

When the URL of a web page is given to the web browser, the browser sends a request for this page to the relevant web server. The web server, upon getting this request, sends the requested web-page to the browser and then the browser displays this page. A web browser knows how to display web pages.

Know More!

Semantic Web

The concept of Semantic Web occurred to Tim Berners-Lee, inventor of the WWW. The Semantic Web provides a common format that allows data to be shared and reused from diverse sources. Semantic Web is an effort to make reuse and republishing of data easier. Semantic web is based on concept based navigation of the web as opposed to fixed term navigation undertaken earlier.

Semantic Web provides technologies that can merge information from two sources. The component that helps in this is the Resource Description Framework (RDF). When information from two sources in RDF needs to be merged, we can concatenate the files into one big file - joining on those terms which are defined to correspond to the indicators in both the files.

More information on this is available on http://www.w3.org/RDF

Let us now start on our web page designing journey. We will use the tools that are already available on our system.

Following will be our first web page:

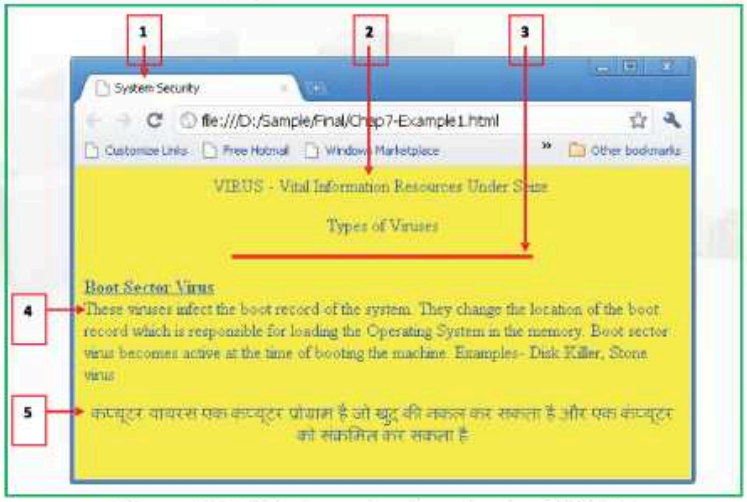


Figure 7.2: Web Page developed using HTML Tags

The highlighted features of this web page are:

1. Title - 'System Security'
2. Center Aligned Paragraph
3. Horizontal Rule
4. Left Aligned Paragraph
5. Contents in Hindi

Now let us look at the source code of the web page

```
<HTML>
   <HEAD>
     <TITLE> System Security</TITLE>
   </HEAD>
 <!--Beginning of Body tag-->
   <BODY bgcolor = "yellow" TEXT="blue"
       LINK="CYAN" ALINK="BLACK" VLINK="TEAL">
     <CENTER> VIRUS - Vital Information Resources Under Seize <BR>
     <P>Types of Viruses</P>
     <HR SIZE= 4 WIDTH= 50% COLOR="RED"> <CENTER>
     <P ALIGN="LEFT"><u><b>Boot Sector Virus</b></u> <BR>
     These viruses infect the boot  record of the system. They
change the location of the boot record which is responsible
     for loading the Operating  System in the memory. Boot sector
virus becomes active at the time of booting the  machine.
Examples- Disk Killer, Stone virus  </P>
     <p lang="hi"> कंप्यूटर वायरस एक कंप्यूटर प्रोग्राम है जो खुद की नकल कर सकता है
और एक कंप्यूटर को संक्रमित कर सकता है </P>
   </BODY>
</HTML>
```

Figure 7.3: Code for Figure 7.2

The code is written using HTML.

HTML

HTML stands for Hyper Text Markup Language. It is a markup language used to create HTML documents. An HTML document defines a web page. For example, consider Figures 7.3 and 7.2 shown above. Figure 7.3 shows a text file which is actually an HTML document. Figure 7.2 shows the corresponding web page created by the browser (Google

Chrome in this example) using the HTML document given in Figure 7.3. Simply put, Figure 7.2 shows a web page defined by the HTML document shown in Figure 7.3. To define web pages HTML provides various markup elements. Using these elements we can specify various parts of a web page and also formatting of the contents of the web page. For example, some of the HTML elements used in Figure 7.3 are <HTML> . . . </HTML>, <P> . . . </P>, <BR> etc. HTML elements are used to specify headings, paragraphs, lists, tables, images and much more. When an HTML document is opened in a web browser, the browser interprets these elements and displays the HTML document as a web-page. Here we should understand the difference and relationship between an HTML document and a web page. An HTML document is a text file that contains HTML elements. An HTML document shown by a web browser is called a web page.

HTML is a subset of Standard Generalized Markup Language (SGML) and is specified by the World Wide Web Consortium (W3C).

Parts of an HTML document :

Any HTML document, in general, contains at least three elements - HTML, HEAD, and BODY. These elements are specified by the following respective tags:

1. <HTML> . . .</HTML>
2. <HEAD> . . .</HEAD>
3. <BODY> . . .</BODY>

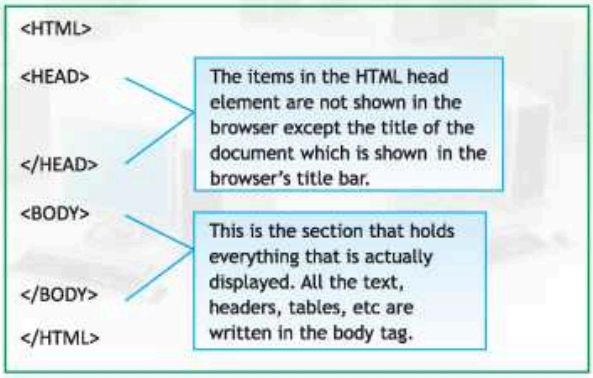


Figure 7.4: Parts of an HTML document

Each document consists of head and body text. The head contains the title and the body contains the actual text that is made up of paragraphs, lists, and other elements. Let us first learn to implement these examples on the system. We can do this easily by following the steps given below:

- Type the code in any text editor, e.g. gedit, Notepad2, Notepad etc.
- Save the file in a specified directory with an extension as .htm or .html
- Open the file in a web browser.

The corresponding web page is displayed in the browser window.

HTML: Elements, tags, and attributes

Any HTML document contains:

(a) information that is to be displayed by the web-browser.

(b) formatting information that tells the web-browser about lay-out of this information when it is displyed.

In an HTML document, the formatting information is given in the form of HTML elements. An HTML element is specified by the corresponding tags. For example BODY element is specified by the pair of tags <BODY> and </BODY>, B (Bold) element is specified by the pair of tags <B> and </B>. A tag may also contain attribute(s). An attribute defines a property of an element and is specified in the format **attribute = "value"** in the start tag of the element.

An example: BODY element is specified by the pair of tags <BODY> and </BODY>. Here <BODY> is the start tag and </BODY> is the end tag. By default, the background colour of a web page is white, but we can change it to yellow (or to any other colour) by specifying the BGCOLOR attribute in the <BODY> tag as <BODY BGCOLOR = "YELLOW">. The complete thing starting from the <BODY> (with or without attributes) tag till the </BODY> tag including the contents between these two tags is the BODY element.

HTML elements are of two types- Container elements and Empty elements. These are discussed below.

Container Elements :

A container element is specified by a pair of tags - Start tag and End tag. These tags also called ON tags and OFF tags. Start tag consists of the tag name enclosed in left and right angular brackets. The end tag is identical to the start tag, except for a slash (/) that precedes the text within angular brackets of the end tag. e.g.,

```
<BODY> . . . </Body>
```

Container elements contain parameters and the parameters of an element are given between the start tag and end tags.

```
<BODY> . . . </BODY>
```

↓

Parameters

Elements in HTML may also contain attributes that can be given along with the tag name in the angular brackets of the start tag. E.g.,

```
<BODY BGCOLOR="BLUE" TEXT="RED">
```

Attributes

Empty Elements

Empty elements have only a start tag and no end tag. Hence, an empty element has no parameters, but can take attributes, which are given within angular brackets of the start tag.

Example <BR> tag

> HTML is not case sensitive, i.e., tags and attributes can be written in small or capital alphabets. They can also be written using a mix of capital and small alphabets.

Let us now discuss HTML elements in detail:

HTML element

It is a container element started by <HTML> tag and ended by </HTML> tag. It identifies the document as an HTML document. It does not have any effect on the appearance of the document, but tells the browser that the current document is an HTML document.

Syntax: **<HTML> . . .</HTML>**

HTML element further contains HEAD and the BODY elements, which can further contain a number of other elements.

HEAD element - It is a container element started by <HEAD> tag and ended by </HEAD> tag. It defines the HTML document header and does not affect the appearance of the document in the browser window. The header contains information about the document.

Syntax: **<HEAD> . . .</HEAD>**

TITLE element - It is a container element started by <TITLE> tag and ended by </TITLE> tag. Every HTML document should contain the title to be displayed in the title bar of the browser window. If an HTML document does not contain a title, then the file name of the HTML document is displayed in the title bar. We can see in Figure 7.3 System Security is written in the TITLE element and thus it is displayed in the browser's title window as observed in figure 7.2. The TITLE element is placed in HEAD element.

Syntax: **<TITLE>. . .</TITLE>**

BODY element - It is a container element started by <BODY> tag and ended by </BODY> tag. It contains the main contents of the document as parameter.

Syntax: **<BODY> . . .</BODY>**

BODY tag contains many attributes. Some of the important attributes are discussed below.

1.	**Attribute:**	Background
	Use:	It is used to specify the path and filename of an image that has to be used as the background of the document. If the referenced image is smaller than the browser window, it will be tiled to fit and will scroll with the text on the page. The filename extension has to be specified along with the filename.
	Syntax:	`BACKGROUND= "path/filename.jpg"`
2.	**Attribute:**	BGCOLOR
	Use:	It sets the background colour of the web page. Most of the browsers recognize the popular colour names like RED, GREEN, YELLOW, GREY, AQUA ETC. If we want to specify a colour which does not have a specific name but we know its RRGGBB composition, then we can specify this RRGGBB composition in the BGCOLOR attribute to get the background color. E.g., in Figure 7.3, instead of writing BGCOLOR = "GREY", we could have written BGCOLOR = "#999999" and would have got the same effect. If a background image is also present, the BGCOLOR specified shines through regions where the background image is transparent.
	Syntax:	`BGCOLOR= "colorname"  OR BGCOLOR= "#rrggbb"`
3.	**Attribute:**	Text
	Use:	It sets the colour of the normal text in the document. Colour values can be given in the same way as that of the BGCOLOR attribute. The default is black (hexadecimal code #000000).
	Syntax:	`TEXT = "colorname"  OR TEXT = "#rrggbb"`

4.	**Attribute:**	Link
	Use:	This sets the colour of all of the non-visited links on the page. We can set this to any color of our choice. The default setting for a non-visited link is usually blue.
	Syntax:	`Link="colorname" OR LINK = "#rrggbb"`
5.	**Attribute:**	Alink
	Use:	This sets the colour of active links on the page. An active link is a link that has just been keypressed by the user. Here 'keypressed' does not mean 'clicked'. 'Keypressed' means that the mouse has been taken over the link and the left mouse button is pressed but not released. 'Clicked' means the button has been pressed and then released so that the browser opens the corresponding link. We can set active link colour to any colour. The default Alink colour is red.
	Syntax:	`Alink="colorname" OR ALINK = "#rrggbb"`
6.	**Attribute:**	Vlink
	Use:	This changes the colour of a visited link on the page. We can set this to any color. The default setting for a visited link is usually violet.
	Syntax:	`Vlink="colorname" OR VLINK = "#rrggbb"`

NOTE:

1. The BACKGROUND attribute when used, overrides the BGCOLOR attribute. Thus if an HTML document contains both of these tags, then the specified image will be displayed in the background of the page and the background color will be ignored. If however, the image is not found, the specified background colour will be used as background.

2. The World Wide Web Consortium (W3C) has listed 16 valid color names for HTML: **aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.** Majority of browsers understand the names of many other colors also. Any color is composed of 3 basic colors: RED, GREEN, BLUE. If a color does not have a specific name (or if we don't know its name), its RGB (Red, Green, Blue) composition may be specified. This composition is specified in hexadecimal notation in the format "#rrggbb". The value of each of the composition colors is specified as a hexadecimal number and may vary from 00 to FF. For example, the color code "#00FF00" specifies pure green color and "#969696" specifies a color which contains a mix equal parts of Red, Green, and Blue. Just try and find out which color it displays.

3. Following this, whenever we say "colorname" in this chapter, it shall mean color name as well as "rrggbb" composition of the color.

To Write in Hindi (and other regional languages) on the Web Page

If lang attribute with a suitable value is added in the <HTML> tag, the whole web page is displayed in the specified language. E.g.

```
<html lang="hi">
```

Tells the web browser to display the web page in Hindi (Devnagari script).

Instead of **"hi"** we can use **"ks", "kok", "ne", "mr", "sa"** for Kashmiri, Konkani, Nepali, Marathi and Sanskrit respectively. If we want to use multiple languages on the same page, we can write this attribute with the <p> tag also.

```
<p lang="hi">
```

हम हिंदी लिख रहे हैं

We can download any of the devnagari script fonts like Susha, Mangal etc. and use them in the HTML document. These fonts use the unicode character encoding. Unicode makes the web pages automatically searchable. In case we use Unicode fonts in our document we have to select Encoding as Unicode while saving the file.

The steps to install and use Hindi font are given in the appendix at the end of this book.

Know More!

You can get more information about UTF-8 and Unicode from the net. There you can also find the information about how to use regional languages in the HTML documents.

P (Paragraph) element - It is a container element started by <P> tag and ended by </P> tag. This element is used to start a new paragraph. HTML does not recognize the return/enter key we enter in text editor; therefore we use <P> tag to start a new paragraph. <P> starts a new paragraph with extra space before the first line.

Syntax: `<P ALIGN="alignment" lang = "language"> . . .</P>`

The end tag </P> of Paragraph element is optional and may be skipped. But it is highly recommended to put this end tag. Skipping it may produce unexpected results or errors in some browsers. Future versions of HTML will not allow you to skip end tags.

ALIGN attribute specifies the horizontal alignment of paragraph. Any of the alignments - Center, Left, Right - can be specified.

LANG attribute specifies the language in which the paragraph is to be displayed as discussed above.

Horizontal Rule (HR) element - This is an empty element specified by <HR> tag. The <HR> tag draws a horizontal line across the document frame or window. We can use a horizontal line to visually divide the information into sections.

Syntax:

```
<HR ALIGN = "alignment" NOSHADE SIZE = "thickness"
                    WIDTH = "width" COLOR="colorname" >
```

The attributes used with <HR> are:

1.	**Attribute:**	ALIGN
	Use:	Specifies how the horizontal rule should be aligned. The alignment can be left, right, or center. Default is center alignment.
	Syntax:	`ALIGN = "Alignment"`
2.	**Attribute:**	NOSHADE
	Use:	produces a solid horizontal rule that has no shading.
	Syntax:	`NOSHADE`
3.	**Attribute:**	Size
	Use:	Size defines the thickness of the horizontal rule. This thickness is specified in the number of pixels. Default is 2 pixels.
	Syntax:	`SIZE = "n"  - where n is a natural number`
4.	**Attribute:**	Width
	Use:	"Width" defines the horizontal width of the line. The default is the width of the page. The measurement value can be the number of pixels, e.g., "50" , or a percentage of the page width, e.g., "75%". Here 75% means that the horizontal rule will cover 75% of the page width.
	Syntax:	`WIDTH = "n/n%"  - where n is a natural number`
5.	**Attribute:**	Color
	Use:	Sets the colour of the horizontal line.
	Syntax:	`COLOR= "colorname"`

Line Break (BR) Element - This is an empty element specified by <BR> tag.
 tag forces a line break which implies that the text/image following the tag will be moved to the next line when displayed in the browser.

Syntax: **
**

Well, now that we have designed our first html page we are familiar with the basic tags. The tags learnt so far will help us in designing very basic web pages, but if we want our web page to have different fonts, font sizes and headings we will have to delve further and learn the usage of formatting elements. Look at the web page shown below in which we have used formatting elements.

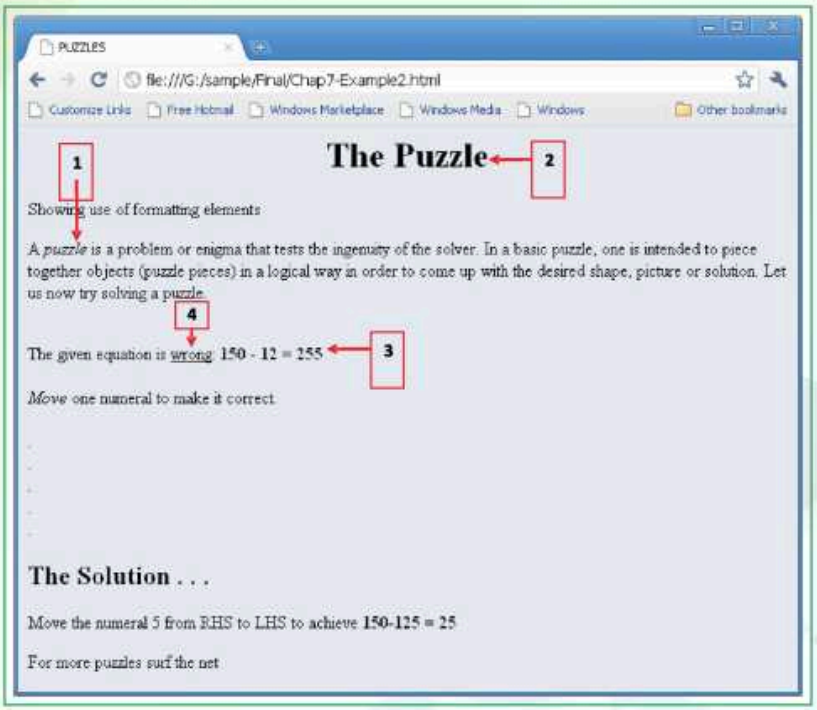


Figure 7.5 Web page created using formatting elements

The HTML code to generate this web page is given below:

```
<HTML>
   <HEAD>
     <TITLE>PUZZLES</TITLE>
   </HEAD>
   <BODY BGCOLOR="#E3E4FA">
     <H1 ALIGN="CENTER">The Puzzle</H1>
     <COMMENT>Showing use of formatting elements</COMMENT>
     <P>A <I>puzzle</I> is a problem or enigma that tests the
  ingenuity of the solver. In a basic puzzle, one is intended to
  piece together objects (puzzle pieces) in a logical way in order
  to come up with the desired shape, picture or solution. Let us
  now try solving a puzzle.<BR>
     <P>The given equation is <U>wrong</U>: <B>150 - 12 =
  255</B><BR><BR/>
     <I>Move</I> one numeral to make it correct.<BR >
     <BR>
      .
     <BR>
      .
     <BR>
      .
     <BR>
      .
     <BR>
      .
     <BR>

     <H2 >The Solution . . .</H2>
     <P>Move the numeral 5 from RHS to LHS to achieve <B>150-125 =
  25</B></P>
     <P>For more puzzles surf the net</P>
   </BODY>
</HTML>
```

Figure 7.6 HTML code for figure 7.5

The text placed between <! and > or **<COMMENT>** and **</COMMENT>** is ignored by the browser. This text is called a comment.

Let us now understand the use of each of the elements that have been used to write this code.

Formatting Elements

Text Headings <Hn> Element - This is a container element specified by <Hn> tag, where n is a natural number from 1 to 6. This element is used to give section headings. H1 gives the most prominent heading, while H6 gives the least prominent heading. Heading element introduces a blank line above and below the header text.

Syntax: `<Hn>. . .<Hn>, where n is an integer in the range 1 to 6`

BoldFace <B> element - This is a container element started by <B> and ended by </B> tags. This element is used to display the text enclosed within the tags in bold form.

Syntax: `<B>. . .</B>`

Italics <I> element - This is a container element started by <I> and ended by </I> tags. This element is used to display the text enclosed within the tags in italics form.

Syntax : `<I>. . .</I>`

Underline <U> element - This is a container element started by <U> and ended by </U> tags. This element is used to underline the text enclosed within the tags.

Syntax: `<U>. . .</U>`

Until now, everything has been text based and we have designed some very nice text based web pages. We know, of course, that the World Wide Web (WWW) is much more than just plain text. It is the ability of the web to also provide images that makes it so popular. Images are of different types like pictures, graphics, icons, clip art, etc. Browse the web and you will see all kinds of images. Let us now look at another web page and see how we can set font, font size, font color and images in a web page. When we combine all the tags to design a web page, the web page can be made visually appealing.

Figure 7.7 Web Page with different fonts and alignment

Now we have designed a web page in which we have used different types of fonts, changed colour of the font and used images with different borders and alignment. The web page is looking nice, bright and colourful. Let us now look at the code of the above created web page and try to understand the use of the tags used to create the web page.

```
<HTML>
<HEAD>
<TITLE>Trees: A Renewable Resource</TITLE></HEAD>
<BODY>
<H1 ALIGN="CENTER">Trees A Renewable Resource</H1><BASEFONT
SIZE="4" >
<P><IMG SRC="E:\CBSE Book\HTML\tree2.jpg" BORDER=15 ALIGN="RIGHT"
WIDTH=100 HEIGHT=120>
<COMMENT>Using different fonts and font sizes</COMMENT>
<FONT SIZE="6" FACE="Georgia, Arial" COLOR="MAROON" >T</FONT>rees
are our breathing partners.We need trees in order to live. People
and animals depend on trees and plants for oxygen. As we breathe in,
our body takes in oxygen. As we breathe out, we give out carbon
dioxide. Trees do just the opposite. They take in carbon dioxide and
then release oxygen.</FONT><P>
<P><IMG SRC="E:\CBSE Book\HTML\t2.jpg" BORDER=8 ALIGN="LEFT"
WIDTH=80 HEIGHT=100>
```

```
<FONT SIZE="7" FACE="Forte, Arial" COLOR="GREEN" >T</font>rees are a
renewable natural resource.They can be renewed by replacing the
trees that have been cut  and planting more trees.We also depend on
forest products for things like the wood and paper. Actually,there
are more than 5,000 things that can be made from trees.</FONT></P>
<P><IMG SRC="E:\CBSE Book\HTML\tree2.jpg" BORDER=5 ALIGN="BOTTOM"
WIDTH=60 HEIGHT=80>
<font size="6" face="Century Gothic, Arial" color="RED"
>T</font>oday,  people are aware that our nation's forests and trees
are a valuable resource. It is in our best interest to conserve
them. Forests also provide a natural home to wildlife.Trees and
forests help us by cleaning our air, soil, and water.So let us
pledge to plant a sapling today!!
</FONT></P>
<P><B><FONT FACE="Curlz MT, Helvetica" SIZE="5" COLOR="#006600">
Plant trees to save our future generations.</FONT> </B> </P>
</BASEFONT>
</BODY>
</HTML>
```

Figure 7.8 Code for Web Page Figure 7.7

Image (IMG) element - This is an empty element specified by <IMG> tag. It is used to insert an image in a web page.

Syntax:

```
<IMG SRC="location" BORDER = "border"
    ALIGN = "alignment" HEIGHT = "height" WIDTH = "width">
```

Common attributes of IMG element are discussed below:

1.	**Attribute:**	SRC
	Use:	SRC tells the browser where the source for the image is, that is, the path and name of the image to be inserted. If no path is given, the source is assumed to be the current folder - the folder of the web page. Most browsers only support a few image formats or file types. They are GIF and JPEG.
	Syntax:	`SRC = "path/filename"`

2.	**Attribute:**	BORDER
	Use:	BORDER is used to place (or eliminate) a border of specified width around the image. Border widths are measured in pixels. The BORDER="0" means that we want no border.
	Syntax:	`BORDER = "n"` - **where n is a natural number.**
3.	**Attribute:**	ALIGN
	Use:	ALIGN specifies the alignment of the image according to surrounding contents of the page. Alignment can be middle, top, bottom, left, right. Bottom is the default alignment!
	Syntax:	`ALIGN = "alignment"`
4.	**Attribute:**	WIDTH
	Use:	WIDTH specifies the horizontal width of the image. Width is specified in number of pixels or as the percentage of page width.
	Syntax:	`WIDTH = "n/n%"` - **where n is a natural number**
5.	**Attribute:**	HEIGHT
	Use:	HEIGHT specifies the height of the image. Height is specified in number of pixels or as the percentage of page height.
	Syntax:	`HEIGHT = "n"` - **where n is a natural number**

Know More!

Exactly how big is one pixel? Does it have a specific size? The answer is that a pixel does not have a specific size. Its size depends on a number of things. One of them is the resolution of the monitor being used. A high resolution monitor has a smaller size pixel than a low resolution monitor because a high resolution monitor has more pixels per inch (or centimetre). Therefore the size of your image will vary a little from computer to computer. However, your images (and text) will always be in the right proportion no matter which monitor they are viewed on - and so they will always look good in any browser.

FONT element - It is a container element started by <FONT> tag and ended by </FONT> tag. The FONT element is used to add style, size, and color to the text. We use the size, color, and face attributes to customize our fonts.

Syntax: `<FONT FACE="font name" SIZE = n COLOR = "color" > . . .</FONT>`

Attributes of FONT element are discussed below:

1.	**Attribute:**	FACE
	Use:	FACE is used to specify the name of the font. A list of font names separated by commas can be specified. If the first font is available on the system, it will be used. Otherwise the second will be tried, and so on. If none is available, the default font will be used.
	Syntax:	`FACE = "font name"`
2.	**Attribute:**	SIZE
	Use:	The SIZE attribute is used to specify the font size between 1 and 7 (7 is the largest).
	Syntax:	`SIZE = "n" (n is a natural number between 1 and 7)`
3.	**Attribute:**	COLOR
	Use:	Sets th color of the text. The color can be set by giving the name of the color or its hexadecimal value.
	Syntax:	`COLOR = "color"`

Given below is a list of common font faces which are supported by most browsers.

Arial	Comic Sans MS	Lucida Console
Arial Black	Courier New	Tahoma
Arial Narrow	Georgia	Times New Roman
Bookman Old Style	**Impact**	Trebuchet MS
Century Gothic	Mangal	Verdana

In the web pages designed so far the text was displayed in the form of a paragraph, but if we want to communicate some information in the form of bulleted/numbered lists then we have to use the LIST element. A page designed using list elements is shown below:

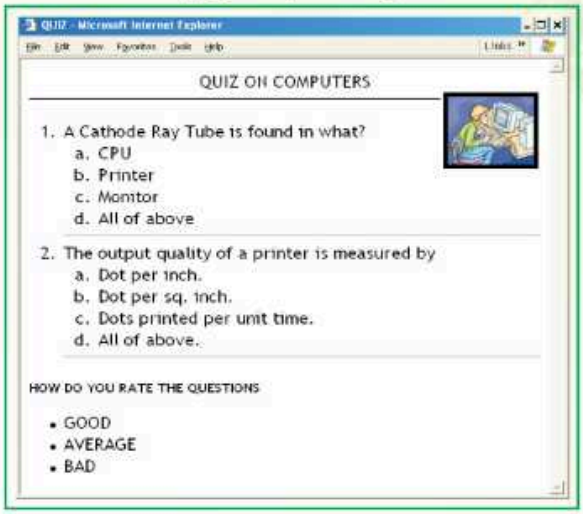


Figure 7.9 Web Page designed using Lists

Let us now look at the code for the web page shown in figure 7.9.

```
<HTML>
   <HEAD> <TITLE>QUIZ</TITLE> </HEAD>

  <BODY>
    <FONT FACE = "Times New Roman" size = "4">
    <CENTER>QUIZ ON COMPUTERS</CENTER>
    <IMG BORDER="5" SRC="QUIZ.jpg" ALIGN="RIGHT">
    <HR color="BLACK">
    <COMMENT>Beginning of first ordered list</COMMENT>
    <OL START=1>
      <LI>A Cathode Ray Tube is found in what?
      <COMMENT>Beginning of second ordered list</COMMENT>
        <OL START=1 TYPE=a>
          <LI>CPU
          <LI>Printer
          <LI>Monitor
          <LI>All of above
```

```
        </OL>
        <HR>
       <LI>The output quality of a printer is measured by
        <OL START=1 TYPE=a>
          <LI>Dot per inch.
          <LI>Dot per sq. inch.
          <LI>Dots printed per unit time.
          <LI>All of above.
        </OL>
        <HR>
    </OL>
    <H5>HOW DO YOU RATE THE QUESTIONS</H5>
    <UL TYPE=CIRCLE>
      <LI>GOOD
      <LI>AVERAGE
      <LI>BAD
      </UL>
     </FONT>
  </BODY>
  </HTML>
```

Figure 7.10 Code for Figure 7.9

Lists - Lists help us to organize the contents on the web page in an ordered and sequential manner. HTML supports three types of lists:

- Unordered lists (bulleted lists)
- Ordered lists (numbered)
- Definition Lists

In this chapter we shall study Unordered Lists and Ordered Lists only.

Ordered List (OL) element - It is a container element started by <OL> tag and ended by </OL> tag. It displays a numbered list. In a numbered list each item is preceded by a number or a letter. This element is used where the items are to be placed in a specific order.

Syntax: `<OL START = "n" TYPE = "A"/"a"/"I"/"i"/"1" > . . . </OL>`

Attributes of OL element are discussed below:

1.	**Attribute:**	START
	Use:	START indicates the starting number (or the serial number of the first element) of the list. This number must be a positive integer.
	Syntax:	START = "n" - where n is a natural number.
2.	**Attribute:**	TYPE
	Use:	TYPE defines the type of numbering sequence used for the list items. "A" specifies a sequence of uppercase letters. "a" specifies sequence of lowercase letters. "I" specifies a sequence of uppercase Roman numerals. "i" specifies a sequence of lowercase Roman numerals. "1" specifies a sequence of numbers.
	Syntax:	TYPE = "A"/"a"/"I"/"i"/"1"

Unordered List (UL) element - It is a container element started by <UL> tag and ended by </UL> tag. It displays a bulleted list. In a bulleted list each item is preceded by a small symbol called a bullet. The shape of the bullet can be a Circle, a Disk, or a Square.

Syntax: `<UL Type="value">. . . </UL>`

Attributes of UL element are discussed below:

1.	**Attribute:**	TYPE
	Use:	TYPE defines the type of bullet used for each list item. The value can be: o CIRCLE specifies a hollow bullet. " DISC specifies a solid round bullet. " SQUARE specifies a square bullet.
	Syntax:	`START = "value"`

List Item (LI) element - The LI element is an empty element specified by <LI> tag. It is used inside OL and UL elements to define list items. Each list item has to be preceded by <LI> tag.

Syntax: **<LI> Item Name**

By itself (if used without <OL> or <UL>), <LI> displays items in a bulleted list form. It means that we can use LI element without OL or UL element also. But it is advised not to do so.

Nested Lists - Nested lists occur when we put one list inside another list. The nested list would be a list item (marked by <LI>) of the parent list.

Can you write the code for the nested list as shown below:

1. **Book Mathematics**
   - a. **Chapter 1**
   - b. **Chapter 2**
   - c. **Chapter 3**
2. **Book English**
   - a. **Chapter 1**
   - b. **Chapter 2**
   - c. **Chapter 3**
3. **Book Science**
   - a. **Chapter 1**
   - b. **Chapter 2**
   - c. **Chapter 3**

Tables - Let us now design a web page where we will display data in an organized manner with the use of tables. Tables are very useful in the communication of data in a meaningful way. Two examples of tables are shown below:

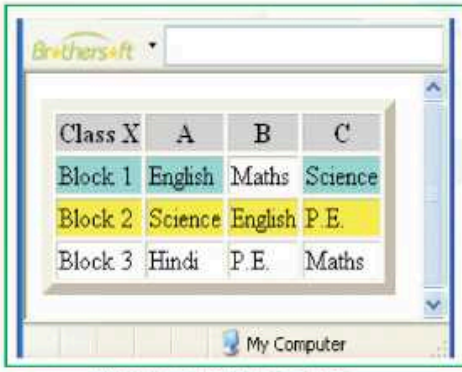


Figure 7.11 Time Table

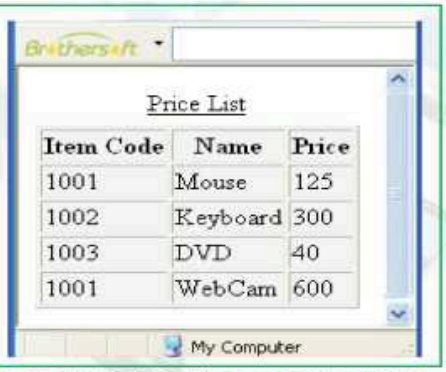


Figure 7.12 Price List made using table

Let us now look at the code for designing these web pages using tables:

```
<HTML>
 <BODY>
   <TABLE BORDER="7" CELLPADDING="7" CELLSPACING="10">
   <COMMENT>Beginning of table row</COMMENT>
     <TR BGCOLOR="#C0C0C0">
       <TD>Class X </TD>
       <TD> A</TD>
       <TD> B</TD>
       <TD> C</TD>
     </TR>
   <COMMENT>Beginning of second table row</COMMENT>
     <TR BGCOLOR="#00FFFF">
       <TD> Block 1</TD>
       <TD>English</TD>
       <TD BGCOLOR="#FFFFFF">Maths</TD>
       <TD>Science</TD>
     <TR BGCOLOR="#FFFF00">
       <TD> Block 2</TD>
       <TD>Science</TD>
       <TD>English</TD>
       <TD ROWSPAN="2">P.E.</TD>
     </TR>
     <TR BGCOLOR="#FFFFFF">
       <TD> Block 3</TD>
       <TD colspan=2>Hindi</TD>
      </TR>
   </TABLE>
 </BODY>
 </HTML>
```

Figure 7.13 Code for Time Table (Figure 7.11)

```
<HTML>
  <BODY>
    <TABLE BGCOLOR="#EEEEEE" BORDER="1">
    <CAPTION> <u>Price List</u> </CAPTION>
    <TR>
      <TH>Item Code</TH>
      <TH>Name</TH>
      <TH>Price</TH>
    </TR>
    <TR>
      <TD>1001</TD> <TD>Mouse</TD> <TD>125</TD>
    </TR>
    <TR>
      <TD>1002</TD> <TD>Keyboard</TD> <TD>300</TD>
    </TR>
    <TR>
      <TD>1003</TD> <TD>DVD</TD> <TD>40</TD>
    </TR>
    <TR>
      <TD>1001</TD> <TD>WebCam</TD> <TD>600</TD>
    </TR>
  </TABLE>
  </BODY>
  </HTML>
```

Figure 7.14 Code for Price List (Figure 7.12)

Let us now try and understand the code:

TABLE element - It is a container element started by <TABLE> tag and ended by </TABLE> tag. It is the basic element for creating a table.

Syntax:

```
<TABLE ALIGN="Alignment" BGCOLOR="colorname" BORDER="n"
                    >. . .</TABLE>
```

Attributes of TABLE how element are discussed below:

1.	**Attribute:**	ALIGN
	Use:	Specifies the horizontal placement of the table in relation to the window screen. LEFT aligns the table to the left (the default). RIGHT aligns the table to the right. CENTER aligns the table to the centre.
	Syntax:	`ALIGN = "Alignment"`
2.	**Attribute:**	BGCOLOR
	Use:	Sets the colour of the background for the table.
	Syntax:	`BGCOLOR = "colorname"`
3.	**Attribute:**	BORDER
	Use:	Sets the border size (width of the boundaries) of the table.
	Syntax:	`BORDER = "n"  - where n is a natural number.`

TABLE ROW (TR) element - It is a container element started by <TR> tag and ended by </TR> tag. It is used to define table rows. The TR element is used inside the TABLE element. The number of rows in a table corresponds to the instances of the TR element within the table element. The contents of a row are given as a parameter to the TR element. The contents of a table row may contain TH tags, which indicate table headings, and TD tags, which indicate table data.

Syntax:

```
<TR ALIGN ="Alignment" VALIGN="VAlignment"
                BGCOLOR="colorname"> . . . < / TR >
```

Attributes of TABLE ROW element are discussed below:

1.	**Attribute:**	ALIGN
	Use:	This controls whether the text in a row is aligned to the left, right, or centre of a cell. Default alignment is Left.
	Syntax:	`ALIGN = "Alignment"`
2.	**Attribute:**	VALIGN
	Use:	This controls whether the text in a row is aligned to the top, bottom, or middle of a cell. BOTTOM aligns the content with the cell's bottom. MIDDLE centres the content within the cell (the default). TOP aligns the content with the cell's top.
	Syntax:	`VALIGN = "VAlignment"`
3.	**Attribute:**	BGCOLOR
	Use:	It sets the background colour of the table row.
	Syntax:	`BGCOLOR = "colorname"`

TABLE DATA (TD) element - It is a container element started by <TD> tag and ended by </TD> tag. It is used to specify the text in a cell of the table. The TD tag is used inside the TR tag, which is inside the table tag. The number of columns in a table depends on the number of TD elements within the TR element.

Syntax:

```
<TD ALIGN="Alignment" VALIGN="VAlignment"
        BGCOLOR="colorname" > . . . </TD>
```

The attributes ALIGN, VALIGN, and BGCOLOR have their usual meaning as defined in <TR> element.

If used the attributes of the <TD> tag will replace the attributes of the <TR> tag.

TABLE HEADING (TH) element - It is a container element started by <TH> tag and ended by </TH> tag. It is used to create header values. The header values are displayed in a bold font and centre aligned. The TH element is used to create column or row headings.

Syntax:

```
<TH ALIGN ="Alignment" VALIGN = "VAlignment"
    BGCOLOR = "colorname" NOWRAP >. . .</TH>
```

- The attributes ALIGN, VALIGN, and BGCOLOR have their usual meaning as defined in <TR> element.
- NOWRAP - Specifies that the contents of a cell cannot be broken, i.e., they do not wrap onto the next line.

So far we have studied about the usual HTML commands like <HTML>, <HEAD> and <BODY>. We have also learnt how to create lists and tables. But there is a limitation in all the web pages that we have created, they are not interactive. When we view our document in a web browser, we can merely view the information, we cannot interact with the web page.

FORMS in HTML - We can make a web page interactive by creating form(s) in it. A form allows the user to enter some data and this data can then be sent to a web server or to another web page to respond accordingly. Forms in HTML are used to handle operations like taking orders, conducting surveys, user registration etc. Let us now design a web page which is interactive wherein the user can enter some data. Look at the web page shown below:

Let us now write the code to make the web page shown above.

Figure 7.15 A Web Page containing a Form

```
<HTML>
   <HEAD> <TITLE> USING FORMS </TITLE> </HEAD>
   <BODY>
     <CENTER>
     <FONT face="Courier New" size=5 color=red>FREE MAIL
 INDIA.COM</FONT>
     <br>
     <FONT size=3>
       <I>Get a FREE MAIL INDIA! ID and connect to people</I>
     </FONT>
 <COMMENT>Beginning of form tag</COMMENT>
     <FORM NAME="FORM1" ACTION = "validate.html" METHOD = "GET">
       <B> Personal Details </B> </CENTER>
       Enter your name:
       <INPUT TYPE="TEXT" SIZE=30 NAME="t1"> <br>
       Gender. . . . . .:
       <INPUT TYPE="RADIO" NAME="r1">  MALE
       <INPUT TYPE="RADIO" NAME="r1">  FEMALE <br>
       Country . . . . . .:
       <INPUT TYPE="TEXT" SIZE=30 NAME="t2"> <br><br>
       <B><CENTER>Select an ID and password:</CENTER></B>
       Desired Email ID . :
       <INPUT TYPE="TEXT" SIZE=30 NAME="t3"> <BR>
       Password . . . . . .:
       <INPUT TYPE="PASSWORD" SIZE=20 NAME="t4">
       <FONT size=2><I> minimum 5 characters</I></FONT> <br>
       Re-type password :
       <INPUT TYPE="PASSWORD"  SIZE=18 NAME="t5"> <br><br>
       <CENTER><INPUT TYPE="SUBMIT" value="Create MyAccount"
 name="s1"></CENTER>
     </FORM>
   </BODY>
 </HTML>
```

Figure 7.16 Code for web Page containing Form tag

We have used a FORM element to create the form. We have used interface element INPUT with different values of TYPE attribute such as TEXT, RADIO, PASSWORD, and SUBMIT, to get the user input. All the elements in the form must be defined between the <FORM> and </FORM> tags. Other elements, such as heading, paragraph, and tables, etc., can also be used in the <FORM> and </FORM> tags.

Thus, when the form is displayed in a web browser, the user can make choices and text can be entered by using the interface elements defined with <INPUT> tag. The form made

can finally be submitted to a destination point whenever required. All these things will be discussed one by one as we progress.

FORM element - It is a container element started by <FORM> tag and ended by </FORM> tag. It is used to create a form on a web page.

Syntax: `<FORM NAME="FormName" ACTION = "URL" METHOD = "method">`

Attributes of FORM element are discussed below:

1.	**Attribute:**	NAME
	Use:	This specifies the name of the form. But this name will not be displayed on the form. As there can be more than one FORMs in an HTML document, a name is required to differentiate one form from another. The NAME attribute is optional if there is only one FORM on the web page.
	Syntax:	NAME = "FormName"
2.	**Attribute:**	ACTION
	Use:	This specifies the URL where the form-data is sent when the form is submitted. This URL is also called the destination of the form.
	Syntax:	ACTION = "URL"
3.	**Attribute:**	METHOD
	Use:	This specifies how the form-data is submitted. Form-data can be submitted using the methods get or post. With METHOD = "get", the form-data is submitted as URL variables, and with METHOD = "post", the form-data is submitted as HTTP post.
	Syntax:	METHOD = "method"

INPUT element - It is an empty element specified by <INPUT> tag. It is used to provide an input field in a form where the user can enter the data. An input field may be a textfield, a checkbox, a radio button, a button, and more. INPUT element is always used within the FORM element. Thus, INPUT element defines an object on the FORM which can receive user input.

Syntax: `<INPUT TYPE="FieldType" NAME="FieldName" VALUE="FieldText">`

Attributes of INPUT element are discussed below:

1.	**Attribute:**	TYPE
	Use:	The TYPE attribute determines the field type of input field to be provided in the form. Field type can be one of the following: TEXT, PASSWORD, RADIO, CHECKBOX, SUBMIT, RESET, BUTTON, IMAGE, HIDDEN, FILE.
	Syntax:	TYPE = "FieldType"
2.	**Attribute:**	NAME
	Use:	Specifies the name of the field. This name does not appear on the FORM. It is required for the identification/ differentiation as there can be more than one fields in a single FORM.
	Syntax:	NAME = "FieldName"
3.	**Attribute:**	VALUE
	Use:	Specifies the text to be displayed on the field.
	Syntax:	VALUE = "FieldText"

We will now discuss different field types (values of TYPE attribute) relevant to our syllabus.

1.	**FieldType: TEXT** Use: It provides a single line text input field where the user can enter text. **Additional Attributes:** SIZE = "n" - Sets the visible size of the text field to n characters. MAXLENGTH = "n" - Set the maximum number of characters that can be input in the text field to n.
2.	**FieldType: PASSWORD** **Use:** It provides a single line text input field where the user can enter password. A password field is different from a text field because a text field displays whatever characters are entered by the user whereas a password field shows one dot for each character input by the user. This is to prevent others from seeing the password. **Additional Attributes:** Same as those for TEXT field.
3.	**FieldType: RADIO** **Use:** It provides a radio button on the form. More than one radio buttons can have (and in general have) the same name. All the radio buttons that have the same name constitute a radio group. Only one radio button of a group can be selected at one time. That is, from a group of radio buttons, if the user selects a button, all the other buttons in the set are deselected. When a form is submitted, selected radio button's value (specified by the VALUE attribute) is submitted to the destination. **Additional Attributes:** **CHECKED** - Indicates that the radio button is selected, which can be deselected when another choice is made. At one time, only one radio button in a radio group can be specified as CHECKED.

4.	**FieldType: CHECKBOX** **Use:** It provides a check box on the form. With checkboxes, we can give the users a list of items to choose from. The user can choose more than one items from the list. We can make a group of checkboxes, by giving them the same name. When a form is submitted, selected cheboxes' values (specified by the VALUE attribute) are submitted to the destination. **Additional Attributes:** **CHECKED -** Indicates that the checkbox is to be displayed with a tick mark to show selection. This attribute is optional.
5.	**FieldType: SUBMIT** **Use:** This provides a button on the form. When this button is clicked, the form is submitted to the destination. **Additional Attributes:** None -
6.	**FieldType: RESET** **Use:** This provides a button on the form. When this button is clicked, the input fields on the form are reset to their default state. **Additional Attributes:** None -

XML

XML (eXtensible Markup Language) is also a markup language like HTML. But XML is different from HTML in the sense that HTML describes how to display and format the data, text and images in the browser whereas XML is used to describe the data. XML has nothing to do with presentation of data in the browser. The XML standard was created by W3C to provide an easy to use and standardized way to store self-describing data (Self-describing data is data that describes both its content and its structure).

The main benefit of XML is that it can be used to share data between two entirely different platforms. For example, we can take data from a database like MySQL, convert it into XML, and then share it with any other platform like MS Excel, HTML etc. Each of these receiving platforms can then convert the XML into a structure the platform uses normally. This way communication between two potentially different platforms is achieved using XML.

To store self-describing data, XML allows us to create our own tags. Once an XML file is created with the help of user-defined tags, it needs to be transformed into the target platform format. We shall take an example in which data in an XML file is transformed into HTML format. The file used for this purpose is an XSLT (eXtensible Stylesheet Language Transformations) file.

Example

Following is an XML code to store the data of a class - names of three students and one teacher. You can type this code in any text editor and save the file with any name and extension XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="class.xsl"?>
<class>
<student>Sumedha</student>
<student>Utkarsh</student>
<student>Tushar</student>
<teacher>Anil Kumar</teacher>
</class>
```

Figure 7.17 XML Code

Let us try to understand this code :

- **<?xml version="1.0" encoding = "UTF-8"?>**

  The first line in the code is the XML declaration which should always be

included. This is called the 'XML prolog'. The prolog in this example specifies that the XML code confirms to version 1.0 of XML standard, and the encoding scheme used is "UTF-8".

- **<?xml-stylesheet type="text/xsl" href="class.xsl"?>**

  This statement links this xml file to the file class.xsl. It means that the file class.xsl (shown later) will use this xml file and apply its rules to transform this XML document.

- **<class> . . . </class>**

  Each XML document contains one and only one root element (or parent element) which encapsulates the data described in the XML file. In this example <class> . . . </class> is the root element. The root element further contains sub-elements (or child elements) which describe the actual data values. In this example <student> . . . </student> and <teacher> . . . </teacher> are the sub-elements of <class> . . . </class>.

So, the above XML file stores the data of a class which has:

3 students: Sumedha, Utkarsh, and Tushar

1 Teacher: Anil Kumar

This XML file data will be transformed and represented in HTML format by the file class.xsl. This file name was specified in the XML code itself. The code of class.xsl is given below. You can type this code in a text editor and save the file as class.xsl. class.xsl becomes an XSLT (Extensible Stylesheet Language Transformations) file.

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="teacher">
            <p><u><xsl:value-of select="."/></u></p>
    </xsl:template>
```

```
        <xsl:template match="student">
                <p><b><xsl:value-of select="."/></b></p>
        </xsl:template>
        <xsl:template match="/">
                <html>
                <body>
                <xsl:apply-templates/>
                </body>
                </html>
        </xsl:template>
</xsl:stylesheet>
```

Figure 7.18 Class.xsl code

Don't worry if this code is not clear to you. At this stage our aim is to understand the concept of data storage in an XML file and not to understand the syntax of XSLT files.

Now if you open the above XML file in a browser, you will see the data in the following format:

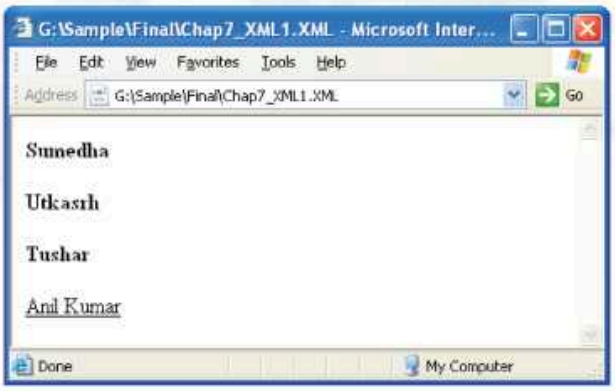


Figure 7.19 Output of XML file

Suppose Class.XSL has the following code:

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="teacher">
        <p>Teacher:<u><xsl:value-of select="."/></u></p>
    </xsl:template>
    <xsl:template match="student">
        <p>Student: <b><xsl:value-of select="."/></b></p>
    </xsl:template>
    <xsl:template match="/">
        <html>
        <body>
        <xsl:apply-templates/>
        </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

Figure 7.20 Code for XSL sheet

Then the output would be:

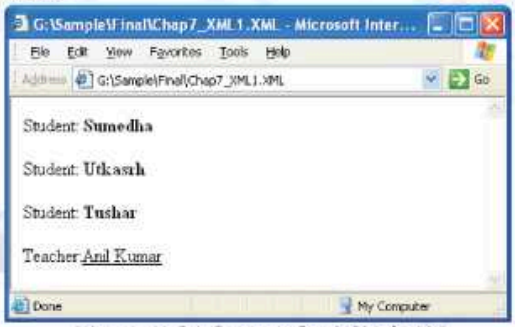


Figure 7.21 Output for XSL sheet

So the same data in an XML file can be represented in different ways. It all depends on how the data is transformed and presented.

The main difference between XML and HTML

- HTML is used to display data and to focus on formatting of data, whereas XML is used to describe data and focus on what data is.
- HTML tags are predefined, whereas XML tags are not predefined. We create our own tags.
- HTML tags are not case sensitive, whereas XML tags are case sensitive.

Features of XML

- XML is extensible. XML allows the user to create his own tags and document structure.
- XML can be used to store data. Applications can be written to store and retrieve information.
- XML can be used to exchange data. In the real world, databases contain data in different formats. It is difficult to exchange data between such systems over the Internet. Converting the data to XML can help in solving this problem and create data that can be read by different types of applications.
- XML is free. It can be written with a simple text editor or one of the many freely available XML authoring tools, such as XML Notepad.
- XML is a W3C recommendation.

Future Trends

In the last fifteen years internet has reinvented itself. Nowadays we shop, bank, work, meet people online and at times share what we are doing at any given moment of time. Technology is always evolving - and none quite as fast as the Internet. Let us try to visualize the future trends of internet.

Audio Web Surfing

In future we might see people traveling with headphones attached to their mobile devices while a Text-to-Speech application reads them the latest articles from their favorite sites. Browser will be able to comprehend the spoken command and open the particular website which the user has requested for. Audio surfing could be useful for commuters, children learning to read, tutors etc. For web developers, there may be new accessibility opportunities especially for multi-lingual sites.

Integration of data and applications

In future with the help of development in semantic web technologies we will click on the web page for the meeting, and our computer, will be able to comprehend it as a form of appointment, will pick up all the right information, and understand it and send it to all the right applications. Further, it will evoke those applications directly (using web services) needing little or no human intervention.

Web Accessibility for all

The W3C Web Accessibility Initiative (WAI) brings together people from industry, disability organizations, government, and research labs from around the world to develop guidelines and resources to help make the Web accessible to people with disabilities like auditory, cognitive, neurological, physical, speech, and visual disabilities.In future hardware might be adaptable for people with motoric disabilities. Software accessibility solutions for the future might include physical modelling. For instance in case of sight disabled there are no or little audio feedback from the games where concepts of physics are involved.In the future such games might be connected with a sound engine or realtime audio.

Summary

- Web server delivers (serves) content, such as web pages, using the Hypertext Transfer Protocol (HTTP), over the World Wide Web.
- A web browser is a client that initiates communication by making a request for a specific resource. The server then responds with the content of that resource, or an error message if unable to do provide the contents due to any reason.
- URL represents a Uniform Resource Locator, a pointer to a "resource" on the World Wide Web. A resource can be a file or a web page.
- Various HTML tags are summarised in the following table:

Structure Elements			
Element	**Start Tag**	**Attributes**	**End Tag**
HTML File	<html>	none	</html>
File Header	<head>	none	</head>
File Title	<title>	none	</title>
Comment	<! . . .> <Comment>	Your comments go between <! and >.	</Comment>
Body	<body>	background, bgcolor, text, link, alink, vlink	</body>

Basic Text Elements			
Element	**Start Tag**	**Attributes**	**End Tag**
Line Break	 	none	
Paragraph		align	
Bold		none	
Italic		none	
Headline	<hn>	align	</hn>
Font		Face, size, color	
Horizontal Rule	<hr>	Size, width, noshade, align	

List Elements			
Element	**Start Tag**	**Attributes**	**End Tag**
Unordered List		type	
Ordered List		type, start	
List Item		none	

Image Element			
Element	**Start Tag**	**Attributes**	**End Tag**
Image	<img>	src, align, width, height	</img>

Table Elements			
Element	**Start Tag**	**Attributes**	**End Tag**
Table	<table>	Align, bgcolor, border	</table>
Table Row	<tr>	Align, valign, bgcolor	</tr>
Table Data	<td>	Align, valign, bgcolor	</td>
Table Header	<th>	Align, valign, bgcolor	</th>
Caption	<caption>	Align, valign, bgcolor	</caption>

Form Elements			
Element	**Start Tag**	**Attributes**	**End Tag**
Form	<form>	name, action, method	</form>
Input Field	<input>	Name, type	</input>

- XML stands for eXtensible Markup Language. It is a markup language like HTML but its is used to store the data and not to describe the presentation of data.
- Using XML data can be shared among various platforms.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. The address of a resource on the net is known as:

   (a) ISP (b) HTTP

   (c) URL (d) WWW

2. A program that serves requested HTML files and pages.

   (a) Web Address (b) Web Page

   (c) Web Server (d) None of these

3. HTML tags must be written within:

   (a) < > (b) { }

   (c) [ ] (d) ( )

4. Which of the following is the correct structure of HTML tags?

   (a) < HTML> </HTML> <HEAD> </HEAD> <BODY> </BODY>

   (b) <HTML> <HEAD> </HEAD> </HTML> <BODY> </BODY>

   (c) <HTML> <HEAD> <BODY> </BODY> </HEAD> </HTML>

   (d) <HTML> <HEAD> </HEAD> <BODY> </BODY> </HTML>

5. Which of the following tags is used to specify the items in a list.

   (a) <LI> (b) <OL>

   (c) <UL> (d) <DL>

6. The IMAGE tag uses the _____ attribute to specify the URL of the image to be displayed.

   (a) SCR (b) SRC

   (c) Source (d) None of these

7. The align attribute of <Table> tag refers to _____ placement of the table in relation to the window screen.

   (a) Vertical (b) horizontal

   (c) Both a and b (d) None of these

8. Choose the best suitable input type to input gender from the user:

   (a) Text (b) Submit

   (c) checkbox (d) Radio

9. Elements, such as heading, paragraph and tables etc can be contained in the FORM element.

   (a) Holds True always (b) Holds True Sometimes

   (c) Is Never True (d) None of these

10. Data entry on a web page can be done using:

    (a) Tables (b) Formatting Tags

    (c) Forms (d) Lists

11. Which tag is used to embed an image in an HTML document.

    (a) <FIX> (b) <IMG>

    (c) <IMAGE> (d) <FIX IMAGE>

12. To create a nested list we use the tag:

    (a) <OL> (b) <UL>

    (c) <NL> (d) combination of (a) and (b) as required.

13. Input type="_____" will send the form to its destination place.

    (a) Button (b) File

    (c) Submit (d) Reset

14. FACE is the attribute of which tag:

    (a) <BODY> (b) <FONT>

    (c) <P> (d) <IMG>

15. Radio buttons can be grouped together so that only one is selected at a time by using the attribute:

    (a) name (b) selected

    (c) checked (d) font

16. XML document is used to

    (a) Only interpret data

    (b) Store any kind of data

    (c) Store only highly structured data (like in databases)

    (d) Store only loosely structured data (like letters)

17. XML can be used to

    (a) Exchange data (b) Store data

    (c) Interpret data (d) All of these

18. Every XML document must begin with a

    (a) Root element (b) Child element

    (c) XML version details (d) Any of these

ANSWER THE FOLLOWING QUESTIONS

1. Explain the basic structure of an HTML document with the help of an example.
2. Explain with an example the difference between container and empty elements.
3. What is the importance of the BODY element? List the commonly used attributes within the <BODY> tag.
4. What is the purpose of the TITLE element in HTML?
5. Explain the src and align attribute of the IMG tag.
6. What is the difference between the <TD> and <TH> tags?
7. Explain align and bgcolor attributes of the <TR> tag.
8. How can we create an HTML page that allows the user to enter his details on that webpage? Explain with the help of suitable HTML code/example.
9. Mention the tags that are required to be used to create the given table.

Student Details

Name	Age	Gender
Nishi	14	Female
Rajat	16	Male

10. Differentiate between HTML and XML.
11. Write a few features of XML.

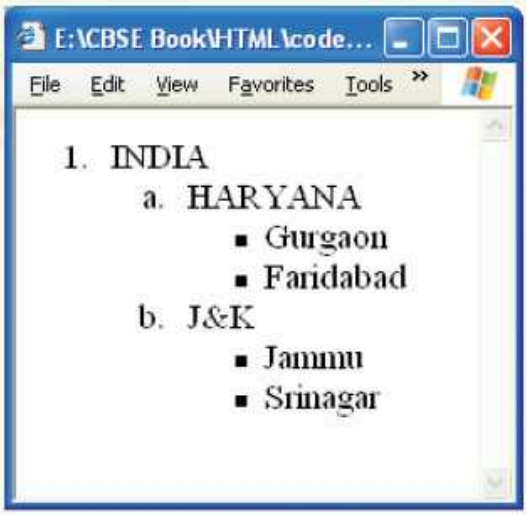
LAB EXERCISES

1. Write HTML code to display the following web pages:

a)

b)

d)



2. Write the HTML code to generate the following result:

a) ONE 1 2 3

b) TWO

ii. ONE

iii. TWO

iv. THREE

c) Things we learned in kindergarten:

1. share
2. play fair
3. don't hit people
4. put things back where we found them
5. say sorry when we hurt somebody

d) 1. Seek expert advice about the area

- Get the best maps. On the map select
  - landmarks
  - mountains
  - lakes
- Get a good compass and
  - check slope of land
  - check direction of flowing streams

2. If there is snow on the ground, stay close to:

- roads
- trails and
- waterways

3. Give the output for the following:

a)
```
< HTML >
< BODY >
<H1>The following is a list of a few chemicals: </H1)
<UL type="disc">
<LI> Sodium
<LI> Sulphur
<LI> Magnesium </UL>
</BODY>
</HTML>
```

b)
```
<HTML>
<HEAD><TITLE>SCHEDULE</TITLE></HEAD>
<BODY>
<TABLE BORDER="3">
<CAPTION>VOLUNTEER SCHEDULE</CAPTION>
<TR BGCOLOR="#AAFFCC">
    <TH>9 A.M.</TH> <TH>12 P.M.</TH> <TH>2 P.M.</TH>
</TR>
<TR>
    <TD>Anmol </TD> <TD>Abhinav </TD> <TD>Anika </TD>
</TR>
<TR>
    <TD>Riti </TD> <TD>Riya </TD> <TD>Sara </TD>
</TR>
<TR>
    <TD>Gul </TD> <TD>Reyana  </TD> <TD>Swayam </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

TEAM BASED TIME BOUND EXERCISE

(Team size recommended: 3 students each team)

Web Page Checklist

1. Students should perform search on a topic, such as: "nuclear armageddon" , "global warming" etc. and ask the team to give a presentation of about 3 minutes each.

2. The team should use this Checklist to try to evaluate systematically some of the search results.

Checklist Parameters	Website 1	Website 2
URL	_____	_____
Title	_____	_____
Personal / Oraganisational / Anonymous	☐ Personal ☐ Organisational ☐ Anonymous	☐ Personal ☐ Organisational ☐ Anonymous
What type of domain is it?	☐ com ☐ org/net ☐ edu ☐ gov/mil/us ☐ in ☐ other	☐ com ☐ org/net ☐ edu ☐ gov/mil/us ☐ in ☐ other
Does it correspond to the name of the site?	☐ Yes ☐ No	☐ Yes ☐ No
Appropriate for the content?	☐ Yes ☐ No	☐ Yes ☐ No
Navigation	☐ Simple ☐ Complex	☐ Simple ☐ Complex
All links on homepage work	☐ Yes ☐ No	☐ Yes ☐ No
Contact details	☐ Available ☐ Not Available	☐ Available ☐ Not Available

FIND OUT INFORMATION ON THE FOLLOWING TOPICS

- **Situation:** You have just acquired stocks of an IT firm. Find out how the stock is doing and find a copy of the annual report.
- **Situation:** Find information about Internet censorship? The class can be divided into 6 groups three groups can write in its favor and the rest can write against the topic.

MySQL - REVISION TOUR

Learning Objectives

After studying this lesson the students will be able to:

- Recall the terminology used in RDBMS concepts.
- Recall and effectively use the SQL commands and clauses learnt in class XI.
- Realize that much more is to be learnt for effective use of databases.

In the previous class, you have learnt some database concepts and SQL commands. You have also learnt how to access data from a table using various clauses of SELECT command. In this lesson we are going to revise the database concepts and SQL studied in class XI. That will make the base for further reading of the concepts and some more SQL commands which are really very useful in practical applications.

Database Concepts

You have already studied the following database concepts in class XI:

1. **Database:** A database is an organised collection of data.

   Software used to manage databases is called Data Base Management System (DBMS).

2. **Relational Database:** A database in which the data is stored in the form of relations (also called tables) is called a Relational Database. In other words a Relational Database is a collection of one or more tables.

3. **RDBMS:** A DBMS used to manage Relational Databases is called an RDBMS (Relational Data Base Management System). Some popular RDBMS software available are: Oracle, MySQL, Sybase, Ingress

4. **Benefits of using a DBMS are:**
   a. Redundancy can be controlled
   b. Inconsistence can be avoided
   c. Data can be shared
   d. Security restrictions can be applied.
5. **MySQL:** It is an Open Source RDBMS Software. It is available free of cost.
6. **Relation/Table:** A table refers to a two dimensional representation of data arranged in columns (also called fields or attributes) and rows (also called records or tuples). The tables in a database are generally related to each other to facilitate efficient management of the database. Interrelated tables also reduce the chances of errors in the database.
7. **Key:** A column or a combination of columns which can be used to identify one or more rows (tuples) in a table is called a key of the table.
8. **Primary Key:** The group of one or more columns used to uniquely identify each row of a relation is called its Primary Key.
9. **Candidate Key:** A column or a group of columns which can be used as the primary key of a relation is called a candidate key because it is one of the candidates available to be the primary key of the relation
10. **Alternate Key:** A candidate key of a table which is not made its primary key is called its Alternate Key.

Let us now revise MySQL concepts studied in class XI with the help of a Library database.

> ***Note:** As this is a revision lesson, all variations of SQL commands are not covered here. It is suggested that they are revised through practical assignments.*

Think of the functioning of a computerised school library.

In a school library, there are books which are issued to the students and the staff. These books are also returned after a certain period of time. For proper functioning of this system the librarian maintains a database (an organised collection of data) which contains the relevant data of books, students, and staff. In a relational database, this data may be organised as a collection of the following tables:

BOOKS

AccNo	Title	Author	Publisher	Edition	Price
1001	Basic Economics	J.P. Mudgil	APH	2009	125
1002	MicroEconomics	Kavi Prakash	KPH	2009	175

AccNo is Accession Number of a book.

STUDENTS

AdmNo	LibCardNo	AccNo	IssueDate	ReturnDate
2010001	101	1002	2009-12-02	2009-12-08
2009012	102	1050	2010-02-16	NULL

STAFF

EmpNo	AccNo	IssueDate	ReturnDate
1	1001	2009-04-13	NULL
2	NULL	NULL	NULL

These three related tables collectively form the **Library** database.

Using the data stored in these tables the librarian can find out any type of relevant information regarding the books at any time. For example, she can find out:

- The list of books in the library which are published by any specific publisher or author.
- The list of books with price in a specified range.

- Admission numbers of the students who have not returned the books issued to them.
- The names of books issued to a particular staff member or student.

For the above tables the primary keys may be set as follows:

Table	Primary Key	Reason
BOOKS	AccNo	No two books have the same accession number.
STUDENTS	AdmNo	No two students have the same admission number.
STAFF	--	As there is no column for which each row has a unique value, there is no primary key of this table.

In the STUDENTS table LibCardNo can also be the primary key (Why?). It means that there are two columns in the table STUDENTS that are unique for each row and can be set as the primary key of the table. These two columns are *candidate keys* as they are the candidates for primary key. Out of these two candidate keys we have chosen AdmNo as the primary key. The other candidate key, i.e., LibCardNo, becomes the Alternate Key of the table. Had we chosen LibCardNo as the primary key, AdmNo would have been called the alternate key of the table.

In class XI we have studied that **degree** of a table is the number of columns in the table, and the number of rows in a table is called its **cardinality**.

Can you tell what are the degrees and cardinalities of tables of Library database?

Libraries function without computers also. Is there any advantage of setting up a computerised Library Database using an RDBMS? Discuss it with your teacher.

We shall be using MySQL for Library database.

Once MySQL is procured (downloaded from net, or copied from somewhere, or received free with some book), it can be installed and the process of creating Library database can be started.

Working with MySQL

You have already studied the following SQL concepts in class XI:

1. **SQL (Structured Query Language):** It is the language used to manipulate and manage databases and tables within them using an RDBMS.
2. **DDL (Data Definition Language):** This is a category of SQL commands. All the commands which are used to create, destroy, or restructure databases and tables come under this category. Examples of DDL commands are - CREATE, DROP, ALTER.
3. **DML (Data Manipulation Language):** This is a category of SQL commands. All the commands which are used to manipulate data within tables come under this category. Examples of DML commands are - INSERT, UPDATE, DELETE.
4. **DCL (Data Control Language):** This is a category of SQL commands. All the commands which are used to control the access to databases and tables fall under this category. Examples of DCL commands are - GRANT, REVOKE.

Different SQL commands studied in class XI are summarized below:

SNO	Command, Syntax, and Purpose	Category
1.	**Command:** CREATE DATABASE **Syntax:** CREATE DATABASE <databasename>; **Purpose:** Creates a database with the specified name.	DDL
2.	**Command:** CREATE TABLE **Syntax:** CREATE TABLE <tablename> (<column name1> <data type1> [,<column name2> <data type2>, . . <column nameN> <data typeN>]); **Purpose:** Creates a table with the specified name.	DDL

3.	**Command:** **Syntax:** **Purpose:**	ALTER TABLE ALTER TABLE <tablename> ADD <columnname> <datatype>; ALTER TABLE <tablename> DROP <columnname>; ALTER TABLE <tablename> MODIFY <column> <new_definition>; Modifies the structure of a table	DDL
4.	**Command:** **Syntax:** **Purpose:**	USE USE <databasename>; Opens the specified database for use.	DML
5.	**Command:** **Syntax:** **Purpose:**	SELECT DATABASE() SELECT DATABASE(); Shows the name of the current database	DML
6.	**Command:** **Syntax:** **Purpose:**	SHOW TABLES SHOW TABLES; Shows a list of tables present in the current database.	DML
7.	**Command:** **Syntax:** **Purpose:**	INSERT INSERT INTO <tablename> [<column1>, <column2>, ..., <columnn>] VALUES (<value1>, <value2>, ... <value n>); Inserts data into a table	DML

8.	**Command:** **Syntax:** **Purpose:**	SELECT SELECT <* / column name / expression> , [<column name/Expression list>] FROM <table name> [WHERE <condition>] [ORDER BY <column name / expression> [ASC/DESC]]; There are multiple ways to use SELECT. These will be explained with the help of examples in this lesson. Retrieves data from a table	DML
9.	**Command:** **Syntax:** **Purpose:**	DESCRIBE DESC[RIBE] <tablename>; Shows the structure of a table.	DML
10.	**Command:** **Syntax:** **Purpose:**	UPDATE UPDATE <tablename> SET <column name> = <value> [,<column name> = <value>, ...] [WHERE <condn>]; Updates/Modifies data in a table	DML
11.	**Command:** **Syntax:** **Purpose:**	DELETE DELETE FROM < tablename> [Where < condition>]; Deletes data from a table	DML

Following are the clauses which can be used with SELECT command:

a.	DISTINCT	Used to display distinct values from a column of a table.
b.	WHERE	Used to specify the condition based on which rows of a table are displayed.
c.	BETWEEN	Used to define the range of values within which the column values must fall to make a condition true. Range includes both the upper and the lower values.
d.	IN	Used to select values that match any value in a list of Specified values.
e.	LIKE	Used for pattern matching of string data using wildcard characters % and _ .
f.	IS NULL / NOT NULL	Used to select rows in which the specified column is NULL (or is NOT NULL)
g.	ORDER BY	Used to display the selected rows in ascending or in descending order of the specified column/expression.

Now, let us revise these commands with the help of Library database discussed above.

- After starting MySQL, our first job is to create a database in which all the tables will be stored. For this we give the statement:

```
CREATE DATABASE Library;
```

  This statement will create a database with the name Library using the command CREATE DATABASE.

- Now we have to open this database so that we can create tables and do other tasks in it. For this we give the following statement

```
USE Library;
```

After creating the database and opening it, we have to create tables. Before we create a table, we should very carefully plan columns that we want to have in the table and type of data to be stored in each column. The type of data decides what data types should be

specified for different columns of a table. There are many data types available in MySQL, but a few which are used most often are:

Class	Data Type
Text	CHAR(size)
	VARCHAR(size)
NUMERIC	DECIMAL(p,s)
	INT(size) or INTEGER(size)
date	DATE

After planning, we arrive at the following data types for columns of our tables in the Library database. Here the column sizes are based on our assumption. These may be different for different library databases.

BOOKS

Coulmn	Data Type	Reason
AccNo	INTEGER(5)	Accession number of a book is an integer in the range 1 to 99999.
Title	VARCHAR(30)	Title of a book may be maximum upto 30 characters long
Author	VARCHAR(30)	An author name may be maximum upto 30 characters long
Publisher	VARCHAR(10)	A publisher name may be maximum upto 10 characters long.
Edition	INTEGER(4)	Year of edition of the book
Price	DECIMAL(7, 2)	We assume that the price of a book will not exceed Rs. 99999.99

STUDENTS

Coulmn	Data Type	Reason
AdmNo	VARCHAR(7)	Here we assume that admission number consists of 4 digit year of admission followed by 3 digit admission serial number. Therefore, AdmNo 2009012 means that the students was admitted in the school in the year 2009 and he/she was the 12th student (in sequence) to take admission in that year.
LibCardNo	INTEGER(4)	Library card number is an integer in the range 1 to 9999.
AccNo	INTEGER(5)	Accession number of a book is an integer in the range 1 to 99999.
IssueDate	DATE	IssueDate is the date of issue of the book.
ReturnDate	DATE	ReturnDate is the date on which the book was returned.

STAFF

Coulmn	Data Type	Reason
EmpNo	INTEGER(3)	Employee number is an integer in the range 1 to 999.
AccNo	INTEGER(5)	Accession number of a book is an integer in the range 1 to 99999.
IssueDate	DATE	It is the date of issue of the book.
ReturnDate	DATE	It is the date on which the book was returned.

After deciding the columns to be kept in each table and their data types, we are ready to create the tables.

The statements to create these tables are as follows:

- To create the BOOKS table:

```
CREATE TABLE Books
    (AccNo INTEGER(5),
    Title VARCHAR(30),
    Author VARCHAR(30),
    Publisher VARCHAR(10),
    Edition INTEGER(4),
    Price DECIMAL(7,2));
```

- To create the STUDENTS table:

```
CREATE TABLE Students
    (AdmNo VARCHAR(7),
    LibCardNo INTEGER(4),
    AccNo INTEGER(5),
    IssueDate DATE,
    ReturnDate DATE);
```

- To create the STAFF table:

```
CREATE TABLE Staff
    (EmpNo INTEGER(3),
    AccNo INTEGER(5),
    IssueDate DATE,
    ReturnDate DATE);
```

- To verify that the tables are created, we give the statement:

```
SHOW TABLES;
```

- To further verify that the tables have been created as per the required specifications, we view the structure of each table by giving the statements:

```
DESC Books;

DESC Students;

DESC Staff;
```

- If the structure of a table is not as per the required specifications, we can modify it by using the ALTER TABLE command. Following are three examples of this:

  (i) Suppose we decide to categorise the books into categories 'A', 'B' and 'C' based on some criteria. Then a new column Category of type CHAR(1) can be added to the table by the statement:

```
ALTER TABLE Books ADD Category CHAR(1);
```

  (ii) Suppose we wish to change the size of column 'publisher' of Books table from 10 characters to 20 characters, then we can give the statement:

```
ALTER TABLE Books MODIFY Publisher VARCHAR(20);
```

  (iii) Suppose we decide to remove the ReturnDate column from the table STAFF. We can do so by the statement:

```
ALTER TABLE Staff DROP ReturnDate;
```

- Once the tables are created as per the required specifications, we can populate these tables with the given sample data as follows:

  → For Books table:

```
INSERT INTO Books VALUES
    (1001, 'Basic Economics', 'J.P.Mudgil',
    'APH', 2009, 125);
INSERT INTO Books VALUES
    (1002, 'Micro Economics', 'Kavi Prakash',
    'KPH', 2009, 175);
```

→ For Students table:

```
INSERT INTO Students VALUES
    ('2010001', 101, 1002, '2009-12-02', '2009-12-08');
INSERT INTO Students VALUES
    ('2009012', 102, 1050,'2010-02-16', NULL);
```

→ For Staff table:

```
INSERT INTO Staff VALUES
        (1, 1001,'2009-04-13', NULL);
INSERT INTO Staff (EmpNO) VALUE (2);
```

- Now we can check whether the data has been entered into the tables correctly or not by entering the statements:

```
SELECT * FROM Books;
SELECT * FROM Students;
SELECT * FROM Staff;
```

We can use various clauses with SELECT command to display the data. Examples are shown below:

- To check the names (without repetition) of various publishers whose books are present in the library, we enter the statement:

```
SELECT DISTINCT Publisher FROM Books;
```

- To check the books of publisher 'APH' present in the library, we enter the statement:

```
SELECT * from Books WHERE Publisher = 'APH';
```

- To display the books list for which the price is between ₹ 250 and ₹ 500, we enter the statement:

```
SELECT * FROM Books WHERE Price > 250 and Price < 500;
```

- To display the books list for which the price is from ₹ 250 to ₹ 500, we enter the statement:

```
SELECT * FROM Books WHERE Price >= 250 and Price <= 500;
```

OR

```
SELECT * FROM Books WHERE Price BETWEEN 250 and 500;
```

- To display the details of books from the publishers 'APH' 'JPH', or 'ABD', we enter the statement:

```
SELECT * from Books
    WHERE Publisher = 'APH'
        OR Publisher = 'JPH'
        OR Publisher = 'ABD';
```

OR

```
SELECT * from Books
    WHERE Publisher IN ('APH', 'JPH', 'ABD');
```

- To list the AccNo, Title, and Price of all the books whose Title contains the word 'History', we can enter the statement:

```
SELECT AccNo, Title, Price from Books
    WHERE Title LIKE '%History%';
```

- To display the AccNo of all the books which have been issued to students but not returned by them, we enter the statement:

```
SELECT AccNo FROM Students
    WHERE IssueDate IS NOT NULL and ReturnDate IS NULL;
```

- To Display a List of all the Books in the alphabetical ascending order of Titles, we enter the statement:

```
SELECT * FROM Books ORDER BY Title;
```

- To get the same list in descending order of Titles, we enter the statement:

```
SELECT * FROM Books ORDER BY Title DESC;
```

- Now suppose a student returns a Book with AccNo 1245 on 12 August 2010, then this information can be recorded in the Students table as follows:

```
UPDATE Students
    SET ReturnDate = '2010-08-12'
        WHERE AccNo = 1245;
```

- If we want to delete the records of all the books by the Publisher 'PPP' with edition year earlier than 1990, we can enter the statement:

```
DELETE FROM Books
    WHERE Publisher = 'PPP' AND Edition < 1990;
```

Suppose for some analysis we decide to keep track of how many times each book is issued. This can be done if there is an extra column 'IssueFreq' in the table Books to keep this count. Every time a book is issued, the corresponding value in this column is incremented by 1. To add this extra column in the table, we enter the statement:

```
ALTER TABLE Books ADD IssueFreq INTEGER(3);
```

Functions in MySQL

Let us now revise the following single-row SQL functions which are available in MySQL:

Numeric Functions:

Sno	Name & Syntax	Description
1.	POWER(*x*,*y*) OR POW(*x*,*y*)	Returns the value of *x* raised to the power of *y*.
2.	ROUND(*x*)	Rounds the argument *x* to the nearest INTEGER.
3.	ROUND(*x*,*d*)	Rounds the argument *x* to d decimal places.
4.	TRUNCATE(*x*,*d*)	Truncates the argument *x* to d decimal places.

String Functions:

SNo	Name & Syntax	Description
1.	LENGTH(str)	Returns the length of a column or a string in bytes.
2.	CONCAT(str1,str2,...)	Returns the string that results from concatenating the arguments. May have one or more arguments.
3.	INSTR(str,substr)	Returns the position of the first occurrence of substring <substr> in the string <str>.
4.	LOWER(str) or LCASE(str)	Returns the argument <str> in lowercase. i.e., It changes all the characters of the passed string to lowercase.
5.	UPPER(str) or UCASE(str)	Returns the argument <str> in uppercase. i.e., It changes all the characters of the passed string to uppercase.
6.	LEFT(str,n)	Returns the first <n> characters from the string <str>
7.	RIGHT(str,n)	Returns the last <n> characters from the string <str>
8.	LTRIM(str)	Removes leading spaces, i.e., removes spaces from the left side of the string <str>.
9.	RTRIM(str)	Removes trailing spaces, i.e., removes spaces from the right side of the string <str>.
10.	TRIM(str)	Removes both leading and trailing spaces from the string <str>.
11.	SUBSTRING(str,m,n) **OR** SUBSTR(str, m, n) **OR** MID(str,m,n)	Returns <n> characters starting from the m^{th} character of the string <str>. If the third argument <n> is missing, then starting from m^{th} position, the rest of the string is returned. If <m> is negative, the beginning of the substring is the m^{th} character from the end of the string.
12.	ASCII(str)	Returns the ASCII value of the first character of the string <str>. Returns 0 if <str> is the empty string. Returns NULL if <str> is NULL.

Date and Time Functions:

SNo	Name & Syntax	Description
1	CURDATE()	Returns the current date in YYYY-MM-DD format or YYYYMMDD format, depending on whether the function is used in a string or numeric context.
2	NOW()	Returns the current date and time in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS.uuuuuu format, depending on whether the function is used in a string or numeric context.
3	SYSDATE()	Returns the current date and time in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS.uuuuuu format, depending on whether the function is used in a string or numeric context.
4	DATE(expr)	Extracts the date part of a date or datetime expression <expr>.
5	MONTH(date)	Returns the numeric month from the specified date, in the range 0 to 12. It returns 0 for dates such as '0000-00-00' or '2010-00-00' that have a zero month part.
6	YEAR(date)	Returns the year for specified date in the range 0 to 9999. It returns 0 for the "zero" date. Returns values like 1998, 2010,1996 etc.
7	DAYNAME(date)	It returns the name of the weekday for the specified date.
8	DAYOFMONTH(date)	Returns the day of the month in the range 0 to 31.
9	DAYOFWEEK(date)	Returns the day of week in number as 1 for Sunday, 2 for Monday and so on.
10	DAYOFYEAR(date)	Return the day of the year for the given date in numeric format in the range 1 to 366.

A few examples of these functions are given below:

1. Suppose a book with AccNo 1002 is lost by a student. The school charges a fine for the lost book. The fine is 10% of the book price. We wish to write a statement to calculate the fine rounded up to 2 DECIMAL places. The statement is

```
SELECT AccNo, ROUND(Price*10/100, 2) AS Fine
    FROM Books WHERE AccNo = 1002;
```

2. Suppose we want to create another table and one column in that table will be 'Author'. In order to optimally utilize the storage space, we wish to find out what is the maximum length of 'Author' in the table 'Books'. This value will guide us in deciding the field width of 'Author' in the new table. So we enter the statement:

```
SELECT Author, LENGTH(Author)
    FROM Books ORDER BY LENGTH(Author);
```

3. In order to produce a report we wish to display the book titles in upper case letters of the alphabet. So we enter the statement:

```
SELECT AccNo, UCASE(Title) FROM Books;
```

4. Suppose a book with accession number 1050 is returned and we wish to enter the current date as the ReturnDate in the table Students. We can give the statement:

```
UPDATE Students
    SET ReturnDate = CURDATE()
    WHERE AccNo = 1050;
```

5. Suppose we wish to find out which staff members got the books issued in the previous calender year, then we can give the statement:

```
SELECT * FROM Staff WHERE YEAR(IssueDate) =
    YEAR(CURDATE())-1;
```

What Next?

So far you have learnt how to work with single table at a time. None of the operations discussed in class XI or in this lesson required us to access data simultaneously from multiple tables. But in real life database implementations we have to access data from multiple tables simultaneously, as will be clear from the following examples.

Try to write queries for the following tasks in Library database:

(i) For all the books which have not been returned by the students, display the AccNO and Title of the book.

(ii) Display the details of all the books by the publisher 'APH' which are issued to the Staff.

(iii) Display the Price of each book which has been issued to the students but has not been returned.

If you observe carefully, you will find that these queries need to access data from two tables simultaneously. You have not yet studied how to do this.

In class XII, you will study SQL commands using which you will be able to do all such jobs and a lot more. For example:

(iv) Display the highest and the lowest values from Price column of the table Books.

(v) Display the Total number of books in the library whose edition is before 2001.

(vi) To cancel the operations done while working in MySQL.

(vii) To make the operations done permanent so that even in the case of system failure the work done stays.

(viii) To make sure that duplicate values are not accepted in the primary key column of a table.

(ix) To make sure that NULL is not accepted for any column in which we do not want to allow NULLs.

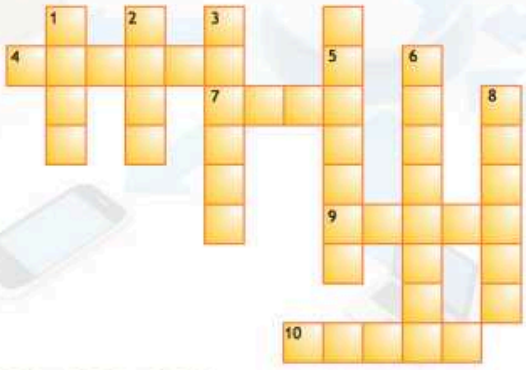
Summary:

- In an RDBMS data is stored in tables.
- A table refers to a two dimensional representation of data arranged in columns and rows.
- SQL (Structured Query Language) is used to manage databases in an RDBMS.
- SQL commands are divided into different categories: DDL, DML, and DCL.
- CREATE command is used to create databases and tables.

- The DESCRIBE command is used to view the structure of a table.
- ALTER TABLE statement is used to change the structure of a table ie. to add, remove or change its column(s).
- The SELECT command is used to fetch data from tables.
- The keyword DISTINCT is used to eliminate display of duplicate data.
- The WHERE clause is used to select specific rows.
- The BETWEEN operator defines the range of values that a column values must fall into to make the condition true.
- The IN operator selects values that match any value in the given list of values.
- % and _ are two wild card characters. The percent (%) symbol is used to represent any sequence of zero or more characters. The underscore (_) symbol is used to represent a single character.
- NULL means a value that is unavailable, unassigned, unknown or inapplicable.
- The results of the SELECT command can be displayed in the ascending or descending values of a column or columns using ORDER BY clause.
- UPDATE command is used to modify existing data in a table.
- DELETE command is used to delete rows from a table.

EXERCISES

Complete the following cross-word puzzle using SQL commands and clauses:



Clues:

Across	Down
4. Remains same until you change the no. of columns in the table.	1. Show the table structure
7. Show me the strings following a specific pattern.	2. Destroy the table
9. It contains data	3. Display the table data
10. To add another column to the table	5. I don't want data in this table
	6. It contains tables
	8. Populate the table

MULTIPLE CHOICE QUESTIONS

1. MySQL is a /an
   a. Relational Database b. Database
   c. RDBMS d. Table
2. USE <databasename> command
   a. is not used in MySQL b. is given before quitting MySQL
   c. opens a table d. opens a database
3. Number of columns in a table is called
   a. Power b. Degree
   c. Cardinality d. Design
4. A database
   a. Contains tables b. Is a part of a table
   c. Is same as a table d. Removes data from a table

5. Number of rows in a table is called
   a. Power
   b. Degree
   c. Cardinality
   d. Design
6. ORDER BY clause is used to sort data
   a. in ascending order
   b. in desceding order
   c. both (a) and (b)
   d. None of the above
7. A table can have
   a. Only one candidate key
   b. Only one primary key
   c. Only one alternate key
   d. No alternate key
8. Wild card characters are used
   a. In LIKE clause
   b. In ORDER BY clause
   c. In BETWEEN clause
   d. In IN clause
9. DDL is
   a. a part of SQL
   b. a part of DML
   c. a part of DCL
   d. None of the above
10. LIKE clause is used
    a. For pattern matching
    b. For table matching
    c. For inserting similar data in a table
    d. For deleting data from a table

LAB EXERCISES

Consider a database LOANS with the following table:

Table: Loan_Accounts

AccNo	Cust_Name	Loan_Amount	Instalments	Int_Rate	Start_Date	Interest
1	R.K. Gupta	300000	36	12.00	19-07-2009	
2	S.P. Sharma	500000	48	10.00	22-03-2008	
3	K.P. Jain	300000	36	NULL	08-03-2007	
4	M.P. Yadav	800000	60	10.00	06-12-2008	
5	S.P. Sinha	200000	36	12.50	03-01-2010	
6	P. Sharma	700000	60	12.50	05-06-2008	
7	K.S. Dhall	500000	48	NULL	05-03-2008	

Write SQL commands for the tasks 1 to 35 and write the output for the SQL commands 36 to 40:

Create Database and use it

1. Create the database LOANS.
2. Use the database LOANS.

Create Table / Insert Into

3. Create the table Loan_Accounts and insert tuples in it.

Simple Select

4. Display the details of all the loans.
5. Display the AccNo, Cust_Name, and Loan_Amount of all the loans.

Conditional Select using Where Clause

6. Display the details of all the loans with less than 40 instalments.
7. Display the AccNo and Loan_Amount of all the loans started before 01-04-2009.
8. Display the Int_Rate of all the loans started after 01-04-2009.

Using NULL

9. Display the details of all the loans whose rate of interest is NULL.
10. Display the details of all the loans whose rate of interest is not NULL.

Using DISTINCT Clause

11. Display the amounts of various loans from the table Loan_Accounts. A loan amount should appear only once.
12. Display the number of instalments of various loans from the table Loan_Accounts. An instalment should appear only once..

Using Logical Operators (NOT, AND, OR)

13. Display the details of all the loans started after 31-12-2008 for which the number of instalments are more than 36.
14. Display the Cust_Name and Loan_Amount for all the loans which do not have number of instalments 36.
15. Display the Cust_Name and Loan_Amount for all the loans for which the loan amount is less than 500000 or int_rate is more than 12.
16. Display the details of all the loans which started in the year 2009.
17. Display the details of all the loans whose Loan_Amount is in the range 400000 to 500000.
18. Display the details of all the loans whose rate of interest is in the range 11% to 12%.

Using IN Operator

19. Display the Cust_Name and Loan_Amount for all the loans for which the number of instalments are 24, 36, or 48. (Using IN operator)

Using BETWEEN Operator

20. Display the details of all the loans whose Loan_Amount is in the range 400000 to 500000. (Using BETWEEN operator)
21. Display the details of all the loans whose rate of interest is in the range 11% to 12%. (Using BETWEEN operator)

Using LIKE Operator

22. Display the AccNo, Cust_Name, and Loan_Amount for all the loans for which the Cust_Name ends with 'Sharma'.
23. Display the AccNo, Cust_Name, and Loan_Amount for all the loans for which the Cust_Name ends with 'a'.
24. Display the AccNo, Cust_Name, and Loan_Amount for all the loans for which the Cust_Name contains 'a'
25. Display the AccNo, Cust_Name, and Loan_Amount for all the loans for which the Cust_Name does not contain 'P'.
26. Display the AccNo, Cust_Name, and Loan_Amount for all the loans for which the Cust_Name contains 'a' as the second last character.

Using ORDER BY clause

27. Display the details of all the loans in the ascending order of their Loan_Amount.
28. Display the details of all the loans in the descending order of their Start_Date.
29. Display the details of all the loans in the ascending order of their Loan_Amount and within Loan_Amount in the descending order of their Start_Date.

Using UPDATE, DELETE, ALTER TABLE

30. Put the interest rate 11.50% for all the loans for which interest rate is NULL.
31. Increase the interest rate by 0.5% for all the loans for which the loan amount is more than 400000.
32. For each loan replace Interest with (Loan_Amount*Int_Rate*Instalments) 12*100.
33. Delete the records of all the loans whose start date is before 2007.
34. Delete the records of all the loans of 'K.P. Jain'
35. Add another column Category of type CHAR(1) in the Loan table.

Find the Output of the following queries

36. SELECT cust_name, LENGTH(Cust_Name), LCASE(Cust_Name), UCASE(Cust_Name) FROM Loan_Accounts WHERE Int_Rate < 11.00;
37. SELECT LEFT(Cust_Name, 3), Right(Cust_Name, 3), SUBSTR(Cust_Name, 1, 3) FROM Loan_Accounts WHERE Int_Rate > 10.00;
38. SELECT RIGHT(Cust_Name, 3), SUBSTR(Cust_Name, 5) FROM Loan_Accounts;
39. SELECT DAYNAME(Start_Date) FROM Loan_Accounts;
40. SELECT ROUND(Int_Rate*110/100, 2) FROM Loan_Account WHERE Int_Rate > 10;

Write the output produced by the following SQL commands:

41. SELECT POW(4,3), POW(3,4);
42. SELECT ROUND(543.5694,2), ROUND(543.5694), ROUND(543.5694,-1);
43. SELECT TRUNCATE(543.5694,2), TRUNCATE(543.5694,-1);
44. SELECT LENGTH("Prof. M. L. Sharma");
45. SELECT CONCAT("SHEIKH", " HAROON") "FULL NAME";
46. SELECT YEAR(CURDATE()), MONTH(CURDATE()), DAY(CURDATE());
47. SELECT DAYOFYEAR(CURDATE()), DAYOFMONTH(CURDATE()), DAYNAME(CURDATE());
48. SELECT LEFT("Unicode",3), RIGHT("Unicode",4);
49. SELECT INSTR("UNICODE","CO"), INSTR("UNICODE","CD");
50. SELECT MID("Informatics",3,4), SUBSTR("Practices",3);

MORE ON DATABASES AND SQL

Learning Objectives

After studying this lesson the students will be able to:

- Define the terms:
  - (i) Group (Aggregate) functions, Constraints
  - (ii) Cartesian Product, Join, Referential Integrity, Foreign Key.
- Write queries using aggregate functions and GROUP BY clause.
- Access data from multiple tables
- Create tables with PRIMARY KEY and NOT NULL constraints
- Add a constraint to a table, remove a constraint from a table, modify a column of a table using ALTER TABLE command.
- Delete a table using DROP TABLE.

In the previous class, you have learnt some database concepts and SQL commands. You have also learnt how to create databases and tables within databases and how to access data from a table using various clauses of SELECT command. In this chapter you shall learn some more clauses and functions in SQL to access data from a table and how to access data from multiple tables of a database.

Puzzle [1]

It was Iftar party in Lucknow that Mr. David met Mr. Naqvi. They became friends and exchanged their phone numbers. After a few days, Mr. David rang up and invited Mr. Naqvi for New Year party at his house and gave him his house number as follows:

"I live in a long street. Numbered on the side of my house are the houses one, two, three and so on. All the numbers on one side of my house add up to exactly the same as all the

numbers on the other side of my house. I know there are more than thirty houses on that side of the street, but not so many as 50."

With this information, Mr. Naqvi was able to find Mr. David's house number. Can you also find?

Such situations are faced by the developers of RDBMS software where they have to think of retrieval of data from multiple tables satisfying some specified conditions.

Let us now move ahead with SQL and more database concepts.

Ms. Shabana Akhtar is in-charge of computer department in a Shoe factory. She has created a database 'Shoes' with the following tables:

SHOES

(To store the information about various types of shoes made in the factory)

Field	Type	Null	Key	Default	Extra
code	char(4)	NO	PRI	NULL	
name	varchar(20)	YES		NULL	
type	varchar(10)	YES		NULL	
size	int(2)	YES		NULL	
cost	decimal(6,2)	YES		NULL	
margin	decimal(4,2)	YES		NULL	
Qty	int(4)	YES		NULL	

CUSTOMERS

(To store the data of customers)

Field	Type	Null	Key	Default	Extra
cust_Code	char(4)	NO	PRI	NULL	
name	varchar(30)	YES		NULL	
address	varchar(50)	YES		NULL	
phone	varchar(30)	YES		NULL	
category	char(1)	YES		NULL	

ORDERS

(To store the data of orders placed by customers)

Field	Type	Null	Key	Default	Extra
order_no	int(5)	NO	PRI	NULL	
cust_code	char(4)	YES		NULL	
Shoe_Code	char(4)	YES		NULL	
order_qty	int(4)	YES		NULL	
order_date	date	YES		NULL	
target_date	date	YES		NULL	

Sample data stored in these tables is given below:

SHOES

Code	Name	type	size	cost	margin	Qty
1001	School Canvas	School	6	132.50	2.00	1200
1002	School Canvas	School	7	135.50	2.00	800
1003	School Canvas	School	8	140.75	2.00	600
1011	School Leather	School	6	232.50	2.00	2200
1012	School Leather	School	7	270.00	2.00	1280
1013	School Leather	School	8	320.75	NULL	1100
1101	Galaxy	Office	7	640.00	3.00	200
1102	Galaxy	Office	8	712.00	3.00	500
1103	Galaxy	Office	9	720.00	3.00	400
1201	Tracker	Sports	6	700.00	NULL	280
1202	Tracker	Sports	7	745.25	3.50	NULL
1203	Tracker	Sports	8	800.50	3.50	600
1204	Tracker	Sports	9	843.00	NULL	860

CUSTOMERS

Cust_Code	name	address	Phone	Category
C001	Novelty Shoes	RajaNagar, Bhopal	4543556, 97878989	A
C002	Aaram Footwear	31, Mangal Bazar, Agra	NULL	B
C003	Foot Comfort	NewMarket, Saharanpur	51917142, 76877888	B
C004	Pooja Shoes	Janak Puri, New Delhi	61345432, 98178989	A
C005	Dev Shoes	Mohan Nagar, Ghaziabad	NULL	C

ORDERS

order_no	cust_code	Shoe_Code	order_qty	order_date	target_date
1	C001	1001	200	2008-12-10	2008-12-15
2	C001	1002	200	2008-12-10	2008-12-15
3	C003	1011	150	2009-01-08	2009-01-13
4	C002	1012	250	2009-01-08	2009-01-13
5	C001	1011	400	2009-01-10	2009-01-15
6	C002	1101	300	2009-01-10	2009-01-15
7	C004	1201	200	2009-01-10	2009-01-15
8	C005	1102	350	2009-01-10	2009-01-15
9	C001	1103	225	2009-01-13	2009-01-18
10	C002	1203	200	2009-01-14	2009-01-19

Let us now see how this database helps Ms. Akhtar in generating various reports quickly.

Aggregate Functions

In class XI we studied about single row functions available in SQL. A single row function works on a single value. SQL also provides us multiple row functions. A multiple row function works on multiple values. These functions are called aggregate functions or group functions. These functions are:

S.No.	Function	Purpose
1	MAX()	Returns the MAXIMUM of the values under the specified column/expression.
2	MIN()	Returns the MINIMUM of the values under the specified column/expression.

3	AVG()	Returns the AVERAGE of the values under the specified column/expression.
4	SUM()	Returns the SUM of the values under the specified column/expression.
5	COUNT()	Returns the COUNT of the number of values under the specified column/expression.

MAX() :

MAX() function is used to find the highest value of any column or any expression based on a column. MAX() takes one argument which can be any column name or a valid expression involving a column name. e.g.,

Purpose	Statement	Output

To find the highest cost of any type of shoe in the factory.	`SELECT MAX(cost) FROM shoes;`	`+-----------+`
		`| MAX(cost) |`
		`+-----------+`
		`| 843.00 |`
		`+-----------+`

To find the highest cost of any shoe of type 'School'.	`SELECT MAX(cost) FROM shoes WHERE type = 'School';`	`+-----------+`
		`| MAX(cost) |`
		`+-----------+`
		`| 320.75 |`
		`+-----------+`

To find the highest selling price of any type of shoe.	`SELECT MAX(cost+cost*margin/100) FROM shoes;`	`+--------------------------+`
		`| MAX(cost+cost*margin/100) |`
		`+--------------------------+`
		`| 828.517500000 |`
		`+--------------------------+`

To find the highest selling price of any type of shoe rounded to 2 decimal places.	`SELECT ROUND(MAX(cost+cost*margin/100),2) AS "Max. SP" FROM shoes;`	`+---------+`
		`| Max. SP |`
		`+---------+`
		`| 733.36 |`
		`+---------+`

To find the highest selling price of any type of shoe rounded to 2 decimal places.	`SELECT ROUND(MAX(cost+cost*margin/100),2) AS "Max. SP" FROM shoes;`	`+---------+`
		`| Max. SP |`
		`+---------+`
		`| 733.36 |`
		`+---------+`

MIN() :

MIN() function is used to find the lowest value of any column or an expression based on a column. MIN() takes one argument which can be any column name or a valid expression involving a column name. e.g.,

Purpose	Statement	Output

To find the lowest cost of any type of shoe in the factory.	`SELECT MIN(cost) FROM shoes;`	`+-----------+`
		`| MIN(cost) |`
		`+-----------+`
		`| 843.00 |`
		`+-----------+`

To find the lowest cost of any shoe of type 'School'.	`SELECT MIN(cost) FROM shoes WHERE type = 'School';`	`+-----------+`
		`| MIN(cost) |`
		`+-----------+`
		`| 320.75 |`
		`+-----------+`

To find the lowest selling price of any type of shoe rounded to 2 decimal places.	`SELECT ROUND(MIN(cost+cost*margin/100),2) AS "Min. SP" FROM shoes;`	`+---------+`
		`| Min. SP |`
		`+---------+`
		`| 135.15 |`
		`+---------+`

AVG() :

AVG() function is used to find the average value of any column or an expression based on a column. AVG() takes one argument which can be any column name or a valid expression involving a column name. Here we have a limitation: the argument of AVG() function can be of numeric (int/decimal) type only. Averages of String and Date type data are not defined. E.g.,

Purpose	Statement	Output

To find the average margin from shoes table.	`SELECT AVG(margin) FROM shoes;`	`+-------------+`
		`| AVG(margin) |`
		`+-------------+`
		`| 2.600000 |`
		`+-------------+`

To find the average cost from the shoes table.	`SELECT AVG(cost) FROM shoes;`	`+------------+`
		`| AVG(cost) |`
		`+------------+`
		`| 491.750000 |`
		`+------------+`

To find the average quantity in stock for the shoes of type Sports.	`SELECT AVG(qty) FROM shoes WHERE type = 'Sports';`	`+----------+`
		`| AVG(qty) |`
		`+----------+`
		`| 580.0000 |`
		`+----------+`

SUM() :

SUM() function is used to find the total value of any column or an expression based on a column. SUM() also takes one argument which can be any column name or a valid expression involving a column name. Like AVG(), the argument of SUM() function can be of numeric (int/decimal) type only. Sums of String and Date type data are not defined. e.g.,

Purpose	Statement	Output

To find the total quantity present in the stock	`SELECT SUM(Qty) FROM Shoes;`	`+----------+`
		`| SUM(Qty) |`
		`+----------+`
		`| 10020 |`
		`+----------+`

To find the total order quantity	`SELECT SUM(order_qty)`
	`FROM orders;`
		`+----------------+`
		`| SUM(order_qty) |`
		`+----------------+`
		`| 2475 |`
		`+----------------+`

To find the the total value (Quanitity x Cost) of Shoes of type 'Office' present in the inventory	`SELECT SUM(cost*qty)`
	`FROM shoes`
	`WHERE type =`
	`'Office';`
		`+---------------+`
		`| SUM(cost*qty) |`
		`+---------------+`
		`| 772000.00 |`
		`+---------------+`

COUNT() :

COUNT() function is used to count the number of values in a column. COUNT() takes one argument which can be any column name, an expression based on a column, or an asterisk (*). When the argument is a column name or an expression based on a column, COUNT() returns the number of non-NULL values in that column. If the argument is a *, then COUNT() counts the total number of rows satisfying the condition, if any, in the table. e.g.,

Purpose	Statement	Output

To count the total number of records in the table Shoes.	`SELECT COUNT(*) FROM shoes;`	`+----------+`
		`| COUNT(*) |`
		`+----------+`
		`| 13 |`
		`+----------+`

To count the different types of shoes that the factory produces	`SELECT COUNT(distinct type)`
	`FROM shoes;`
		`+----------------------+`
		`| COUNT(distinct type) |`
		`+----------------------+`
		`| 3 |`
		`+----------------------+`

To count the records for which the margin is greater than 2.00	`SELECT COUNT(margin)`
	`FROM shoes`
	`WHERE margin > 2;`
		`+---------------+`
		`| COUNT(margin) |`
		`+---------------+`
		`| 5 |`
		`+---------------+`

To count the number of customers in 'A' category	`SELECT COUNT(*)`
	`FROM customers`
	`WHERE category ='A';`
		`+----------+`
		`| COUNT(*) |`
		`+----------+`
		`| 2 |`
		`+----------+`

To count the number of orders of quantity more than 300	`SELECT COUNT(*)`
	`FROM orders`
	`WHERE order_qty > 300;`
		`+----------+`
		`| COUNT(*) |`
		`+----------+`
		`| 2 |`
		`+----------+`

Aggregate functions and NULL values:

None of the aggregate functions takes NULL into consideration. NULL is simply ignored by all the aggregate functions. For example, the statement:

```
SELECT COUNT(*) FROM shoes;
```

produces the following output:

```
+----------+
| COUNT(*) |
+----------+
|       13 |
+----------+
```

Indicating that there are 13 records in the Shoes table. Whereas the query:

```
SELECT COUNT(margin) FROM shoes;
```

produces the output:

```
+---------------+
| COUNT(margin) |
+---------------+
|            10 |
+---------------+
```

This output indicates that there are 10 values in the margin column of Shoes table. This means there are 3 (13-10) NULLs in the margin column.

This feature of aggregate functions ensures that NULLs don't play any role in actual calculations. For example, the following statement:

```
SELECT AVG(margin) FROM shoes;
```

produces the output:

```
+--------------+
| AVG(margin)  |
+--------------+
|    2.600000  |
+--------------+
```

The average margin has been calculated by adding all the 10 non NULL values from the margin column and dividing the sum by 10 and not by 13.

Know more!

There are some more aggregate functions available in MySQL. Try to find out what are those. Also try to use them.

GROUP BY

In practical applications many times there arises a need to get reports based on some groups of data. These groups are based on some column values. For example,

- *The management of the shoe factory may want to know what is the total quantity of shoes of various types. i.e., what is the total quantity of shoes of type School, Office, and Sports each.*
- *The management may also want to know what is the maximum, minimum, and average margin of each type of shoes.*
- *It may also be required to find the total number of customers in each category.*

There are many such requirements.

SQL provides GROUP BY clause to handle all such requirements.

For the above three situations, the statements with GROUP BY clause are given below:

In the first situation we want MySQL to divide all the records of shoes table into different groups based on their type (GROUP BY type) and for each group it should display the type and the corresponding total quantity (SELECT type, SUM(qty)). So the complete statement to do this is:

```
SELECT type, SUM(qty) FROM shoes
    GROUP BY type;
```

G1

and the corresponding output is:

```
+---------+-----------+
| type    | SUM(qty)  |
+---------+-----------+
| Office  |      1100 |
| School  |      7180 |
| Sports  |      1740 |
+---------+-----------+
```

Similarly, for the second situation the statement is:

```
SELECT type, MIN(margin), MAX(margin), AVG(margin)
    FROM shoes GROUP BY type;
```
——— G2

and the corresponding output is:

```
+--------+-------------+-------------+-------------+
| type   | MIN(margin) | MAX(margin) | AVG(margin) |
+--------+-------------+-------------+-------------+
| Office |        3.00 |        3.00 |    3.000000 |
| School |        2.00 |        2.00 |    2.000000 |
| Sports |        3.50 |        3.50 |    3.500000 |
+--------+-------------+-------------+-------------+
```

In the third situation we want MySQL to divide all the records of Customers table into different groups based on the their Category (GROUP BY Category) and for each group it should display the Category and the corresponding number of records (SELECT Category, COUNT(*)). So the complete statement to do this is:

```
SELECT category, COUNT(*) FROM customers GROUP BY category;
```
— G3

```
+----------+----------+
| category | COUNT(*) |
+----------+----------+
| A        |        2 |
| B        |        2 |
| C        |        1 |
+----------+----------+
```

Let us have some more examples.

Consider the following statement:

```
SELECT cust_code, SUM(order_qty)
    FROM orders GROUP BY cust_code;
```

This statement produces the following output. Try to explain this this output.

```
+-----------+----------------+
| cust_code | SUM(order_qty) |
+-----------+----------------+
| C001      |           1025 |
| C002      |            750 |
| C003      |            150 |
| C004      |            200 |
| C005      |            350 |
+-----------+----------------+
```

Do the same for the following statement also:

```
SELECT shoe_code, SUM(order_qty)
    FROM orders GROUP BY shoe_code;
```

```
+-----------+----------------+
| shoe_code | SUM(order_qty) |
+-----------+----------------+
| 1001      |            200 |
| 1002      |            200 |
| 1011      |            550 |
| 1012      |            250 |
| 1101      |            300 |
| 1102      |            350 |
| 1103      |            225 |
| 1201      |            200 |
| 1203      |            200 |
+-----------+----------------+
```

If you carefully observe these examples, you will find that GROUP BY is always used in conjunction with some aggregate function(s). A SELECT command with GROUP BY clause has a column name and one or more aggregate functions which are applied on that column and grouping is also done on this column only.

HAVING :

Sometimes we do not want to see the whole output produced by a statement with GROUP BY clause. We want to see the output only for those groups which satisfy some condition. It means we want to put some condition on individual groups (and not on individual records). A condition on groups is applied by HAVING clause. As an example reconsider the

statement G1 discussed above. The statement produced three records in the output - one for each group. Suppose, we are interested in viewing only those groups' output for which the total quantity is more than 1500 (SUM(Qty) > 1500). As this condition is applicable to groups and not to individual rows, we use HAVING clause as shown below:

```
SELECT type, SUM(qty) FROM shoes
   GROUP BY type HAVING SUM(qty) > 1500;
```

```
+--------+----------+
| type   | SUM(qty) |
+--------+----------+
| School |     7180 |
| Sports |     1740 |
+--------+----------+
```

Now suppose for G2 we want the report only for those types for which the average margin is more than 2. For this, following is the statement and the corresponding output:

```
SELECT type, SUM(qty) FROM shoes
   GROUP BY type HAVING AVG(margin) >2;
```

```
+--------+----------+
| type   | SUM(qty) |
+--------+----------+
| Office |     1100 |
| Sports |     1740 |
+--------+----------+
```

In these statements if we try to put the condition using WHERE instead of HAVING, we shall get an error. Another way of remembering this is that whenever a condition involves an aggregate function, then we use HAVING clause in conjunction with GROUP BY clause.

Situations may also arise when we want to put the conditions on individual records as well as on groups. In such situations we use both WHERE (for individual records) and HAVING (for groups) clauses. This can be explained with the help of the following examples:

- The management of the shoe factory may want to know what is the total quantity of shoes, of sizes other than 6, of various types. i.e., what is the total quantity of shoes (of sizes other than 6) of type School, Office, and Sports each.

Moreover, the report is required only for those groups for which the total quantity is more than 1500.

- The management may also want to know what is the maximum, minimum, and average margin of each type of shoes. But in this reports shoes of sizes 6 and 7 only should be included. Report is required only for those groups for which the minimum margin is more than 2.

The statements and their outputs corresponding to above requirements are given below:

```
SELECT type, SUM(qty) FROM shoes
    WHERE size <> 6◄──────────Checks individual row
    GROUP BY type HAVING sum (qty) > 1500;◄── Checks individual group
```

```
+--------+----------+
| type   | SUM(qty) |
+--------+----------+
| School |     3780 |
+--------+----------+
```

```
SELECT type, MIN(margin), MAX(margin), AVG(margin) FROM shoes
    WHERE size in (6,7)
    GROUP BY type having MIN(margin) > 2;
```

```
+--------+-------------+-------------+-------------+
| type   | MIN(margin) | MAX(margin) | AVG(margin) |
+--------+-------------+-------------+-------------+
| Office |        3.00 |        3.00 |    3.000000 |
| Sports |        3.50 |        3.50 |    3.500000 |
+--------+-------------+-------------+-------------+
```

Displaying Data from Multiple Tables

In each situation that we have faced so far, the data was extracted from a single table. There was no need to refer to more than one tables in the same statement. But many times, in real applications of databases, it is required to produce reports which need data from more than one tables. To understand this consider the following situations:

- *The management of the shoe factory wants a report of orders which lists three columns: Order_No, corresponding customer name, and phone number. - (MT-1)*

  *In this case order number will be taken from **Orders** table and corresponding customer name from **Customers** table.*

- *The management wants a four-column report containing order_no, order_qty, name of the corresponding shoe and its cost. - (MT-2)*

  *In this case order number and order quantity will be taken from **Orders** table and corresponding shoe name and cost from **Shoes** table.*

- *The management wants the names of customers who have placed any order of quantity more than 300. - (MT-3)*

  *In this case Order quantity will be checked in **Orders** table and for each record with quantity more than 300, corresponding **Customer** name will be taken from Customers table.*

- *The management wants a report in which with each Order_No management needs name of the corresponding customer and also the total cost (Order quantity x Cost of the shoe) of the order are shown. - (MT-4)*

  *In this case order number will be taken from **Orders** table and corresponding customer name from **Customers** table. For the cost of each order the quantity will be taken from Orders table and the Cost from **Shoes** table.*

In all these cases, the data is to be retrieved from multiple tables. SQL allows us to write statements which retrieve data from multiple tables.

To understand how this is done, consider the following tables of a database.

Product

Code	Name
P001	Toothpaste
P002	Shampoo
P003	Conditioner

Supplier

Sup_Code	Name	Address
S001	DC & Company	Uttam Nagar
S002	SURY Traders	Model Town

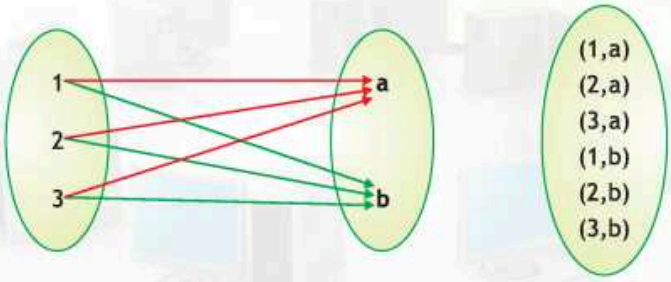
Order_table

Order_No	P_Code	Sup_Code
1	P001	S002
2	P002	S002

These tables are taken just to explain the current concept.

Cartesian Product or Cross Join of tables :

Cartesian product (also called Cross Join) of two tables is a table obtained by pairing up each row of one table with each row of the other table. This way if two tables contain 3 rows and 2 rows respectively, then their Cartesian product will contain 6 (=3x2) rows. This can be illustrated as follows:



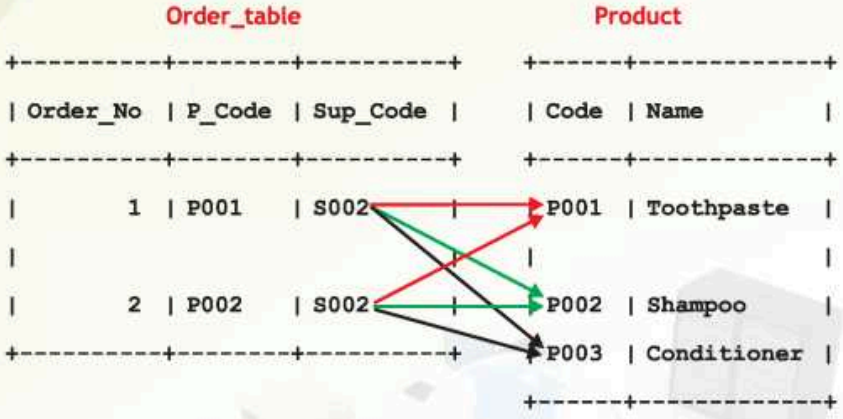
Cartesian product of two tables

Notice that the arrows indicate the 'ordered pairing'.

The number of columns in the Cartesian product is the sum of the number of columns in both the tables. In SQL, Cartesian product of two rows is obtained by giving the names of both tables in FROM clause. An example of Cartesian product is shown below:

```
SELECT * FROM order_table, product;
```

To give the output of this query, MySQL will pair the rows of the mentioned tables as follows:



And the following output will be produced:

```
+----------+--------+----------+------+-------------+
| Order_No | P_Code | Sup_Code | Code | Name        |
+----------+--------+----------+------+-------------+
|        1 | P001   | S002     | P001 | Toothpaste  |
|        2 | P002   | S002     | P001 | Toothpaste  |
|        1 | P001   | S002     | P002 | Shampoo     |
|        2 | P002   | S002     | P002 | Shampoo     |
|        1 | P001   | S002     | P003 | Conditioner |
|        2 | P002   | S002     | P003 | Conditioner |
+----------+--------+----------+------+-------------+
```

-(CP-1)

Here we observe that the Cartesian product contains all the columns from both tables. Each row of the first table (Order_table) is paired with each row of the second table (Product).

If we change the sequence of table names in the FROM clause, the result will remain the same but the sequence of rows and columns will change. This can be observed in the following statement and the corresponding output.

```
SELECT * FROM product, order_table;
```

-(CP-2)

Code	Name	Order_No	P_Code	Sup_Code
P001	Toothpaste	1	P001	S002
P001	Toothpaste	2	P002	S002
P002	Shampoo	1	P001	S002
P002	Shampoo	2	P002	S002
P003	Conditioner	1	P001	S002
P003	Conditioner	2	P002	S002

We can have Cartesian product of more than two tables also. Following is the Cartesian Product of three tables:

```
SELECT * FROM order_table, supplier, product;
```

-(CP-3)

Order_No	P_Code	Sup_Code	Sup_Code	Name	Address	Code	Name
1	P001	S002	S001	DC & Company	Uttam Nagar	P001	Toothpaste
2	P002	S002	S001	DC & Company	Uttam Nagar	P001	Toothpaste
1	P001	S002	S002	SURY Traders	Model Town	P001	Toothpaste
2	P002	S002	S002	SURY Traders	Model Town	P001	Toothpaste
1	P001	S002	S001	DC & Company	Uttam Nagar	P002	Shampoo
2	P002	S002	S001	DC & Company	Uttam Nagar	P002	Shampoo
1	P001	S002	S002	SURY Traders	Model Town	P002	Shampoo
2	P002	S002	S002	SURY Traders	Model Town	P002	Shampoo
1	P001	S002	S001	DC & Company	Uttam Nagar	P003	Conditioner
2	P002	S002	S001	DC & Company	Uttam Nagar	P003	Conditioner
1	P001	S002	S002	SURY Traders	Model Town	P003	Conditioner
2	P002	S002	S002	SURY Traders	Model Town	P003	Conditioner

The complete Cartesian product of two or more tables is, generally, not used directly. But, some times it is required. Suppose the company with the above database wants to send information of each of its products to each of its suppliers. For follow-up, the management wants a complete list in which each Supplier's detail is paired with each Product's detail. For this, the computer department can produce a list which is the Cartesian product of Product and Supplier tables, as follows:

```
SELECT *, '          ' AS Remarks FROM Product, Supplier;
```

to get the following report:

```
+------+-------------+----------+--------------+-------------+----------+
| Code | Name        | Sup_Code | Name         | Address     | Remarks  |
+------+-------------+----------+--------------+-------------+----------+
| P001 | Toothpaste  | S001     | DC & Company | Uttam Nagar |          |
| P001 | Toothpaste  | S002     | SURY Traders | Model Town  |          |
| P002 | Shampoo     | S001     | DC & Company | Uttam Nagar |          |
| P002 | Shampoo     | S002     | SURY Traders | Model Town  |          |
| P003 | Conditioner | S001     | DC & Company | Uttam Nagar |          |
| P003 | Conditioner | S002     | SURY Traders | Model Town  |          |
+------+-------------+----------+--------------+-------------+----------+
```

Equi- Join of tables :

The complete Cartesian product of two or more tables is, generally, not used directly. Sometimes the complete Cartesian product of two tables may give some confusing information also. For example, the first Cartesian product (CP-1) indicates that each order (Order Numbers 1 and 2) is placed for each Product (Code 'P001', 'P002', 'P003'). But this is incorrect!

Similar is the case with CP-2 and CP-3 also.

But we can extract meaningful information from the Cartesian product by placing some conditions in the statement. For example, to find out the product details corresponding to each Order details, we can enter the following statement:

```
SELECT * FROM order_table, product WHERE p_code = code;

+----------+--------+----------+------+-------------+
| Order_No | P_Code | Sup_Code | Code | Name        |
+----------+--------+----------+------+-------------+
|        1 | P001   | S002     | P001 | Toothpaste  |
|        2 | P002   | S002     | P002 | Shampoo     |
+----------+--------+----------+------+-------------+
```

Two table names are specified in the FROM clause of this statement, therefore MySQL creates a Cartesian product of the tables. From this Cartesian product MySQL selects only those records for which P_Code (Product code specified in the Order_table table) matches Code (Product code in the Product table). These selected records are then displayed.

It always happens that whenever we have to get the data from more than one tables, there is some common column based on which the meaningful data is extracted from the tables. We specify table names in the FROM clause of SELECT command. We also give the condition specifying the matching of common column. (When we say common column, it does not mean that the column names have to be the same. It means that the columns should represent the same data with the same data types.) Corresponding to this statement, internally the Cartesian product of the tables is made. Then based on the specified condition the meaningful data is extracted from this Cartesian product and displayed.

Let us take another example of producing a report which displays the supplier name and address corresponding to each order.

```
SELECT Order_No, Order_table.Sup_Code, Name, Address
    FROM order_table, supplier
    WHERE order_table.sup_code = supplier.sup_code;
```

```
+----------+----------+--------------+------------+
| Order_No | Sup_Code | Name         | Address    |
+----------+----------+--------------+------------+
|        1 | S002     | SURY Traders | Model Town |
|        2 | S002     | SURY Traders | Model Town |
+----------+----------+--------------+------------+
```

In this statement the tables referred are Order_table and Supplier. In these tables sup_code is the common column. This column exists with same name in both the tables. Therefore whenever we mention it, we have to specify the table from which we want to extract this column. This is known as qualifying the column name. If we don't qualify the common column name, the statement would result into an error due to the ambiguous the column names.

Following is another example of equi-join. This time with three tables.

```
Select Order_no, Product.name as Product, Supplier.Name as Supplier
    From order_table, Product, Supplier
        WHERE order_table.Sup_Code = Supplier.Sup_Code
        and P_Code = Code;
```

The output produced by this statement is:

```
+----------+-----------+--------------+
| Order_no | Product   | Supplier     |
+----------+-----------+--------------+
|        1 | Toothpaste | SURY Traders |
|        2 | Shampoo   | SURY Traders |
+----------+-----------+--------------+
```

Let us now get back to our original Shoe database and see how Ms. Akhtar uses the concept of joins to extract data from multiple tables.

For the situation MT-1, she writes the query:

```
SELECT order_no , name, phone
    FROM orders, customers
    WHERE orders.cust_code = customers.cust_code;
```

and get the following required output:

```
+----------+----------------+--------------------+
| order_no | name           | phone              |
+----------+----------------+--------------------+
|        1 | Novelty Shoes  | 4543556, 97878989  |
|        2 | Novelty Shoes  | 4543556, 97878989  |
|        5 | Novelty Shoes  | 4543556, 97878989  |
|        9 | Novelty Shoes  | 4543556, 97878989  |
|        4 | Aaram Footwear | NULL               |
|        6 | Aaram Footwear | NULL               |
|       10 | Aaram Footwear | NULL               |
|        3 | Foot Comfort   | 51917142, 76877888 |
|        7 | Pooja Shoes    | 61345432, 98178989 |
|        8 | Dev Shoes      | NULL               |
+----------+----------------+--------------------+
```

Following are the queries and corresponding outputs for the situations MT-2, MT-3, and MT-4 respectively:

```
SELECT order_no , Order_Qty, name, cost
    FROM orders, shoes WHERE Shoe_Code = code;
```

order_no	Order_Qty	name	cost
1	200	School Canvas	132.50
2	200	School Canvas	135.50
3	150	School Leather	232.50
4	250	School Leather	270.00
5	400	School Leather	232.50
6	300	Galaxy	640.00
7	200	Tracker	700.00
8	350	Galaxy	712.00
9	225	Galaxy	720.00
10	200	Tracker	800.50

```
SELECT name, address FROM orders, customers
WHERE orders.cust_code = customers.cust_code
and order_qty > 300;
```

name	address
Novelty Shoes	Raja Nagar, Bhopal
Dev Shoes	Mohan Nagar, Ghaziabad

```
SELECT order_no, Order_Qty, customers.name,
    cost*order_qty as 'Order Cost'
    FROM orders, shoes, Customers
    WHERE Shoe_Code = code
          and Orders.Cust_Code = Customers.Cust_Code
    order by order_no;
```

order_no	Order_Qty	name	Order Cost
1	200	Novelty Shoes	26500.00
2	200	Novelty Shoes	27100.00
3	150	Foot Comfort	34875.00
4	250	Aaram Footwear	67500.00
5	400	Novelty Shoes	93000.00
6	300	Aaram Footwear	192000.00
7	200	Pooja Shoes	140000.00
8	350	Dev Shoes	249200.00
9	225	Novelty Shoes	162000.00
10	200	Aaram Footwear	160100.00

Here is another statement extracting data from multiple tables. Try to find out what will be its output and then try this statement on computer and check whether you thought of the correct output.

```
SELECT order_no , Order_Qty, name, cost
    FROM orders, shoes
    WHERE Shoe_Code = code and order_qty > 200;
```

Foreign Key :

As we have just seen, in a join the data is retrieved from the Cartesian product of two tables by giving a condition of equality of two corresponding columns - one from each table. Generally, this column is the Primary Key of one table. In the other table this column is the Foreign key. Such a join which is obtained by putting a condition of equality on cross join is called an 'equi-join'. As an example, once again consider the Product, Supplier, and Order tables referenced earlier. For quick reference these tables are shown once again:

Product

Code	Name
P001	Toothpaste
P002	Shampoo
P003	Conditioner

Supplier

Sup_Code	Name	Address
S001	DC & Company	Uttam Nagar
S002	SURY Traders	Model Town

Order_table

Order_No	P_Code	Sup_Code
1	P001	S002
2	P002	S002

In these tables there is a common column between Product and Order_table tables (Code and P_Code respectively) which is used to get the Equi-Join of these two tables. Code is the Primary Key of Product table and in Order_table table it is not so (we can place more than one orders for the same product). In the order_table, P_Code is a Foreign Key. Similarly, Sup_Code is the primary key in Supplier table whereas it is a Foreign Key is Order_table table. A foreign key in a table is used to ensure referential integrity and to get Equi-Join of two tables.

Referential Integrity: Suppose while entering data in Order_table we enter a P_Code that does not exist in the Product table. It means we have placed an order for an item that does not exist! We should and can always avoid such human errors. Such errors are avoided by explicitly making P_Code a foreign key of Order_table table which always references the Product table to make sure that a non-existing product code is not entered in the Order_table table. Similarly, we can also make Sup_Code a Foreign key in Order_table table which always references Customer table to check validity of Cust_code. This can be done, but how to do it is beyond the scope of this book.

This property of a relational database which ensures that no entry in a foreign key column of a table can be made unless it matches a primary key value in the corresponding related table is called Referential Integrity.

Union

Union is an operation of combining the output of two SELECT statements. Union of two SELECT statements can be performed only if their outputs contain same number of columns and data types of corresponding columns are also the same. The syntax of UNION in its simplest form is:

```
SELECT <select_list>
    FROM <tablename>
    [WHERE <condition> ]
 UNION [ALL]
SELECT <select_list>
    FROM <tablename>
    [WHERE <condition> ];
```

Union does not display any duplicate rows unless ALL is specified with it.

Example:

Suppose a company deals in two different categories of items. Each category contains a number of items and for each category there are different customers. In the database there are two customer tables: Customer_Cat_1 and Customer_Cat_2. If it is required to produce a combined list of all the customers, then it can be done as follows:

```
SELECT Cust_Code from Customer_Cat_1
UNION
SELECT Cust_Code from Customer_Cat_2;
```

If a customer exists with same customer code in both the tables, its code will be displayed only once - because Union does display duplicate rows. If we explicitly want the duplicate rows, then we can enter the statement:

```
SELECT Cust_Code from Customer_Cat_1
UNION ALL
SELECT Cust_Code from Customer_Cat_2;
```

Constraints

Many times it is not possible to keep a manual check on the data that is going into the tables using INSERT or UPDATE commands. The data entered may be invalid. MySQL provides some rules, called Constraints, which help us, to some extent, ensure validity of the data. These constraints are:

S.No.	Constraint	Purpose
1.	PRIMARY KEY	Sets a column or a group of columns as the Primary Key of a table. Therefore, NULLs and Duplicate values in this column are not accepted.
2.	NOT NULL	Makes sure that NULLs are not accepted in the specified column.
3.	FOREIGN KEY	Data will be accepted in this column, if same data value exists in a column in another related table. This other related table name and column name are specified while creating the foreign key constraint.
4.	UNIQUE	Makes sure that duplicate values in the specified column are not accepted.
5.	ENUM	Defines a set of values as the column domain. So any value in this column will be from the specified values only.
6.	SET	Defines a set of values as the column domain. Any value in this column will be a seubset of the specied set only.

We shall discuss only the PRIMARY KEY and NOT NULL constraints in this book. Other constraints are beyond the scope of this book.

PRIMARY KEY:

Recall that primary key of a table is a column or a group of columns that uniquely identifies a row of the table. Therefore no two rows of a table can have the same primary key value. Now suppose that the table Shoes is created with the following statement:

```
CREATE TABLE Shoes
    (Code CHAR(4), Name VARCHAR(20), type VARCHAR(10),
    size INT(2), cost DECIMAL(6,2), margin DECIMAL(4,2),
    Qty INT(4));
```

We know that in this table Code is the Primary key. But, MySQL does not know that! Therefore it is possible to enter duplicate values in this column or to enter NULLs in this column. Both these situations are unacceptable.

To make sure that such data is not accepted by MySQL, we can set Code as the primary key of Shoes table. It can be done by using the PRIMARY KEY clause at the time of table creation as follows:

```
CREATE TABLE Shoes
    (Code CHAR(4) PRIMARY KEY, Name VARCHAR(20),
    type VARCHAR(10), size INT(2), cost DECIMAL(6,2),
    margin DECIMAL(4,2), Qty INT(4));
```

or as follows:

```
CREATE TABLE Shoes
    (Code CHAR(4), Name VARCHAR(20), type VARCHAR(10),
    size INT(2), cost DECIMAL(6,2), margin DECIMAL(4,2),
    Qty INT(4), PRIMARY KEY (Code));
```

To create a table Bills with the combination of columns Order_No and Cust_Code as the primary key, we enter the statement:

```
CREATE TABLE bills
    (Order_Num INT(4) PRIMARY KEY,
    cust_code VARCHAR(4) PRIMARY KEY,
    bill_Date DATE, Bill_Amt DECIMAL(8,2));
```

Contrary to our expectation, we get an error (Multiple primary key defined) with this statement. The reason is that MySQL interprets this statement as if we are trying to create two primary keys of the table - Order_Num, and Cust_code. But a table can have at most one primary key. To set this combination of columns a primary key we have to enter the statement as follows:

```
CREATE TABLE bills
    (Order_Num INT(4), cust_code VARCHAR(4),
    bill_Date date, Bill_Amt DECIMAL(8,2),
    PRIMARY KEY(Order_Num, cust_code));
```

Let us now check the table structure with the command: DESC bills;

The table structure is as shown below:

```
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| Order_Num | INT(4)       | NO   | PRI | 0       |       |
| cust_code | VARCHAR(4)   | NO   | PRI |         |       |
| bill_Date | date         | YES  |     | NULL    |       |
| Bill_Amt  | DECIMAL(8,2) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
```

These columns constitute the primary key of the table

NULLs cannot be accepted in these columns.

NOT NULL:

Many times there are some columns of a table in which NULL values should not be accepted. We always want some known valid data values in these columns. For example, we cannot have an order for which the customer code is not known. It means whenever we enter a row in the orders table, corresponding customer code cannot be NULL. Similarly while entering records in the Shoes table, we have to mention the Shoe size, it cannot be set NULL. There may be any number of such situations.

While creating a table we can specify in which columns NULLs should not be accepted as follows:

```
CREATE TABLE Shoes
    (Code CHAR(4) PRIMARY KEY, Name VARCHAR(20),
    type VARCHAR(10), size INT(2) NOT NULL,
    cost DECIMAL(6,2), margin DECIMAL(4,2), Qty INT(4));
CREATE TABLE bills
    (Order_Num INT(4), cust_code VARCHAR(4),
    bill_Date DATE, Bill_Amt DECIMAL(8,2) NOT NULL,
    PRIMARY KEY (Order_Num, cust_code));
```

Now if we try to enter a NULL in the specified column, MySQL will reject the entry and give an error.

Viewing Constraints, Viewing the Columns Associated with Constraints :

After creating a table, we can view its structure using DESC command. The table structure also includes the constraints, if any. Therefore, when we use DESC command, we are shown the table structure as well as constraints, if any. A constraint is shown beside the column name on which it is applicable. E.g., the statement:

```
DESC Shoes;
```

displays the table structure as follows:

```
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| Code   | CHAR(4)      | NO   | PRI | NULL    |       |
| Name   | VARCHAR(20)  | YES  |     |         |       |
| type   | VARCHAR(10)  | YES  |     | NULL    |       |
| size   | INT(2)       | NO   |     | 0       |       |
| cost   | DECIMAL(6,2) | YES  |     | NULL    |       |
| margin | DECIMAL(4,2) | YES  |     | NULL    |       |
| Qty    | INT(4)       | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
```

ALTER TABLE

In class XI, we have studied that a new column can be added to a table using ALTER TABLE command. Now we shall study how ALTER TABLE can be used:

- to add a constraint
- to remove a constraint
- to remove a column from a table
- to modify a table column

Add, Modify, and Remove constraints :

If we create a table without specifying any primary key, we can still specify its primary key by ALTER TABLE command. Suppose we have created the Shoes table without specifying any Primary key, then later we can enter the statement as follows:

```
ALTER TABLE Shoe ADD PRIMARY KEY(code);
```

This will set Code as the primary key of the table. But if the Code column already contains some duplicate values, then this statement will give an error.

In MySQL, it is also possible to change the primary key column(s) of a table. Suppose, in the Shoes table, istread of Code, we want to set the combination of 'Name' and 'Size' as the primary key. For this first we have to DROP the already existing primary key (i.e., Code) and then add the new primary key (i.e., Name and Size). The corresponding statements are as follows:

```
ALTER TABLE Shoes DROP PRIMARY KEY;
```

After this statement, there is no primary key of Shoe table. Now we can add the new primary key as follows:

```
ALTER TABLE Shoe ADD PRIMARY KEY (Name, Size);
```

Now if we see the table structure by DESC Shoes; statement, it will be shown as follows:

```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Code   | CHAR(4)     | NO   |     | NULL    |       |
| Name   | VARCHAR(20) | NO   | PRI |         |       |
| type   | VARCHAR(10) | YES  |     | NULL    |       |
| size   | INT(2)      | NO   | PRI | 0       |       |
| cost   | DECIMAL(6,2)| YES  |     | NULL    |       |
| margin | DECIMAL(4,2)| YES  |     | NULL    |       |
| Qty    | INT(4)      | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

In MySQL, it is not possible to add or drop NOT NULL constraint explicitly after the table creation. But it can be done using MODIFY clause of ALTER TABLE command. As an example, suppose we don't want to accept NULL values in bill_date column of bills table, we can issue the statement:

```
ALTER TABLE bills MODIFY bill_date DATE NOT NULL;
```

Later on if we wish to change this status again, we can do so by entering the command:

```
ALTER TABLE bills MODIFY bill_date DATE NULL;
```

Remove and Modify columns :

ALTER TABLE can be used to remove a column from a table. This is done using DROP clause in ALTER TABLE command. The syntax is as follws:

```
ALTER TABLE <tablename> DROP <columnname>
    [, DROP <columnname> [, DROP <columnname> [, . . . ]]];
```

Following are some self explanatory examples of SQL statemenets to remove columns from tables:

```
ALTER TABLE Shoes DROP Qty;

ALTER TABLE Orders DROP Cust_Code;

ALTER TABLE Student DROP Class, DROP RNo, DROP Section;
```

Although any column of a table can be removed, MySQL puts the restriction that a primary key column can be removed only if the remaining, primary key columns, if any, do not contain any duplicate entry. This can be understood more clearly with the help of following example:

The Name and Size columns of the Shoe table constitute its primary key. Now if we drop the Name column from the table, Size will be the remaining Primary Key column of the table. Therefore, duplicate entries in the Size column should not be allowed. To ensure this, before removing Name column from the table, MySQL checks that there are no duplicate entries present in the Size column of the table. If there are any, then the statement trying to remove Name column from the table will result in an error and the Name column will not be removed. If there are no duplicate enteries in the Size column, then Name column will be removed. Similar will be the case with the Name column, if we try to remove Size column. But there won't be any problem if we try to remove both the primary key columns simultaneously with one ALTER TABLE statement as follows:

```
ALTER TABLE Shoes DROP name, DROP size;
```

ALTER TABLE can also be used to change the data type of a table column. For this the syntax is as follows:

```
ALTER TABLE <tablename> MODIFY <col_name> <new datatype>
    [,MODIFY <col_name> <new datatype>
        [,MODIFY <col_name> <new data type>  [, . . . ]]];
```

e.g., the statement:

```
ALTER TABLE shoes modify code CHAR(5), modify type VARCHAR(20);
```

changes the data type of column Code to CHAR(5) and that of type to VARCHAR(20).

When we give a statement to chage the data type of a column, MySQL executes that statement correctly only if the change in data type does not lead to any data loss. E.g., if we try to change the data type of order_date column of orders table from date to int, we'll get an error. This is because the data already stored in this column cannot be converted into int type. Similarly, if a column of VARCHAR(10) type conatins some data value which is 10 characters long, then the data type of this column cannot be converted to VARCHAR(n), where n is an integer less than 10.

DROP TABLE

Sometimes there is a requirement to remove a table from the database. In such cases we don't want merely to delete the data from the table, but we want to delete the table itself. DROP TABLE command is used for this purpose. The syntax of DROP TABLE command is as follows:

```
DROP TABLE <tablename>;
```

e.g. to remove the table Orders from the database we enter the statement:

```
DROP TABLE Orders;
```

And after this statement orders table is no longer available in the database. It has been removed.

Summary

- **Aggregate or Group functions:** MySQL provides Aggregate or Group functions which work on a number of values of a column/expression and return a single value as the result. Some of the most frequently used. Aggregate functions in MySQL are : MIN(), MAX(), AVG(), SUM(), COUNT().
- **Data Types in aggregate functions:** MIN(), MAX(), and COUNT() work on any type of values - Numeric, Date, or String. AVG(), and SUM() work on only Numeric values (INT and DECIMAL).
- **NULLs in aggregate functions:** Aggregate functions totally ignore NULL values present in a column.
- **GROUP BY:** GROUP BY clause is used in a SELECT statement in conjunction with aggregate functions to group the result based on distinct values in a column.
- **HAVING:** HAVING clause is used in conjuction with GROUP BY clause in a SELECT statement to put condition on groups.
- **WHERE Vs HAVING:** WHERE is used to put a condition on individual row of a table whereas HAVING is used to put condition on individual group formed by GROUP BY clause in a SELECT statement.

- **Cartesian Product (or Cross Join):** Cartesian product of two tables is a table obtained by pairing each row of one table with each row of the other. A cartesian product of two tables contains all the columns of both the tables.
- **Equi-Join:** An equi join of two tables is obtained by putting an equality condition on the Cartesian product of two tables. This equality condition is put on the common column of the tables. This common column is, generally, primary key of one table and foreign key of the other.
- **Foreign Key:** It is a column of a table which is the primary key of another table in the same database. It is used to enforce referential integrity of the data.
- **Referential Integrity:** The property of a relational database which ensures that no entry in a foreign key column of a table can be made unless it matches a primary key value in the corresponding column of the related table.
- **Union:** Union is an operation of combining the output of two SELECT statements.
- **Constraints:** These are the rules which are applied on the columns of tables to ensure data integrity and consistency.
- **ALTER TABLE:** ALTER TABLE command can be used to Add, Remove, and Modify columns of a table. It can also be used to Add and Remove constraints.
- **DROP TABLE:** DROP TABLE command is used to delete tables.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. Which of the following will give the same answer irrespective of the NULL values in the specified column:
   a. MIN()
   b. MAX()
   c. SUM()
   d. None of the above
2. An aggregate function:
   a. Takes a column name as its arguments
   b. May take an expression as its argument

c. Both (a) and (b)

d. None of (a) and (b)

3. HAVING is used in conjunction with

a. WHERE
b. GROUP BY clause
c. Aggregate functions
d. None of the above

4. In the FROM clause of a SELECT statement

a. Multiple Column Names are specified.

b. Multiple table names are specified.

c. Multiple Column Names may be specified.

d. Multiple table names may be specified.

5. JOIN in RDBMS refers to

a. Combination of multiple columns
b. Combination of multiple rows
c. Combination of multiple tables
d. Combination of multiple databases

6. Equi-join is formed by equating

a. Foreign key with Primary key
b. Each row with all other rows
c. Primary key with Primary key
d. Two tables

7. Referential integrity

a. Must be maintained

b. Cannot be maintained

c. Is automatically maintained by databases

d. Should not be maintained

8. A Primary key column

a. Can have NULL values
b. Can have duplicate values
c. Both (a) and (b)
d. Neither (a) nor (b)

9. Primary Key of a table can be
   a. Defined at the time of table creation only.
   b. Defined after table creation only.
   c. Can be changed after table creation
   d. Cannot be changed after table creation
10. Two SELECT commands in a UNION
   a. Should select same number of columns.
   b. Should have different number of columns
   c. Both (a) and (b)
   d. Neither (a) nor (b)

ANSWER THE FOLLOWING QUESTIONS

1. Why are aggregate functions called so? Name some aggregate functions.
2. Why is it not allowed to give String and Date type arguments for SUM() and AVG() functions? Can we give these type of arguments for other functions?
3. How are NULL values treated by aggregate functions?
4. There is a column C1 in a table T1. The following two statements:

   ```
   SELECT COUNT(*) FROM T1;    and SELECT COUNT(C1) from T1;
   ```

   are giving different outputs. What may be the possible reason?
5. What is the purpose of GROUP BY clause?
6. What is the difference between HAVING and WHERE clauses? Explain with the help of an example.
7. What is the Cartesian product of two table? Is it same as an Equi-join?
8. There are two table T1 and T2 in a database. Cardinality and degree of T1 are 3 and 8 respectively. Cardinality and degree of T2 are 4 and 5 respectively. What will be the degree and Cardinality of their Cartesian product?

9. What is a Foreign key? What is its importance?
10. What are constraints? Are constraints useful or are they hinderance to effective management of databases?
11. In a database there is a table Cabinet. The data entry operator is not able to put NULL in a column of Cabinet? What may be the possible reason(s)?
12. In a database there is a table Cabinet. The data entry operator is not able to put duplicate values in a column of Cabinet? What may be the possible reason(s)?
13. Do Primary Key column(s) of a table accept NULL values?
14. There is a table T1 with combination of columns C1, C2, and C3 as its primary key? Is it possible to enter:
    a. NULL values in any of these columns?
    b. Duplicate values in any of these columns?
15. At the time of creation of table X, the data base administrator specified Y as the Primary key. Later on he realized that instead of Y, the combination of column P and Q should have been the primary key of the table. Based on this scenario, answer the following questions:
    a. Is it possible to keep Y as well as the combination of P and Q as the primary key?
    b. What statement(s) should be entered to change the primary key as per the requirement.
16. Does MySQL allow to change the primary key in all cases? If there is some special case, please mention.
17. What are the differences between DELETE and DROP commands of SQL?

LAB EXERCISES

1. In a database create the following tables with suitable constraints :

STUDENTS

AdmNo	Name	Class	Sec	RNo	Address	Phone
1271	Utkarsh Madaan	12	C	1	C-32, Punjabi Bagh	4356154
1324	Naresh Sharma	10	A	1	31, Mohan Nagar	435654
1325	Md. Yusuf	10	A	2	12/21, Chand Nagar	145654
1328	Sumedha	10	B	23	59, Moti Nagar	4135654
1364	Subya Akhtar	11	B	13	12, Janak Puri	NULL
1434	Varuna	12	B	21	69, Rohini	NULL
1461	David DSouza	11	B	1	D-34, Model Town	243554, 98787665
2324	Satinder Singh	12	C	1	1/2, Gulmohar Park	143654
2328	Peter Jones	10	A	18	21/32B, Vishal Enclave	24356154
2371	Mohini Mehta	11	C	12	37, Raja Garden	435654, 6765787

SPORTS

AdmNo	Game	CoachName	Grade
1324	Cricket	Narendra	A
1364	Volleball	M.P. Singh	A
1271	Volleball	M.P. Singh	B
1434	Basket Ball	I. Malhotra	B
1461	Cricket	Narendra	B
2328	Basket Ball	I. Malhotra	A
2371	Basket Ball	I. Malhotra	A
1271	Basket Ball	I. Malhotra	A
1434	Cricket	Narendra	A
2328	Cricket	Narendra	B
1364	Basket Ball	I. Malhotra	B

a) Based on these tables write SQL statements for the following queries:

   i. Display the lowest and the highest classes from the table STUDENTS.

   ii. Display the number of students in each class from the table STUDENTS.

   iii. Display the number of students in class 10.

   iv. Display details of the students of Cricket team.

v. Display the Admission number, name, class, section, and roll number of the students whose grade in Sports table is 'A'.

vi. Display the name and phone number of the students of class 12 who are play some game.

vii. Display the Number of students with each coach.

viii. Display the names and phone numbers of the students whose grade is 'A' and whose coach is Narendra.

b) Identify the Foreign Keys (if any) of these tables. Justify your choices.

c) Predict the the output of each of the following SQL statements, and then verify the output by actually entering these statements:

i. SELECT class, sec, count(*) FROM students GROUP BY class, sec;

ii. SELECT Game, COUNT(*) FROM Sports GROUP BY Game;

iii. SELECT game, name, address FROM students, Sports

WHERE students.admno = sports.admno AND grade = 'A';

iv. SELECT Game FROM students, Sports

WHERE students.admno = sports.admno AND Students.AdmNo = 1434;

2. In a database create the following tables with suitable constraints :

ITEMS

I_Code	Name	Category	Rate
1001	Masala Dosa	South Indian	60
1002	Vada Sambhar	South Indian	40
1003	Idli Sambhar	South Indian	40
2001	Chow Mein	Chinese	80
2002	Dimsum	Chinese	60
2003	Soup	Chinese	50
3001	Pizza	Italian	240
3002	Pasta	Italian	125

BILLS

BillNo	Date	I_Code	qty
1	2010-04-01	1002	2
1	2010-04-01	3001	1
2	2010-04-01	1001	3
2	2010-04-01	1002	1
2	2010-04-01	2003	2
3	2010-04-02	2002	1
4	2010-04-02	2002	4
4	2010-04-02	2003	2
5	2010-04-03	2003	2
5	2010-04-03	3001	1
5	2010-04-03	3002	3

a) Based on these tables write SQL statements for the following queries:

i. Display the average rate of a South Indian item.

ii. Display the number of items in each category.

iii. Display the total quantity sold for each item.

iv. Display total quanity of each item sold but don't display this data for the items whose total quantity sold is less than 3.

v. Display the details of bill records along with Name of each corresponding item.

vi. Display the details of the bill records for which the item is 'Dosa'.

vii. Display the bill records for each Italian item sold.

viii. Display the total value of items sold for each bill.

b) Identify the Foreign Keys (if any) of these tables. Justify your answer.

c) Answer with justification (Think independently. More than one answers may be correct. It all depends on your logical thinking):

i. Is it easy to remember the Category of item with a given item code? Do you find any kind of pattern in the items code? What could be the item code of another South Indian item?

ii. What can be the possible uses of Bills table? Can it be used for some analysis purpose?

iii. Do you find any columns in these tables which can be NULL? Is there any column which must not be NULL?

3. In a database create the following tables with suitable constraints :

VEHICLE

Field	Type	Null	Key	Default	Extra
RegNo	char(10)	NO	PRI		
RegDate	date	YES		NULL	
Owner	varchar(30)	YES		NULL	
Address	varchar(50)	YES		NULL	

CHALLAN

Field	Type	Null	Key	Default	Extra
Challan_No	int(11)	NO	PRI	0	
Ch_Date	date	YES		NULL	
RegNo	char(10)	YES		NULL	
Offence	int(3)	YES		NULL	

OFFENCE

Field	Type	Null	Key	Default	Extra
Offence_Code	int(3)	NO	PRI	0	
Off_desc	varchar(30)	YES		NULL	
Challan_Amt	int(4)	YES		NULL	

a) Based on these tables write SQL statements for the following queries:

i. Display the dates of first registration and last registration from the table Vehicle.

ii. Display the number of challans issued on each date.

iii. Display the total number of challans issued for each offence.

iv. Display the total number of vehicles for which the 3rd and 4th characters of RegNo are '6C'.

v. Display the total value of challans issued for which the Off_Desc is 'Driving without License'.

vi. Display details of the challans issued on '2010-04-03' along with Off_Desc for each challan.

vii. Display the RegNo of all vehicles which have been challaned more than once.

viii. Display details of each challan alongwith vehicle details, Off_desc, and Challan_Amt.

b) Identify the Foreign Keys (if any) of these tables. Justify your choices.

c) Should any of these tables have some more column(s)? Think, discuss in peer groups, and discuss with your teacher.

4. In a database create the following tables with suitable constraints:

Table: Employee

No	Name	Salary	Zone	Age	Grade	Dept
1	Mukul	30000	West	28	A	10
2	Kritika	35000	Centre	30	A	10
3	Naveen	32000	West	40		20
4	Uday	38000	North	38	C	30
5	Nupur	32000	East	26		20
6	Moksh	37000	South	28	B	10
7	Shelly	36000	North	26	A	30

Table: Department

Dept	DName	MinSal	MaxSal	HOD
10	Sales	25000	32000	1
20	Finance	30000	50000	5
30	Admin	25000	40000	7

a) Based on these tables write SQL statements for the following queries:

i. Display the details of all the employees who work in Sales department.

ii. Display the Salary, Zone, and Grade of all the employees whose HOD is Nupur.

iii. Display the Name and Department Name of all the employees.

iv. Display the names of all the employees whose salary is not within the specified range for the corresponding department.

v. Display the name of the department and the name of the corresponding HOD for all the departments.

b) Identify the Foreign Keys (if any) of these tables. Justify your choices.

TEAM BASED TIME BOUND EXERCISE:

(Team size recommended: 3 students each team)

1. A chemist shop sells medicines manufactured by various pharmaceutical companies. When some medicine is sold, the corresponding stock decreases and when some medicines are bought (by the chemist shop) from their suppliers, the corresponding stock increases. Now the shop wants to keep computerized track of its inventory. The shop owner should be able to find

   - The current stock of any medicine.
   - The total sale amount of any specific time period (a specific day, or month, or any period between two specific dates)
   - The details of all the medicines from a specific supplier.

- The details of all the medicines from a specific manufacturer.
- Total value of the medicines in the stock.

There may be a number of other reports which the shop owner may like to have.

The job of each team is to design a database for this purpose. Each team has to specify:

- The structure (with constraints) of each of the tables designed (with justification).
- How the tables are related to each other (foreign keys).
- How the design will fulfill all the mentioned requirements.
- At least 10 reports that can be generated with the database designed.

2. To expand its business, XYZ Mall plans to go online. Anyone who shops at the Mall will be given a membership number and Password which can be used for online shopping. With this membership number and password, customers can place their orders online. The mall will maintain the customers' data and orders' data. A person is put on duty to keep constantly checking the Orders data. Whenever an order is received, its processing has to start at the earliest possible.

   The Orders' data will be analysed periodically (monthly, quarterly, annually - whatever is suitable) to further improve business and customer satisfaction.

   The job of each team is to design a database for this purpose. Each team has to specify:

   - The structure (with constraints) of each of the tables designed (with justification).
   - How the tables are related to each other (foreign keys).
   - How the design will fulfill all the mentioned requirements.
   - At least 10 reports that can be generated with the database designed.

ADVANCED RDBMS CONCEPTS

Learning Objectives

After studying this lesson the students will be able to:

- Define a Transaction
- Describe reason why all the tasks in a transaction should be executed fully or not at all.
- Perform basic transactions.
- Commit a transaction.
- Add Save Points to a transaction.
- Roll back a Transaction
- Roll back a Transaction to a Savepoint.

Till now we have studied about various SQL statements manipulating data stored in a MySQL database. We executed SQL statements without concern about inconsistencies arising due to group of statements not being executed in entirety. In this lesson, we will study the basic concepts of Transaction processing and how MySQL ensures consistency of data when a group of statements is executed.

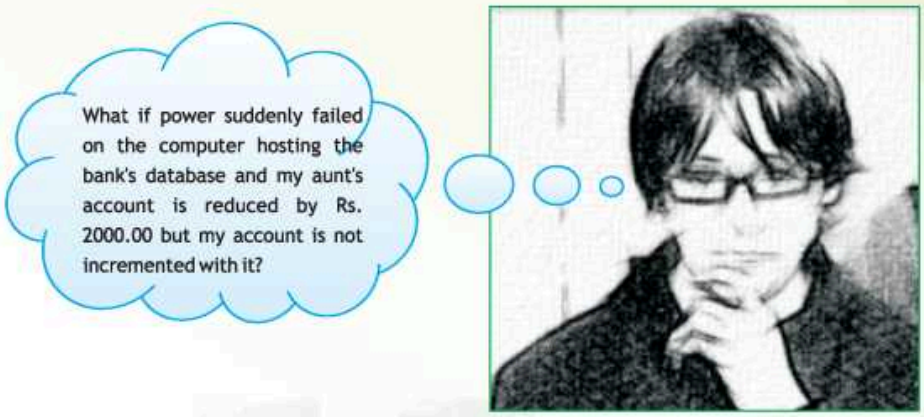
Puzzle

Vijaya has to withdraw ₹ 2500.00 from her account in the bank. She asked for m notes of ₹ 50.00 and n notes of ₹ 100.00. The cashier made a mistake and handed her m notes of ₹ 100.00 and n notes of ₹ 50.00. When she returned back home she realized that she got ₹ 500.00 less. How many notes of ₹ 50.00 and ₹ 100.00 did she ask for?

Introduction

Raunak studies in Class XII. He is very helpful. During summer vacations, he helped his aunt's son in his studies. His aunt was very pleased with him and gave him a cheque of ₹ 2000.00. Raunak knows that after presentation of the cheque to the bank, his aunt's account will be reduced by ₹ 2000.00 and his account will be increased by ₹ 2000.00. Raunak walked up to the bank to present the cheque.

While returning from bank, Raunak is apprehensive about one thing:



Are Raunak's fears valid? Can you recollect something like this happening with you or with your friends or relatives?

Raunak should not fear at all as the Bank's DBMS looks after these eventualities. DBMSs ensure consistency (correctness) of data by managing Transactions.

DBMS and Transaction Management

Mostly customers view an operation like transfer of funds as a single operation but actually it consists of series of operations.

Suppose Raunak's account number is 3246 and his aunt's account number is 5135. In order to process the cheque presented by Raunak, the following two SQL commands need to be executed on the database maintained by the bank:

```
UPDATE Savings
SET balance = balance - 2000        <-- For Aunt's account
WHERE account_no = 5135;

UPDATE Savings
SET balance = balance + 2000        <-- For Raunak's account
WHERE account_no = 3246;
```

The above two Updates should both take place. If the first Update takes place and there is a system failure, the first updation should be undone. Either both the updations should be done and if it is not possible for both the updations to be done, then no updation should be done.

What is a Transaction?

A Transaction is a unit of work that must be done in logical order and successfully as a group or not done at all. Unit of work means that a Transaction consists of different tasks - but together they are considered as one unit. Each transaction has a beginning and an end. If anything goes wrong in between the execution of transaction, the entire transaction (No matter to what extent has been done) should be cancelled. If it is successful, then the entire transaction should be saved to the database.

> A transaction is a unit of work that must be done in logical order and successfully as a group or not done at all.

In Raunak's case, both the updation statements constitute a transaction. Both are together treated as a single unit.

To understand how transactions are managed, let us study the following 3 statements of SQL:

- START TRANSACTION statement
- COMMIT statement
- ROLLBACK statement

START TRANSACTION Statement :

START TRANSACTION statement commits the current transaction and starts a new transaction. It tells MySQL that the transaction is beginning and the statements that follow should be treated as a unit, until the transaction ends. It is written like this:

```
START TRANSACTION;
```

> The **START TRANSACTION** statement has no clauses.

COMMIT Statement :

The COMMIT statement is used to save all changes made to the database during the transaction to the database. Commit statement is issued at a time when the transaction is complete- all the changes have been successful and the changes should be saved to the database. COMMIT ends the current transaction.

COMMIT statement is used like this:

```
COMMIT;
```

Or

```
COMMIT WORK;
```

Here WORK is a keyword and is optional.

In the following example, the table named savings has 2 rows. A transaction is started and balance in Siddharth's account (with account number 1004) is increased by Rs. 2000.00 and the balance in Akriti's account (with account number 1006) is decreased by Rs. 2000.00. COMMIT statement makes the changes made by the transaction permanent.

Example 1:

```
mysql> select * from savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 87000.00 |
|       1006 | Akriti Malik     | 87000.00 |
+------------+------------------+----------+
mysql> START TRANSACTION;
mysql> UPDATE Savings
    -> SET balance = balance + 2000
    -> WHERE account_no = 1004;
mysql> UPDATE Savings
    -> SET balance = balance - 2000
    -> WHERE account_no = 1006;
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 89000.00 |
|       1006 | Akriti Malik     | 85000.00 |
+------------+------------------+----------+
2 rows in set (0.00 sec)
mysql> COMMIT;
```

ROLLBACK Statement :

When a transaction is being executed, some type of error checking is usually performed to check whether it is executing successfully or not. If not, the entire transaction is undone using the ROLLBACK statement. The ROLLBACK statement cancels the entire transaction i.e. It rolls the transaction to the beginning. It aborts any changes made during the transaction and the state of database is returned to what it was before the transaction began to execute and does not save any of the changes made to the database during the transaction.

ROLLBACK statement is used like this:

```
ROLLBACK;
```

Or

```
ROLLBACK WORK;
```

Here WORK is a keyword and is optional.

If in Example 1 shown above ROLLBACK was used instead of COMMIT, the updation of incrementing Siddharth's account by ₹ 2000.00 and decrementing Akriti's account by ₹ 2000 wouldn't have taken place. Let us now initiate a transaction, increase Akriti's account by ₹ 3000.00, then Rollback the transaction and see what happens to the updation done on Akriti's account.

```
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 89000.00 |
|       1006 | Akriti Malik     | 85000.00 |
+------------+------------------+----------+
mysql> START TRANSACTION;
mysql> UPDATE Savings
```

Before the transaction starts, Siddharth's balance is Rs. 89000 and Akriti's balance is Rs. 85000.00

```
    -> SET balance = balance + 3000
    -> WHERE account_no = 1006;
mysql> ROLLBACK;
mysql> SELECT * FROM Savings;
+------------+-----------------+----------+
| account_no | name            | balance  |
+------------+-----------------+----------+
|       1004 | Siddharth Sehgal | 89000.00 |
|       1006 | Akriti Malik     | 85000.00 |
+------------+-----------------+----------+
```

Akriti's balance is increased by Rs. 3000.00

Because of the Rollback, Akriti's balance is not updated and is displayed as it was before the transaction started.

- After the ROLLBACK command is issued to the database, the database itself starts a new transaction; though no explicit command of starting a transaction like START TRANSACTION is issued.

Example 2:

Let us try out some more SQL statements on Savings table to understand transactions well.

```
mysql> SELECT * FROM savings;
+------------+-----------------+----------+
| account_no | name            | balance  |
+------------+-----------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
|       1009 | Raunak Singh     | 56000.00 |
+------------+-----------------+----------+
```

```
mysql> INSERT INTO Savings VALUES
    (1010,'Lakshmi Swamy',34000);
mysql> START TRANSACTION;
mysql> UPDATE Savings SET balance =
balance +2000 WHERE account_no = 1010;
mysql> ROLLBACK;
mysql> SELECT * FROM Savings;
+------------+-----------------+----------+
| account_no | name            | balance  |
+------------+-----------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
|       1009 | Raunak Singh     | 56000.00 |
|       1010 | Lakshmi Swamy    | 34000.00 |
+------------+-----------------+----------+
5 rows in set (0.00 sec)
```

Start transaction statement starts a transaction and commits the previous INSERT INTO statement.

Rollback cancels the effect of Update statement.

SELECT statement displays Lakshmi Swamy's row with balance of 34000.00

Inserting SavePoints :

The SAVEPOINT statement defines a marker in a transaction. These markers are useful in rolling back a transaction till the marker.

We can add a savepoint anywhere in a transaction. When you roll back to that savepoint, any changes made to the database after the savepoint are discarded, and any changes made prior to the savepoint are saved. It is like semicomitting a transaction.

To define a savepoint, we enter the SAVEPOINT statement like this:

```
SAVEPOINT <savepoint-name>;
```

Example :

```
SAVEPOINT Mark1;
```

In the above statement a marker (savepoint) with the name Mark1 is defined. It becomes a bookmark in the transaction. Now we can write the following statement:

```
ROLLBACK TO SAVEPOINT Mark1;
```

to rollback the transaction till the bookmark named Mark1.

Setting Autocommit :

By default, Autocommit mode is on in MySQL. It means that MySQL does a COMMIT after every SQL statement that does not return an error. If it returns an error then either Rollback or Commit happens depending on the type of error. If we do not want individual statements of SQL to be automatically committed, we should set the autocommit mode to off.

When Autocommit is off then we have to issue COMMIT statement explicitly to save changes made to the database.

The following statement sets the autocommit mode to off. It also starts a new transaction

```
SET AUTOCOMMIT=0;
```

The following statement sets the autocommit mode to ON. It also commits and terminates the current transaction.

```
SET AUTOCOMMIT=1;
```

If autocommit is set to ON. we can still perform a multiple-statement transaction by starting it with an explicit START TRANSACTION statement and ending it with COMMIT or ROLLBACK.

Let us look at the following example to understand it:

Example

```
mysql> SET AUTOCOMMIT = 0;
```

Autocommit is disabled.

```
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
+------------+------------------+----------+
```

Table Savings has 3 rows.

```
mysql> INSERT INTO Savings values
(1009,'Raunak Singh', 56000);
```

Another row for Raunak Singh added.

```
mysql> ROLLBACK;
```

Insert statement was not committed so it is undone by Rollback

```
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
+------------+------------------+----------+
```

Table does not show Raunak Singh's row.

```
mysql> SET AUTOCOMMIT = 1;
```

Autocommit is enabled.

```
mysql> INSERT INTO Savings VALUES
(1009,'Raunak Singh', 56000);
```

Raunak's row is added and is committed too.

```
mysql> ROLLBACK;
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
|       1009 | Raunak Singh     | 56000.00 |
+------------+------------------+----------+
```

Rollback cannot undo insertion of Raunak's row.

If the autocommit mode has been set to off in a session and you end that session, the autocommit mode is automatically set to on when you start a new session.

Let us try out some more SQL statements :

Example

```
mysql> SET AUTOCOMMIT = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM Savings WHERE account_no = 1006;
Query OK, 1 row affected (0.00 sec)

mysql> ROLLBACK WORK;
Query OK, 0 rows affected (0.03 sec)
```

Autocommit is enabled

Start transaction sets autocommit off.

Row with account_no 1006 deleted but is not committed.

Deletion of Row with account_no 1006 is cancelled.

An implicit COMMIT takes place, even if AUTOCOMMIT is set OFF, on the database when the user issues a Data Definition language command like CREATE TABLE, ALTER TABLE etc .

Future Trends

Cloud Computing

A major shift is predicted in the IT industry in the way that software and services are going to be delivered in future. In this regard, cloud computing is an emerging area these days. It is the computing model where the infrastructure and the applications are offered as a service over the Internet. Cloud computing takes place out on someone else's network. Since the details of how it is set up or how it works is hidden from the user, the term cloud is used. Cloud means a large network that is away and is not in our control. Just about any database can be run in a cloud-based infrastructure. Search the web and find out how RDBMSs like MySQL will be significant in such a computing model.

Summary

- Work done during a transaction is a series of operations.
- If one of the operations of a transaction is not executed successfully, then the entire transaction should be cancelled. If all the operations are executed successfully, the transaction should be saved to a database.
- START TRANSACTION statement is used to start a transaction.
- The process of cancelling a transaction is called Rolling back.
- ROLLBACK statement is used to terminate a transaction and roll back the database to its original state before the transaction.
- COMMIT statement is used to save changes to the database.
- When AutoCommit is ON, each SQL statement is a transaction. The changes resulting from each statement are automatically committed.

- When Auto Commit is Off then changes made to database are not committed unless explicitly requested.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. A _____is a logical unit of work that must succeed or fail in its entirety.
   a) Primary key
   b) Database
   c) Transaction
   d) none of these
2. When AutoCommit is _____ ,changes made to database are not committed unless explicitly requested.
   a) Equal to "-"
   b) on
   c) off
   d) Equal to "%"
3. When a CREATE TABLE command is issued, a _____occurs on the database.
   a) ROLLBACK
   b) COMMIT
   c) SAVEPOINT
   d) ROLLBACK TO SAVEPOINT
4. By default AUTOCOMMIT is _____.
   a) disabled
   b) enabled

c) inactive

d) none of the above

5. Which of the following statement or command? Completes a transaction?

a) INSERT INTO

b) COMMIT

c) DELETE

d) SELECT

6. If Feroze deposits a cheque of Rs. 1200.00 in his account, which was given to him by Ali, two tasks: decreasing of Rs. 1200.00 from Ali's account and increment of Rs. 1200.00 in Feroze's account are done. _____ constitute(s) a transaction.

a) First task

b) Both the tasks

c) None of the tasks

d) Second task.

7. START TRANSACTION statement

a) Updates the current database.

b) Rolls back the current transaction.

c) Commits the current transaction and starts a new transaction

d) Starts a new transaction only.

8. ROLLBACK statement

a) Cancels the entire transaction.

b) Deletes the rows added to the table.

c) Commits the current transaction and starts a new transaction

d) Rolls back all the insertions of rows made during the current transaction.

ANSWER THE FOLLOWING QUESTIONS

1. Define a Transaction.
2. Explain with the help of an example that why should a transaction be executed as a whole or it should be not executed at all.
3. Distinguish between COMMIT and ROLLBACK.
4. Write the purpose of inserting Savepoints in a transaction.
5. What happens when Autocommit is set on?
6. Write SQL statement to set Autocommit to off.
7. What does the ROLLBACK statement do?
8. How do you tell the system that a transaction is beginning?
9. Why do we generally execute a COMMIT statement before beginning a transaction?
10. Name two statements that can be used to end a transaction?
11. Does executing the COMMIT or ROLLBACK statement end the current transaction?
12. What happens to the current transaction if a DDL Statement is executed?

LAB EXERCISES

a) **Perform the following tasks:**

Start MySQL session .

Create a table named Student with columns RollNumber, Name and Marks.

Start a transaction and insert two rows to the Student table.

Verify the inserts by SELECT statement.

Commit the changes.

Start another transaction.

Delete a row that was recently inserted.

Verify that the row has been deleted.

Rollback the changes.

Verify that the delete has been cancelled.

b) A table named ITEM has the following contents:

```
+-------+--------------+----------+
| icode | iname        | iprice   |
+-------+--------------+----------+
|   101 | CHAIR        |  1500.00 |
|   102 | DINING TABLE | 24000.00 |
+-------+--------------+----------+
```

Write the output that will be displayed by each SELECT statement as the SQL statements given below are executed:

```
mysql> SELECT * FROM ITEM;
mysql> SET AUTOCOMMIT = 0;
mysql> INSERT INTO ITEM VALUES(103,'COFFEE TABLE',340);
mysql> SELECT * FROM ITEM;
mysql> ROLLBACK;
mysql> SELECT * FROM ITEM;
mysql> START TRANSACTION;
mysql> UPDATE ITEM SET IPRICE = IPRICE +200;
mysql> SAVEPOINT S1;
mysql> UPDATE ITEM SET IPRICE = IPRICE +400;
mysql> SELECT * FROM ITEM;
mysql> ROLLBACK TO  S1;
mysql> SELECT * FROM ITEM;
```

Now verify the output by creating the table using MySQL and executing the above statements.

c) A table named Bill has the following rows:

```
+-----------+-----------+------------+-----------+
| Order_Num | cust_code | bill_date  | Bill_Amt - |
+-----------+-----------+------------+-----------+
|         1 | C101      | 2010-08-02 |     2300  |
|         2 | C105      |  2010-08-02|     5500  |
|         3 | C099      |  2010-08-23|     3000  |
|         4 | C165      |  2010-09-24|      6500 |
|         5 | C105      |  2010-09-24|      1400 |
+-----------+-----------+------------+-----------+
```

Write the output that will be displayed due to last SQL SELECT statement:

```
mysql> START TRANSACTION;
mysql> INSERT INTO BILLS VALUES(7,'C101','2010-09-02',5000);
mysql> UPDATE BILLS SET Bill_Amt = Bill_Amt+500 WHERE
Order_Num = 3;
mysql> SAVEPOINT A;
mysql> INSERT INTO BILLS VALUES(8,'C97','2010-09-03',4500);
mysql> DELETE FROM BILL WHERE cust_code = 'C105';
mysql> ROLLBACK TO A;
mysql> SELECT * FROM bills;
```

Now verify the output by actually executing the statements.

TIME BOUND TEAM BASED EXERCISE

(Team size recommended : 3 students each team)

Suppose the Government has nominated you as Head of Ministry of Information Technology. You have come out with the idea of a National Identity Card that should be

present with each citizen. The card will hold a limited amount of information but will be backed up by further information held in a database with the Government. Some information will be visible on the card, some encoded on chip in the card and other detailed information about each citizen will be present in tables in a database.

Think about the information that Government should have about each citizen and fill in the blanks. To give you track, some blanks are already filled up.

INFORMATION THAT WILL BE VISIBLE ON THE CARD:

- Name
- Date of Birth
- Photograph
- _____
- _____
- _____
- _____
- _____

INFORMATION THAT WILL BE ENCODED ON THE CHIP:

- Facial image
- Fingerprints
- _____
- _____
- _____
- _____
- _____

INFORMATION THAT WILL BE STORED IN THE DATABASE

- Signature
- biometric information
- _____
- _____
- _____
- _____
- _____

These days some people feel that everyone's DNA numeric profiles should also be available in database. It would really help to combat crime. However, a lot of people are worried about this idea; they think that it would have disadvantages.

Brainstorm and discuss the advantages and disadvantages of storing DNA profiles also in the database.

IT APPLICATIONS

Learning Objectives

After studying this lesson the students will be able to:

- Differentiate between front-end and back-end of an application.
- Identify various components of the front-end of an application.
- Design and develop simple IT applications.
- List the impacts of ICT on society

We have already seen in class XI that IT applications are essential requirement of every individual and organization to simplify their day-to-day work, efficiently manage and execute projects. These applications save time and efforts both. Now, it is the time to get into the real world of IT applications by first exploring the existing applications and then developing new applications to solve real life problems.

In class 11, you have already learnt about broad categories of IT application as e-Gaming, e-Business, e-Governance, e-Learning etc. e-Business involves applications dealing with buying and selling of products and services. e-Governance involves applications which are used by government agencies/organizations to provide better governance. e-Learning involves applications which are developed to help learning of any concept/skill. Similar applications are also possible in other sectors of economy and social service.

You must have used or seen others using many such applications several times. Whenever you perform an activity online, like register for a new email account, apply for a Visa while going abroad, reserve a seat on a flight/train, buy a book online - you are actually using IT applications only. So, you can see how these applications save us time and efforts in getting various jobs done. These applications have become an integral part of our modern society.

In this chapter let us learn about what goes into developing such applications. We shall also take up a few examples for better understanding.

Front-End Interface

All IT applications process some data entered by the user. For example, when an examinee has to see his result on the net, he has to enter his roll number. When a person has to deposit his house tax online, he has to enter information about his house and his credit/debit card using which the house tax has to be deposited. To place an order online for some purchase, the buyer has to enter some information about himself and the item to be purchased. Similarly for any IT application the user has to enter some data which may be just a number or a lot of data like buyer's details. Every IT application provides some sort of form using which users enter the data. This form is called the Front End Interface (or just Front-End or Interface or user-interface) of the application.

To create a front-end various components, like those studied in Java GUI application development, are used. Some of the most commonly used components are discussed below.

TextField: TextField is used to get small textual information like Name, RollNo, email address, quantity, etc. Disabled/Uneditable TextFields are also used to display such information.

TextArea: TextArea is used to get long textual information which may span multiple lines of text. E.g. to get Address, Complaint, Suggestion etc. Disabled/ Uneditable TextAreas are also used to display such information.

Radio Button: Radio buttons are used to get an option out of several mutually exclusive (out of which only one can be selected) options. Examples of such options are Gender (Male or Female or Other), Type of Credit Card (Master or Visa or Other), Type of internet connection (DialUp or Braodband), etc.

CheckBox: Check boxes are used to get one or more options out of several given options which are not mutually exclusive. These are the cases where multiple options are given to the user and the user can select zero or more out of the given options. Examples of such options are Hobbies (a user may have zero or more hobbies), Magazines to subscribe for (a user may subscribe to zero or more of the given magazines) etc.

List: A list is used to get one or more options out of several given options which may or may not be mutually exclusive. This may seem to be the case where CheckBoxes are to be used, but the difference is in the number of options available. If the number of options is small, then CheckBoxes can be used. In case of large number of options, using CheckBoxes may take up a lot of space on the form and it may also be inconvenient for the user to select the desired options. In such cases Lists are preferred over checkboxes. Examples of such cases are: To select cities out of a given list of cities, to select magazines out of a given list of magazines, etc.

ComboBox: A ComboBox is used to get an option out of several given options which are mutually exclusive. This may seem to be the case where RadioButtons are to be used, but the difference is in the number of options available. If the number of options is small, then RadioButtons can be used. In case of large number of options, using RadioButtons may take up a lot of space on the form and it may also be inconvenient for the user to select the desired option. In such cases ComboBoxes are preferred over radio buttons. Examples of such cases are: To select a city out of a given list of cities, to select a train out of a given list of trains, etc.

When the options are mutually exclusive, then a List can also be used instead of a ComboBox. It all depends on the space available on the form (a ComboBox consumes less space as compared to a List) and the look of the form (which the form designer has to decide).

PasswordField: A PasswordField is used to get some secret textual information like Password, CVV number of a credit card etc.

Front-end interface is the face of any application. In most of the cases, the front-end decides whether the application will be easily accepted or not. If the front-end is convenient and unambiguous for the user, then the user will like to use it and hence the application will be given positive reviews. If the front-end interface is inconvenient for the user, then the user will not like to use the application. Therefore, front-end of an

application must be user-friendly. Following are a few tips to make the front-end more and more user friendly:

1. **Consistency:** Consistency in looks and operations plays a major role in front-end design. If in one window the buttons are placed at the bottom, then in all the other windows also they should be placed at the bottom. If double-clicking an item pops-up a short-cut menu, then double-clicking any other item should pop-up the relevant short-cut menu. Labels, color-scheme etc. should also be consistent through-out the application. Consistency enables users to make an idea of how the application works, and this idea leads to fast acceptance of the application.
2. **Make it convenient for the user:**
   a) Place the most important items at the top-left position of the form. When a user looks at a window, top-left is the first position where user attention goes. So an item placed at top-left position has least chances of getting skipped.
   b) Don't use such bright colors which put pressure on users' eyes. The colors which look very fantastic are not necessarily convenient for the user when it comes to entering data or viewing reports.
3. **Help the user enter correct data in the first go:** Ask for minimum textual data to be entered by the user. If you have to ask for class and section, provide a list to choose the class, provide radio buttons to choose the section. This way user has the options only to enter the valid data. If you ask the user to enter the class and section in a text box, then the user has all the options to enter the data and hence more chances of entering invalid data.
4. **Listen to all:** Before creating the user interface, you should speak to the potential users and get their ideas to decide the design of user interface. You should put a limit there only. You must get the ideas but you are not bound to use these ideas. Use your skill and commonsense to decide which of these should be incorporated and which one should not be. The aim is to create a consistent, convenient, and logically correct user interface.
5. **Smooth shifting from one window to the next (or the previous):** Make the sequence of moving from one window to another exactly same as the flow of work the application is made to do.

Back-End Database

Front-End is just one part of an IT application. Any IT application usually stores a lot of data in the form of a database which is not visible to the user. This database is used by the application to give suitable responses to the user. This database is called Back-End Database (or just Back-End or Database). For Example, the database of train reservation system stores all the data about trains and passengers, the database of an online shopping system stores the data of all the items available in the store, and so on. If the front-end interface makes the user like or dislike the application in the first go, then the back-end decides whether the user will keep liking the application or not. A good back-end improves the speed of the application. A good back-end also ensures easy modification of the application whenever required. Following are the features of a good back-end database:

- It should use multiple tables for storing data to avoid data redundancy.
- Tables in the database should be created using constraints wherever applicable.
- Keys (Primary and Foreign) of tables must be defined.

To make the application efficient and effective, you should also follow the guidelines given below:

1. It should meet all the requirements of the problem, for which the application was created.
2. It should have user-friendly interface to make the user comfortable while using.
3. Code should have sufficient number of comments to help the programmer/yourself to modify/update the code in future.
4. Keep the navigation of input in a standard order as much as possible. Most significant information should be entered first.
5. There should not be any ambiguity in data and information and it should avoid inputting duplicate information anywhere in any form.

Front-End and Database Connectivity

A database application consists of Front-End and Database (Back-end). These two entities cannot work in isolation. Whatever data is entered by the user has to go to the database

and whatever relevant data is extracted from the database is to be shown to the user through the Front-End. Therefore, the Front-End and the Database of an IT application must be connected. This connectivity is achieved as learnt in Chapter 6 (Database Connectivity). If the application is web based then the connectivity is achieved using some scripting language (like vbScript or JavaScript).

Examples of IT Applications

There are numerous IT applications. We consider herein IT applications for e-Governance, e-Business, and e-Learning. Web addresses of a few of these are given below:

e-Governance:

To reach the citizens in an effective and transparent manner ICT enabled counters have been setup by government where several services like Birth/Death certificate registration, Railway enquiry and ticket booking, submission of RTI application etc. are provided. These centres are accessible to anyone and people can use these to get guidance, information, and services without paying any money to touts or middle men.

1. **india.gov.in (The National Portal of India) -** This portal not only gives the information about Government of India, but also allows the users to apply online for various services provided by the government.

2. **goidirectory.nic.in (Government of India Web Directory) -** Through this portal one can access various government web sites. These sites include sites of various states and union territories, and sites of central government departments etc. All these sites are examples of e-Governance applications of IT. Some of these sites are:

   a) **`mcchandigarh.gov.in:`**

      Portal of Municipal Corporation of Chandigarh

   b) **`Jammukashnir.nic.in:`**

      Portal of Municipal Government of Jammu and Kashmir

Know More!

Bhoomi (meaning land) is the project of on-line delivery and management of land records in Karnataka. It provides transparency in land records management with better citizen services and takes discretion away from civil servants at operating levels.

The Revenue Department in Karnataka, with the technical assistance from National Informatics Centre (NIC), Bangalore, has built and operationalised the BHOOMI system throughout the state. The BHOOMI has computerized 20 million records of land ownership of 6.7 million farmers in the state.

Know More!

National Informatics Centre (NIC) is a premiere Science & Technology institution of the Government of India, established in 1976, for providing e-Government / e- Governance Solutions adopting best practices, integrated services and global solutions in Government Sector.

Hindi version of Government of India portal is http://bharat.gov.in/. The image of home page of this website is shown below:

e-Business:

To reach the customers and business associates in an effective and fast manner business houses (now a days many small shops like snacks corners and paan shops also) provide their services on the net. These ICT enabled counters are used to get orders and feedbacks from the customers and also for inter-business transactions. This helps the businesses to widen their customer base.

1. **nafed-india.com/ebusiness.asp (e-business site of NAFED)** - Through this URL NAFD (National Agricultural Cooperative Marketing Federation of India Ltd.) offers its e-business services to various corporates and customers.
2. **Amazon.com (e-Business site of Amazon.com)** - Amazon is the world's largest online store. Through this URL Amazon does its online business

e-Learning :

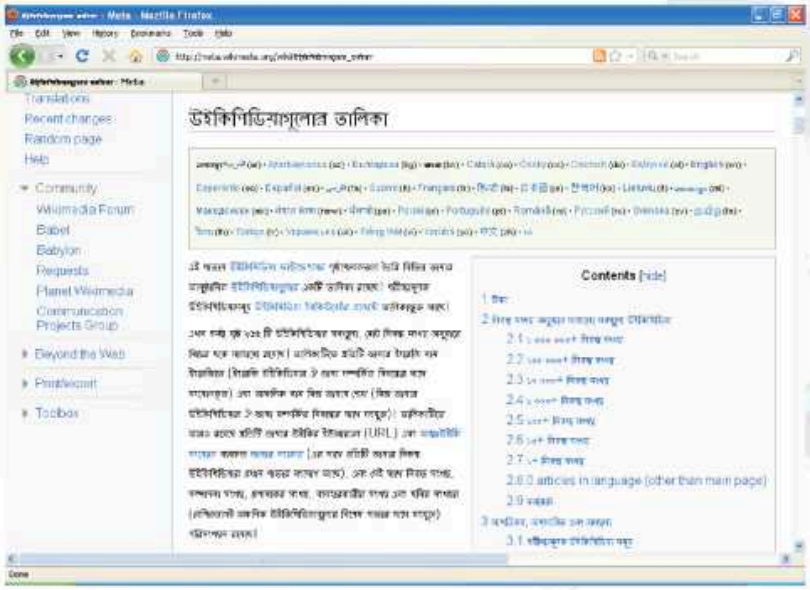
e-Learning has multiple goals. It is much more than having a net connection and/or CDs through which people learn. E-Learning is about giving freedom to people to learn whatever they want to learn and whenever they want to learn. This is irrespective of (except in exceptional cases) age, caste, gender, economical background, or qualification of the learner. The only requirement is the will to learn. E-learning is available on almost all the topics imaginable.

1. **w3schools.com (Website Developers e-Learning site) -** At w3schools.com you will learn how to make a website. It offers free tutorials in all web development technologies.
2. **www.gcflearnfree.org -** It is an educational part of the GCF mission. GCF creates and provides quality, innovative online learning opportunities to anyone who wants to improve the technology, literacy, and math skills necessary for them to be successful in both work and life. GCF believes that there's freedom in the ability to learn what you want, when you want, regardless of your circumstances.
3. **educationportal.mp.gov.in/public/multimedia.aspx -** This government of Madhya Pradesh portal provides multimedia tutorials on various topics of different subjects like maths, science, social sciences etc.
4. **ncert.nic.in/html/learning_basket.htm** - This NCERT portal provides interactive modules for students to learn various topics.

Websites in Indian languages

These days multiple Government and private organizations are providing their websites in Hindi and other regional languages also. The aim is to provide their services even to the common people in remote areas. Small towns where computers and internet have reached, information on the net should also be available in regional languages so that people not knowing English can also have access to the information. Language should not be a hinderance but a support to learning. Understaning the importance of regional languages, many websites have also provided translation services so that the same page can be viewed in any language of user's choice. Following are the home pages of a few websites in Hindi and other regional languages:


math.info
स्वागत !
FLASHGAMES247.COM
3D SUDOKU


উইকিপিডিয়াগুলোর তালিকা
Contents [hide]


ਵੈਬ
ਤਸਵੀਰਾਂ
Realtime
ਗਰੁੱਪ
Google
ਅੰਮ੍ਰਿਤਸਰ
ਤੇ ਲੱਭ
ਵੈਬ
ਅੰਮ੍ਰਿਤਸਰ ਲਈ 41,200 ਵਿੱਚੋਂ 1 - 10 ਨਤੀਜੇ (0.24 ਸੈਕੰਡ)
ਹੁਕਮਨਾਮਾ ਦਰਬਾਰ ਸਾਹਿਬ ਸ੍ਰੀ ਅੰਮ੍ਰਿਤਸਰ - DC Forum
ਅੰਮ੍ਰਿਤਸਰ: ਬਲੂਸਟਾਰ ਦੇ ਵਿਰੋਧ ਵਿਚ ...
ਖੂਨ ਚ ਡੁੱਬ ਡੁੱਬ ਰੰਗਿ: ਧਰਤਿ ਅੰਮ੍ਰਿਤਸਰ ਦੀ - Saanj
ਅੰਮ੍ਰਿਤਸਰ 'ਚ ਭਾਰੀ ਮੀਂਹ - Punjabi News


This page is in Hindi
Would you like to translate it?
Translate
Nope
Options
ताजमहल
वास्तु कला

Know More!

Guidelines for Multilingual websites:

NIC has developed guidelines for Indian Government websites. These guidelines are accessible at http://www.pon.nic.in/homeinfo/govt-website-guidelines.pdf. Article 5.7 of this document lays guidelines for Multilingual versions of Government websites. The main points of this article are:

a) Ideally all the pages on the website should be translated in Hindi and other regional languages. In case it becomes difficult to do so, corresponding Departments should identify the content which is widely accessed by the public and put up such content in regional languages.

b) It MUST be ensured that the documents/pages in multiple languages are updated simultaneously so that there are no inconsistencies, at any point, between the various language versions.

c) In case it is practically difficult to update the versions in all the languages simultaneously due to delays on account of translation etc., the obsolete information should be removed from the site till the latest information is uploaded. In any case, a time stamp indicating the date of uploading the information and its validity should be put along with all the time sensitive documents.

Development of IT applications

After going through these various IT Application sites, you must have realied that all these applications simplify the processes. Now, let us find out how these applications are created.

For developing such application, one needs to follow the following steps:

Step 1: Identify the problem for which the application is to be developed and discuss about its feasibility. If the applications is technically and economically feasible (possible and profitable to carry out), then steps are taken for its development, otherwise the project is scrapped.

Step 2: Identify and decide, which database tables and table structures will be required in the application. Make sure that the data types and sizes of the columns in the tables are carefully planned and used. Create database and tables as per the requirement of the application.

Step 3: Identify and decide, which all inputs are required to be taken from the user in the Front-End of the application. Find out, where you can minimize the typing efforts of user by introducing known options using RadioButton/CheckBox/ List/ComboBox etc. Develop the front-end of the application as per the requirement and ease of use.

Step 4: Establish the data connectivity between the Front-End interface and Back-End Database.

Step 5: Test the full application (Front-End and Back-End) with multiple sample sets of data. It is always better if the sample data are collected from potential users of the application randomly.

Now, the application is ready for implementation.

Examples: We shall take 3 examples, one from each of the categories: e-Business, e-Governance, and e-Learning, to better understand the process of IT application development.

Example 1 - e-Business :

Let us see how can we develop the application which we described as 2nd Team based exercise in chapter 9. To recall it, it is reproduced below:

To expand its business, XYZ Mall plans to go online. Anyone who shops at the Mall will be given a membership number and Password which can be used for online shopping. With this membership number and password, customers can place their orders online. The mall will maintain the customers' data and orders' data. A person is put on duty to keep constantly checking the Orders data. Whenever an order is received, its processing has to start at the earliest possible.

The Orders' data will be analysed periodically (monthly, quarterly, annually - whatever is suitable) to further improve business and customer satisfaction.

The application development steps are given below:

Step 1: The problem is identified. Its economic and technical feasibility are discussed. It is found that the application development and implementation are technically and economically feasible. So the management gives a green signal for application development.

Step 2: After discussing with the management and the potential users of the application, it is decided to create the following tables in the database:

Table: Customers

S.No.	Field	Datatype	Remarks
1.	Membership_number	Int(9)	Primary key. First 6 digits will identify the membership date and the last 3 digits the serial number on that date.
2.	Password	Varchar(15)	-
3.	Name	Varchar(25)	Customer's Name
4.	Address	Varchar(50)	Customer's address.
5.	Phone	Varchar(30)	Customer's phone number(s).

Table: Items

S.No.	Field	Datatype	Remarks
1.	Item_Code	Int(5)	Primary key. First 2 digits will identify the category of the item, and the last 3 digits the serial number in that category.
2.	Description	Varchar(30)	Item's Name
3.	Price	Decimal(7,2)	Item's Price
4.	Discount	Decimal(5,2)	Percentage Discount allowed on the item.

Table: Orders

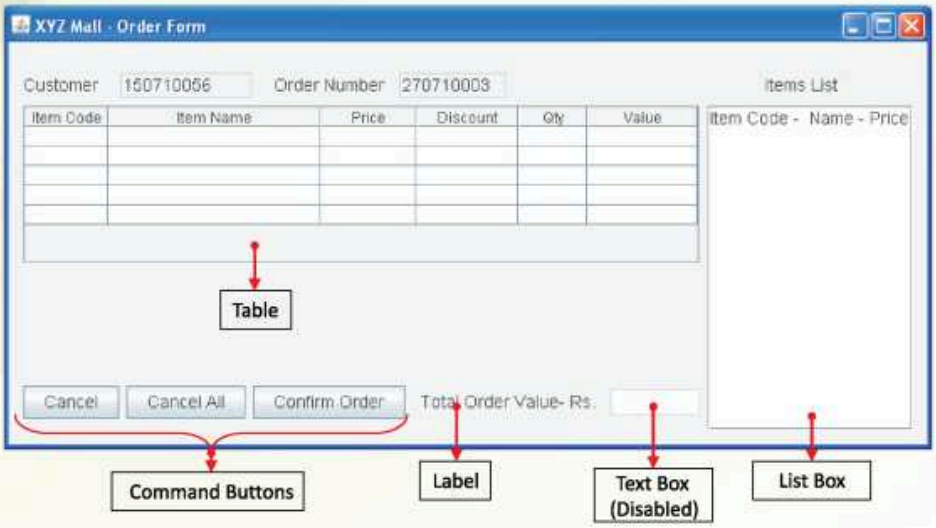
S.No.	Field	Datatype	Remarks
1.	Ord_number	Int(5)	Order Number.
2.	membership_number	Int(9)	Membership number of the customer who has placed this order.
3.	Ord_date	Date	Date of order placement.
4.	Item_Code	Int(5)	Code of the item ordered.
5.	Qty	Decimal(5,1)	Quantity of the item ordered.
6.	Status	Varchar(10)	Current status of this item. Status can be: Waiting/Completed/Cancelled.

Step 3: After careful analysis, it is found that the inputs required from the user are as follows:

- Membership_number
- Password
- Item_codes of the items to be ordered
- Quantities of the items ordered

All the other fields like order_number, item_name, Order_date, Discount etc. will be auto-generated/calculated by the application.

To get input from customers, following front-end interface can be designed:



Step 4: Data connectivity between the Front-end interface and back-end database is established using some scripting language like Javascript or VBScript etc.

Step 5: After development the application is tested by giving membership numbers and passwords to the employees of the mall itself so that the application can be tested with multiple sets of random data.

If any discrepancies are found during test, the corresponding corrections are made in the application. After complete testing, the application is implemented and the customers are given membership numbers and passwords.

Example 2 - e-Governance :

The state administration wants to make vehicles' data (RegNo, RegDate, Owner, OwnerShipNumber, Address, HP) easily available to citizens. Each registration authority in-charge will regularly update the data. Citizens will be given *read only* access to this data.

The application development steps are given below:

Step 1: The problem is identified. Its economic and technical feasibility are discussed. It is found that the application development and implementation are technically and economically feasible. So the corresponding authority (may be the State Transport Authority) gives a green signal for application development.

Step 2: After discussion with the management and the potential users of the application, it is decided to keep the following tables in the database:

Table: Vehicle

S.No.	Field	Datatype	Remarks
1.	RegNo	Char(12)	Primary key. Resgistration Number is 10 characters long. To make it future proof, a provision of 12 characters is made.
2.	RegDate	Date	Registration date of the vehicle.
3.	VehicleType	Varchar(10)	Car/Truck/Bus/Scooter/MCycle etc.
4.	Manufacturer	Varchar(20)	Vehicle Manufacturer's Name
5.	Model	Varchar(20)	Name of the Model

Table: Owner

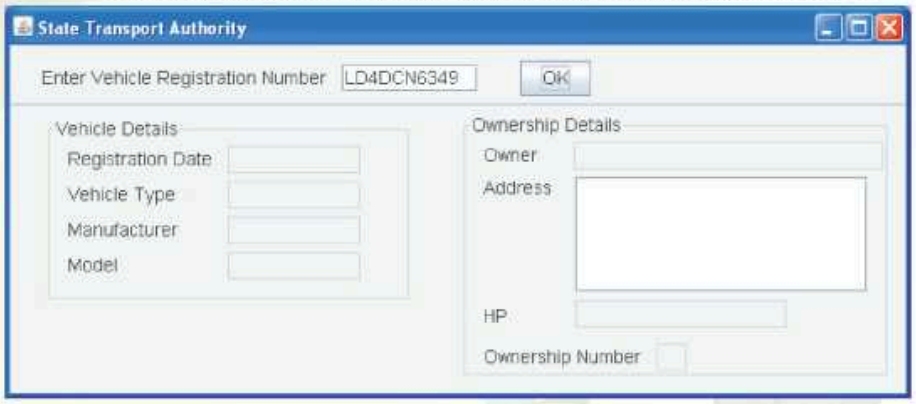
S.No.	Field	Datatype	Remarks
1.	RegNo	Char(12)	Primary Key.
3.	Owner	Varchar(25)	Owner's Name
4.	Address	Varchar(50)	Owner's address.
5.	OwnerShipNo	Int(2)	Ownership Number of the vehicle.
6.	HP	Varchar(25)	Name of the bank/institution which has financed the vehicle.

Step 3: After careful analysis, it is found that the inputs required from the user are as follows:

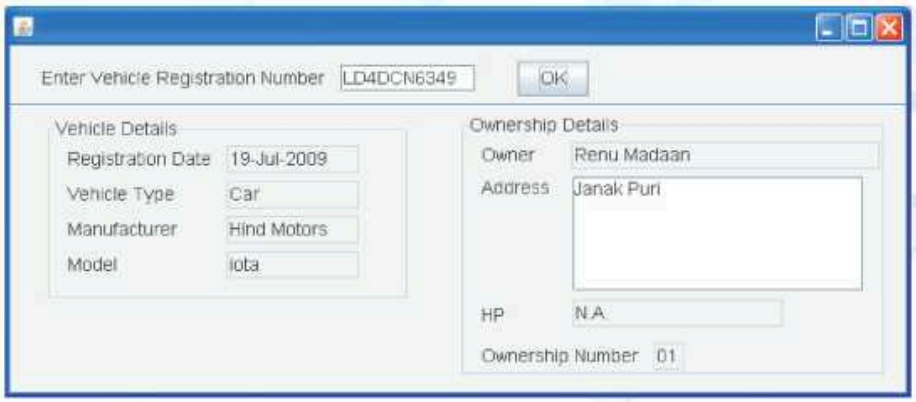
- Registration Number

All the other fields like Owner's Name and Address, HP etc. will be provided by the application from the database.

To get input from the users, following front-end interface can be designed:



Once the registration number is input, database can be searched for it and the corresponding details of the vehicle and the owner can be shown to the user in the following format:

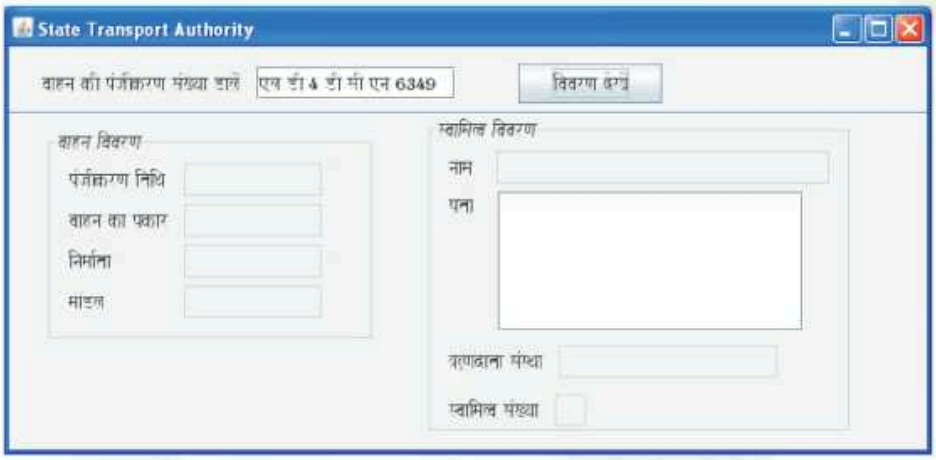


Step 4: Data connectivity between the Front-end interface and back-end database can be established using some scripting language like Javascript or VBScript etc.

Step 5: After development the application can be tested by entering various registration numbers.

If any discrepancies are found during test, the corresponding corrections can be made in the application. After complete testing, the application can be implemented.

Hindi version of this IT application can look like as follows:



Example 3 - e-Learning :

An organization of dedicated teachers, 'Meticulous Teachers Consortium', decides to invite computer aided teaching modules from individuals and organizations so that these can be put on the internet for students' use free of cost. No money will be charged from users and no money will be paid to the developers. Once the modules start pouring in, a front-end is created for the students where the students can select any of the available modules to learn any topic.

The application development steps are given below:

Step 1: The aim is identified. Its economic and technical feasibility are discussed. It is found that the application development and implementation are technically and economically feasible. So it is decided to go ahead with the application development.

Step 2: To keep track of various modules contributed and used, the following database is created:

Table: Modules

S.No.	Field	Datatype	Remarks
1.	ModuleNo	Int(5)	Primary key. Serial Number allotted to the module.
2.	Subject	Varchar(20)	Subject name to which the module caters.
3.	Topic	Varchar(20)	Topic Name to which the module caters.
4.	Level	Varchar(10)	Beginner/Intermediate/Advanced.
5.	Duration	Int(4)	Duration of the module in seconds.
6.	Developer	Varchar(20)	Name of the module developer.
7.	SubmissionDate	Date	Date on which the module is submitted.
8.	NoOfHits	Int	Number of times the module is viewed by the users. Each time a user views the module, NoOfHits is incremented by 1.
9.	LastUsedDate	Date	Date on which the module was viewed most recently.

Step 3: After careful analysis, it is found that the inputs required from the user are as follows:

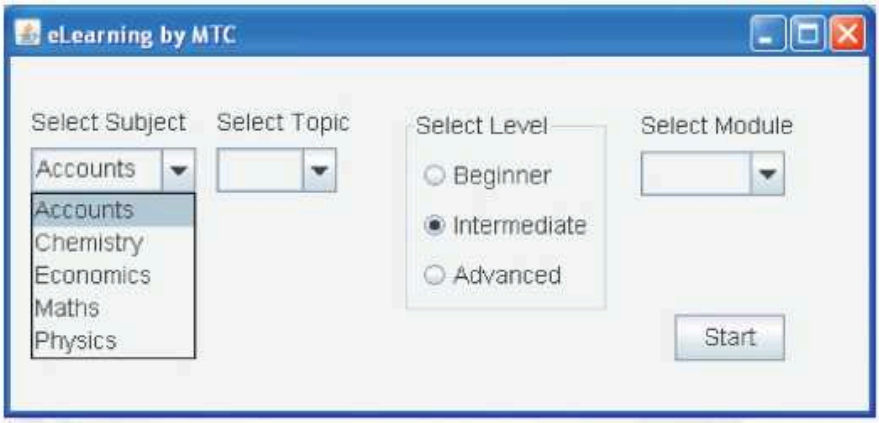
If a user has already viewed a module and he again wants to view the same module which he remembers by its module number, then the only input required is:

- Module Number

If a user wants to see a module for the first time or he does not remember the module number, then the input required is as follows:

- Subject (from a list of subjects)
- Topic (from a list of topics corresponding to the subject chosen)
- Level (Beginner / Intermediate / Advanced)
- Module Number (from a list of module numbers corresponding to the above 3 enteries)

To get input from a user, following front-end interface can be designed:



Once the desired input is obtained the corresponding module is executed.

Step 4: Database connectivity between the Front-end interface and back-end database can be established using some scripting language like Javascript or VBScript etc.

Step 5: After development the application can be tested by selecting various modules and running them.

If any discrepancies are found during test, the corresponding corrections can be made in the application. After complete testing, the application can be implemented.

Impact of ICT on society

Like everything else that is used by common man, ICT (Information and Communication Technology) also has impacted the society. ICT has impacted the society in a much wider way than any other technology. Most of these impacts are positive, though there are some negative impacts also.

Social and Economic benefits of ICT:

- Social networking sites help people remain in touch with their nears and dears even when they are staying on opposite sides of the globe.
- Social networking sites help like minded people come together and work for some cause.
- e-Governance sites help people save their productive time by performing various government related jobs like getting some forms, depositing bills online.
- ICT helps economy grow at a faster rate as it provides transparency in the processes and helps the government to keep check on defaulters.
- Due to e-Banking and use of plastic money more money is put in circulation leading to faster growth of GDP.
- e-Learning sites make quality study material available even to the students staying at remote places.

Infomania:

Infomania is the condition of reduced concentration caused by continually responding to electronic communications such as e-mail, SMSs, MMSs etc. ICT is making more and more people infomaniac. This is making some people waste their productive time in the office, neglect their families and duties. Some people are also in a habit of frequently checking their e-mails even when they are on vacation with their families. We have to be careful in the use of ICT so that we use it constructively and not get obsessed with it and become infomaniacs.

Summary

- Three major groups of IT applications covered in this chapter are: **e-governance, e-business, and e-learning.**
- **e-Governance** involves applications which are used by government agencies/organizations to provide better governance.
- **e-Business** applications use technology to effectively access and deliver business related services and perform various kinds of business transactions.
- **e-Learning** applications use technology to effectively deliver and monitor learning and teaching processes. They help the trainer to organize and manage his/her lesson plans, present them to students/learners, evaluate and take the feedback to enhance & fine-tune this process in future.
- An IT application has two major parts: Front-end (The user interface) and **back-end** (The database)
- The **front-end** of an IT application is usually a group of one or more forms through which the user enters the input values and is shown the corresponding output. A good front-end ensures the acceptance of the application in the first go.
- The **back-end** of an IT application is the database in which all the data is stored. This database resides in the server. All the data which is requested by the front-end is supplied by back-end. A good back-end ensures sustainability, efficiency and easy modification of the application.
- Development of an IT application involves creation of front-end, back-end, and connecting these two. It also involves testing the application and then implementing it.
- Use of ICT has its social and economic impacts. Society is impacted as due to ICT people change their way of conducting the transactions and thus save their time, money, and energy. Economy is impacted as ICT leads to fast completion of data transfer and data processing jobs. ICT also brings transparency in the administration.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. A web site to provide online information and services to the citizens is an example of
   a. e-Business b. e-Mail
   c. e-Governance d. e-Learning
2. The web-site of an electricity supply company which allows its customers to pay bills online is an example of
   a. e-Business b. e-Mail
   c. e-Governance d. e-Learning
3. The web-site of a school which allows the students to go through various lessons in their subjects is an example of
   a. e-Business b. e-Mail
   c. e-Governance d. e-Learning
4. Web address of national portal of India is:
   a. India.gov.in b. GOI.gov.in
   c. ncert.nic.in d. None of the above
5. A form through which users interact with an IT application is a part of
   a. database b. front-end
   c. back-end d. Javascript
6. A good front-end is
   a. consistent b. user-friendly
   c. neither of the above d. both a and b.
7. Mr. X is an infomaniac. It means he
   a. Uses information carefully
   b. Uses computers to get information.

c. Responds to almost all his SMSs, eMails etc.

d. Tries to get correct information

8. Javascript is a

a. database b. front end

c. back-end d. scripting language

ANSWER THE FOLLOWING QUESTIONS

1. What is the advantage of using IT applications over the manual operations?
2. Write two important features of each of the following types of applications?
   a. e-Governance
   b. e-Business
   c. e-Learning
3. Give some examples of input values, where Radio Button and Check Boxes should be used for efficiency in the application.
4. What are the important guidelines we should keep in mind while developing an efficient application?
5. Is it a good practice to take in the inputs using TextFields only? Justify your answer.

TEAM BASED TIME BOUND EXERCISE:

(Team size recommended: 3 students each team)

1. **Online Applications for New Water Connection:** Municiple Corporation of Indirapuram has decided to develop an IT application so that citizens can apply online for new water connections. A rough idea of how the application will function is given below:

   Application forms for new water connection will be made available online. These forms can be downloaded from the net, filled in, and submitted online only.

   When an application is received, the server automatically puts it in the table of corresponding zone. Zonal files are accessible by respective zonal heads.

It is responsibility of the zonal head to ensure that an application from his/her zone is processed within a stipulated time period. Whenever a new connection is granted or rejected, the ResponseDate, and Remarks columns of corresponding row are updated by the respective zonal head. For this purpose, one of the tables of the database is given below:

S.No.	Field	Datatype	Remarks
1.	App_Number	Int(9)	Primary key. This is the application number of the received application. App_number is auto-generated by the application. First 6 digits will identify the date of application and the last 3 digits, the serial number of the application received on that date.
2.	NameOfApplicant	Varchar(30)	Name of the applicant
3.	Address	Varchar(50)	Address of the applicant
4.	AppDate	Date	Date of application
5.	Conn_Type	Varchar(10)	Type of connection required. It can be domestic/commercial/charity.
6.	Zone	Varchar(10)	Zone of the applicant's residence. It can be East, West, South, North, Centre.
7.	ResponseDate	Date	Date when the action on the application is started by the authorities.
8.	Remarks	Varchar(30)	Remarks to be entered by the corresponding zonal head. It can be In Progress / Rejected / Completed /Incorrect Address.

The job of each team is to describe the remaining tables of the database and to suggest the design of front-end.

2. **Birth Certificate:** Babu Lal lives in a small village. He is blessed with a girl at home only. The village teacher visits his house and congratulates him. She tells him to register the birth of his daughter with the government, and shares the benefits of having the Birth Certificate. Birth certificate will help Babu Lal's daughter name inclusion in ration card, at the time of admission to school, to avail facilities like Ladli etc. The information base is used for forecasting and planning activities related to healthcare, Welfare measures etc. in an effective way. Babu Lal is happy and immediately goes to the centre for his daughter's Birth certificate generation.

   Your job is to develop an application which generates birth certificates. The database for this purpose is as follows:

S.No.	Field	Datatype	Remarks
1.	BirthID	Int	This is th primary key of the table.
2.	Gender	Char(1)	'M', 'F', or 'O'
3.	DateOfBirth	Date	
4.	TimeOfBirth	Varchar(10)	Time of birth in hh:mm:ss format according to 24hr clock.
5.	PlaceOfBirth	Varchar(30)	
6.	FatherName	Varchar(30)	
7.	MotherName	Varchar(30)	
8.	ResAddress	Varchar(30)	
9.	IssuingAuthority	Varchar(30)	

The birth certificate can be of the form:

Name / नाम :	_____
Gender / लिंग :	_____
Date of Birth / जन्म की तारीख :	_____
Time of Birth / जन्म का समय :	_____
Place of Birth/ जन्म स्थान :	_____
Hospital Name/ अस्पताल का नाम :	_____
Delivery Type / प्रसूति का प्रकार :	_____
Father's Name / पिता का नाम :	_____
Mother's Name / माता का नाम :	_____
Residential Address / घर का पता :	_____
Issuing Authority / जारी करने वाला अधिकारी :	_____

3. **Sales Agency (B2B):** A sales agency gets FMCG from manufacturers and sells these to Whole sellers. For this the agency has a Suppliers table, a Goods table, and a Customers table.

   Customers can place their orders online. The details of orders accepted are placed in an Orders table.

   The Goods table is regularly updated. Orders table is checked twice a day for any new orders.

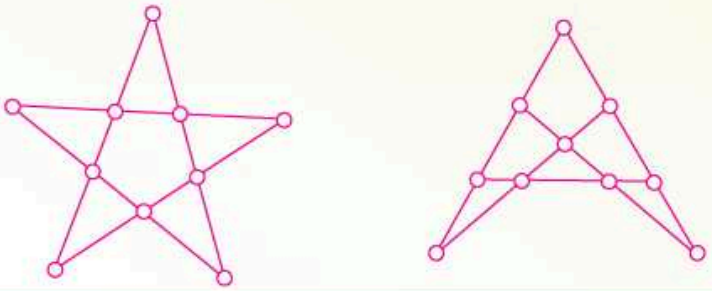
   The job of each team is to describe the attributes with data-types to be placed in each table and and to suggest the design of front-end.

APPENDIX - I

The teacher should divide the class in teams of 3-4 students each and challenge teams to solve the given puzzle. The teams that solve the puzzle within 3 minutes, 4 minutes, and 5 minutes are awarded the title of 'Fasttrack', 'Achiever', and 'Able' respectively.

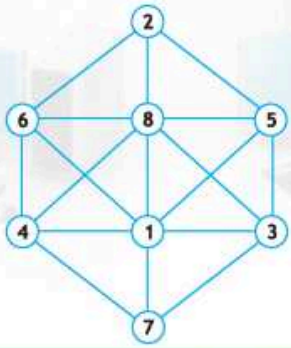
Solution1

There are two different ways to place ten coins satisfying the given conditions.



Solution 2

We are giving you one possible solution but there are many other solutions possible. So keep trying.....



Solution3

A List Of Palindromes (Phrases). Each line of the phrase is a palindrome.

Solution 4

The lady's children are aged 2, 2 and 9.

As per the first two conditions that she has 3 children and the product of their ages is 36(6 runs each in 6 balls = 6 x 6) the probable ages are:

(1,1,36),(1,2,18),(1,3,12),(1,4,9),(1,6,6),(2,3,6),(3,3,4) and (2,2,9).

The student knows her berth number but he is not able to reach the answer. That means the sum must be same for two of the pairs. This reduces the chances to two: { (2,2,9),(1,6,6) }.

From the third option, we get a constraint that, there is only one elder child. And we reach the correct answer of (2,2,9)

Solution 5

9-digit number is: 473816952

First rounding (down) yields: 473816950	(as 2 rounds down to zero, 5 stays the same)
2nd rounding (up) yields : 473817000	(5 rounds up to zero, add the carried over one to 9 to get 0, add the carried over one to 6 to get 7)
3rd rounding (down) yields: 473817000	(0 rounds down to zero, 7 stays a seven)
4th rounding (up): 473820000	(7 rounds up to zero, 1 adds one and becomes 2)

At this point sum of all the not rounded digits is 2+8+3+7+4 =24

5th rounding (down) yields: 473800000	(2 rounds down to zero, 8 stays a eight)
6th rounding (up) yields: 47400000	(8 rounds up to 0, 3 adds one and becomes 4)
7th rounding (down) yields: 47000000	(4 rounds down to 0, 7 stays a seven)
8th rounding (up) yields: 50000000	(7 rounds up to 0, 4 adds one and becomes 5)

Solution 6

One paise = 1/100 of a rupee.

Given the conditions, we know that:

(100*X + Y)*2 = Y*100 + X - 20

200X + 2Y = 100Y + X - 20

199X - 98Y = -20

We also know that X and Y must both be integers.

So, putting the equation in Y = MX + B form:

199X + 20 = 98Y

(199X + 20)/98 = Y

Now put this in the Y= function of a graphing calculator, set the table to start at 1 and go up by increments of 1 and then scroll down the table until you see an integer pair.

The integer pair occurs at (26,53)

Therefore, X = 26 and Y = 53

Solution 7

1. Mark the jars with numbers 1, 2, 3, 4, and 5.
2. Take 1 pill from jar 1, take 2 pills from jar 2, take 3 pills from jar 3, take 4 pills from jar 4 and take 5 pills from jar 5.
3. Put all of them on the scale at once and take the measurement.
4. Now, subtract the measurement from 150 (1*10 + 2*10 + 3*10 + 4*10 + 5*10)
5. The result will give you the jar number which has contaminated pill.

Solution 8

The numbers of the houses on each side will add up alike if the number of houses be 1 and there are no other houses except David's house, or if David's house number be 6 with 8 houses in all, or if David's house number be 35 with 49 houses, or if David's house number be 204 with 288 houses, and so on. But we know that there are more than 30 and less than 50 houses, so we are limited to a single case.

Mr. David's house number must be 35.

Solution 9

Required : 50m + 100n = 2500 ---------- 1

By mistake, Vijaya was given : 100m + 50 n

50m + 100n - (100m+50n) = 500

50m + 100n - 100m -50n = 500

50n - 50m = 500 ---------- 2

Adding 1 and 2

150n = 3000

n = 20 ---------- 3

Substituting value of n in Equation 2.

50 x 20 -50 m = 500

1000-50m = 500

-50m = -500

m = 10

Therefore Vijaya asked for 10 notes of Rs.50.00 and 20 notes of Rs. 100.00

APPENDIX - II

Indic Language Support

Mac OS 10.5 supports Devanagari, Gujarati, Gurmukhi and Tamil

Linux based desktops support Bengali, Devnagari, Gujarati, Kannada, Malayalam, Oriya, Tamil, Telugu and Gurmukhi,

Creating Documents in Multiple Languages in BOSS, GNU/Linux

There are various Input Methods available in BOSS, GNU/Linux. In these operating systems you can select the language of your choice in which you want to create your document. For example to choose Hindi:

Within each language there are multiple keyboard layout options are available. You can choose keyboard layout of your choice.

- Three different type of keyboards are supported
  - i) **Phonetic Keyboard:** प्रक्रिया is typed as 'p r a k r i y a' through English phonetic keyboard.
  - ii) **INSCRIPT Keyboard :** Keyboards are used which contains Indian alphabets as the key of this keyboard. So by typing those keys the content of that language can be written. (REF: www.tdil.mit.gov.in/isciichart.pdf)
  - iii) **Remington Keyboard:** Keyboard also contains keys of the Indian languages and the arrangement of the keys follows the arrangement of a typewriter.

Enabling Indic Language Support in Windows

Windows 7 and Windows Vista include all the necessary files to support Indic languages i.e. Complex(Indic) text support is automatically enabled. Therefore you just need to enable the keyboard for the language that you want to use by following the steps in the Enable a keyboard layout section.

For Windows XP, some additional setup may be required to support Indic languages. Therefore you first follow the steps given under Enabling International Language Support in Windows and then proceed with the steps given under the Enable a keyboard layout section.

Enabling International Language Support in Windows :

Step 1: Click **Start** and then go to **Control Panel.**

Step 2: Click on **Date, Time, Language, and Regional Options** and choose **Add Other Languages** from the task list.

Step 3: In the **Regional and Language Options** dialog box Highlight the **Languages** tab.

Step 4: In the **Regional and Language Options** dialog box, under **Supplemental Language Support,** select the **Install files for complex script and right-to-left languages** check box. Click OK or Apply.

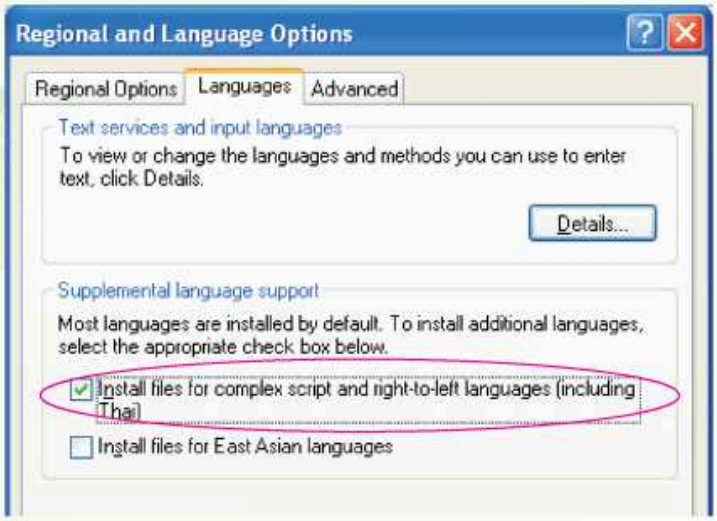


Figure 1 Languages tab in Regional and languages option in Windows XP

Step 5: You will be prompted to insert the Windows CD-ROM or point to a network location where the files are located. After the files are installed, you must restart your computer.

Enable a Keyboard Layout in Windows :

Step 1: Under "Text services and input languages," click on the "Details..." button.

Step 2: Under Installed Services, click "Add..."

Step 3: In the **Text Services and Input Languages** dialog box, on the **Settings** tab, click **Add.**

Step 4: In the **Add Input Language** dialog box, click the **Input language** list and select your preferred language and dialect. If you want to change the standard keyboard layout, click the **Keyboard layout/IME** list and select a new keyboard layout. Then, click **OK.**

Step 5: In the **Text Services and Input Languages** dialog box, on the **Settings** tab, click the **Default input language** list, and select the language you will use most often. The language you select as the default will display when you first start your computer. If you have finished adding languages, click **OK.**

Step 6: Click the **Regional Options** tab. Click the **Standards and formats** list, and then select your region.

Step 7: Click the **Location list,** and then select your location.

Step 8: Once done, click OK to exit. On the Text Services and Input Languages page, click OK again to close Regional Options. You should now see a language indicator in the System Tray (located at bottom right hand corner of the desktop by default)

Turning on the language bar :

If you do not see the language bar in the task bar (at the bottom of the desktop) or floating on the desktop please do the following:

Step 1: Click Start, click Control Panel, and then double-click Regional and Language Options.

Step 2: On the Languages tab, under Text services and input languages, click Details as shown in Figure 8.

Step 3: Under Preferences, click Language Bar.

Figure 2 Turning on the Language Bar in Windows XP

Step 4: Select the Show the Language bar on the desktop check box.

Figure 3 Language Bar Settings in Windows XP

Note: You can switch between different languages by clicking on the language bar and changing the language or by pressing the left ALT+SHIFT keys.

Using Microsoft Indic Language Input Tool ("ILIT") :

Do the following to use Microsoft ILIT in any application, such as Notepad.

Step 1: Open the application in which you would like to enter Indic text.

Step 2: Change the language using the language bar, which typically appears in the taskbar as follows.

Figure 4 - Language Bar in System Tray in Windows XP

Step 3: The language bar will now show the current language.

Step 4: You can now start typing in English and whatever you type automatically gets transliterated after a word-breaking character like a space, comma, etc. is entered. Note that this language setting is per application. You may have to repeat the steps above for each application you want to use Microsoft ILIT in. The Microsoft indic input tool comes with a virtual keyboard that can be used to type in notepad.

Figure 5 Virtual Hindi Keyboard

Note: You can switch between different languages by clicking on the language bar and changing the language or by pressing the left ALT+SHIFT keys.

Installing Fonts in Windows :

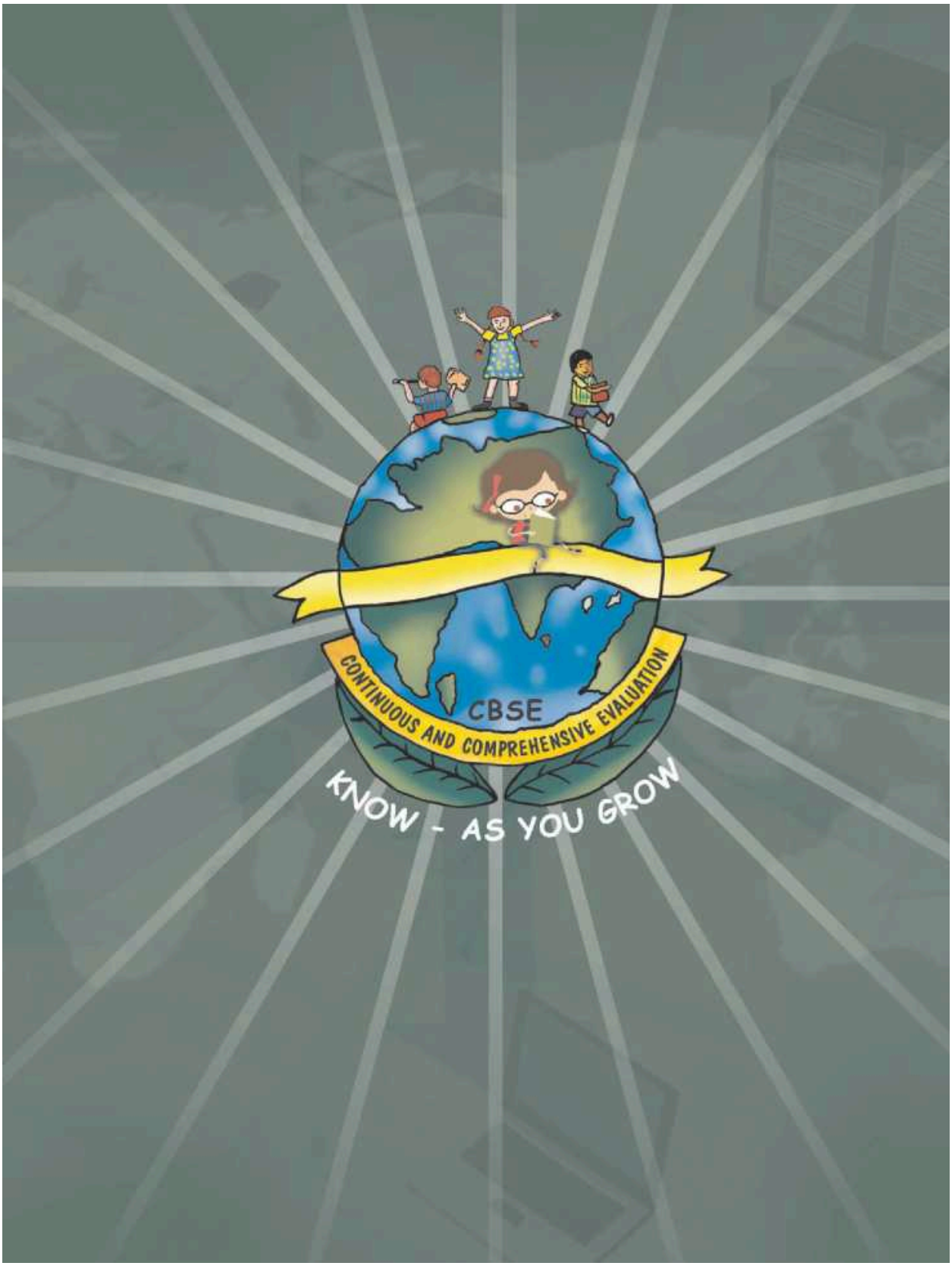
Step 1: Go to Windows Fonts folder e.g. C:\Windows\Fonts. (The path may differ on some computers.)

Step 2: Copy-paste the font file into this folder. Windows will now install the font file.

Step 3: Once installed the font will be available in your text-based applications.

Source:

1. http://www.microsoft.com/windowsxp/using/setup/winxp/yourlanguage.mspx
2. http://office.microsoft.com/en-us/support/enable-keyboard-layouts-for-different-languages-HA010354267.aspx


CBSE
CONTINUOUS AND COMPREHENSIVE EVALUATION
KNOW - AS YOU GROW

CENTRAL BOARD OF SECONDARY EDUCATION

Shiksha Kendra, 2, Community Centre, Preet Vihar, Delhi - 110092, India
Tel.: 91-11-22509252-59 Fax : 91-11-22515826
E-mail : cbsedli@nda.vsnl.net.in website : www.cbse.nic.in

CLASS XII

INFORMATICS PRACTICES

CENTRAL BOARD OF SECONDARY EDUCATION
Shiksha Kendra, 2, Community Centre, Preet Vihar, Delhi-110 092 India