

# Introduction

## 1.1 History of Web

The WWW stands for World Wide Web. The world wide web is a information system various documents containing information are interlinked together. User can access this information or write the information via computers. This information is typically stored on the web pages and through web browsers we can access these web pages. The web pages may contain the information in the form of text, audio, video, images and graphics. We can navigate between the web pages using hyperlinks.

- The concept of WWW was introduced by **Sir Tim Berners-Lee** the contractor at the **European Organization for Nuclear Research (CERN)**, Switzerland in 1980. He built a personal database of people and software models and used hypertext so that each new page of information was linked to an existing page.
- In 1990, Berners-Lee introduced the tools such as HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML) and the web browser. This is the time when the first website <http://info.cern.ch/> was built.
- During 1992-1995, along with HTTP protocol a new protocol named **Gopher protocol** which provided access to content through hypertext menus presented as a file system rather than through HTML files. In 1993 a new web browser with graphical user interface **Mosaic** got introduced.
- In 1994, the **World Wide Web Consortium (W3C)** was founded by Berners-Lee at the **Massachusetts Institute of Technology (MIT)** with support from the **Defense Advanced Research Projects Agency (DARPA)**. This organisation was built for creating standards and recommendations to improve the quality of the Web. Berners-Lee made the Web available freely, with no patents.. The World Wide Web Consortium (W3C) decided that their standards must be based on royalty-free technology, so they can be easily adopted by anyone. And then at the

end of 1994 a large number of websites got activated with popular web services.

- During 1996-1998, trade marketing started using WWW. The term E-commerce got introduced during this period only.
- During 1999-2000, many entrepreneurs started selling their ideas using the **dotcom boom**.
- From 2002- till date, the WWW has got an evolving nature due to various development such as online-booking, efficient search engines and agent based technologies, Facebook, social networking sites and so on.

## 1.2 What is Computer Network?

The computer network can be defined as a group of computers and other devices (such as printers, scanners, routers) connected together in order to send or receive data.

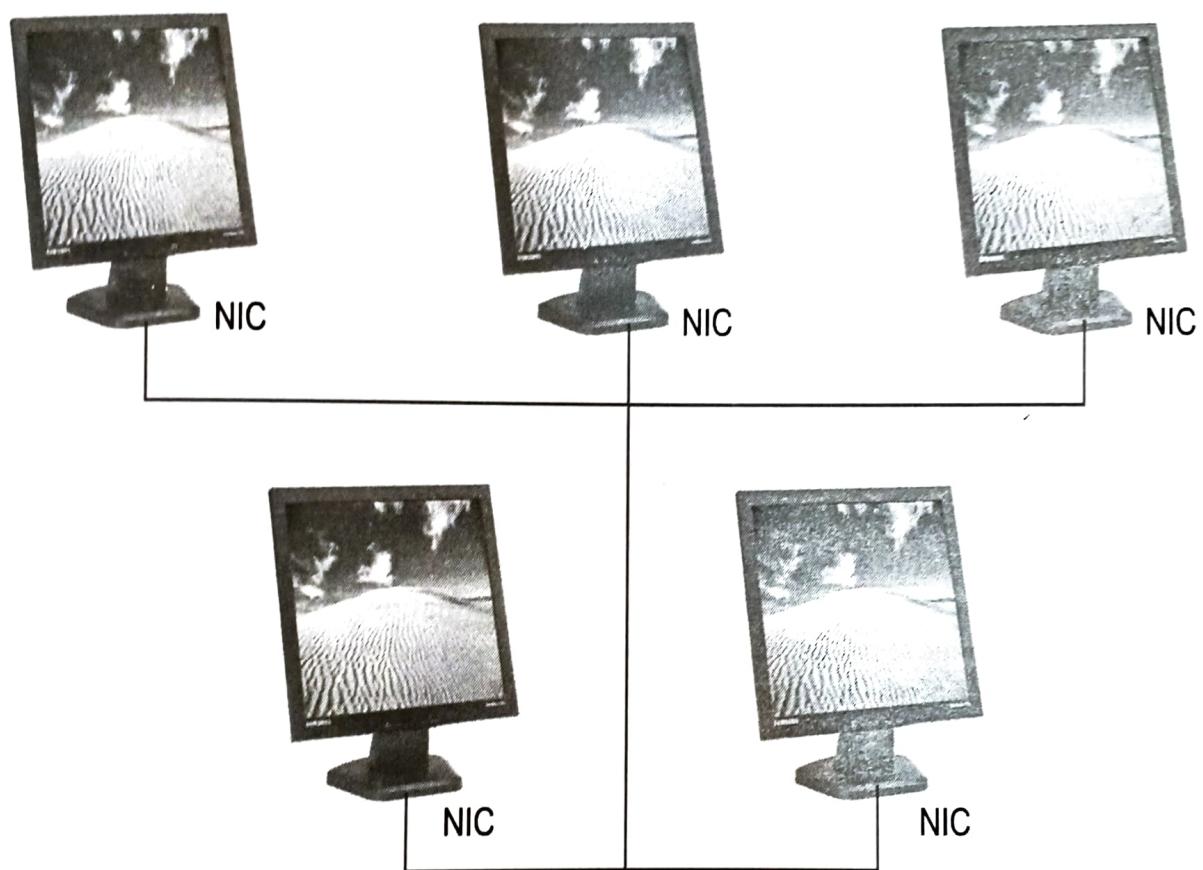
- Each of these devices in the computer network is called **node**. In computer network each node is connected by a wire through which the data can be moved in the form of electromagnetic waves. However there is another terminology which is commonly used in computer network and i.e. **host**. The host is synonym to the computer node, however there is one clear difference between node and host. Node refers to any device on the computer network and host refers to the fully functional general-purpose computer.
- A computer network comprises the following components:
  - A minimum of at least 2 computers
  - Cables that connect the computers to each other, although wireless communication is becoming more common
  - A network interface device on each computer. This is called a **network interface card** or NIC
  - A 'Switch' used to switch the data from one point to another.
  - Network operating system software
- You can create a computer network by hooking all the computers in your college together with cables and installing a special **network interface card** (an electronic circuit card that resides inside your computer popularly called as **NIC**) in each computer so you have a place to plug in the cable. Then you set up your computer's operating-system software to establish networking.
- Each node in the network has some unique address. These **addresses** help to identify the computer quickly. Addresses are assigned differently on different kinds of networks. These addresses are assigned by the

organizations that are setting up the corresponding network. Typically these addresses are numeric that is easy for computers to work with, but not for humans to remember.

Example: 192.168.0.165

Some networks also provide names that humans can more easily remember than numbers.

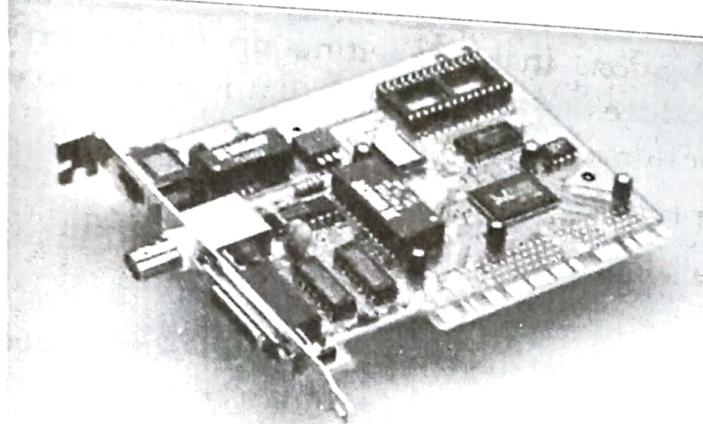
**Example:** [www.vtubooks.com](http://www.vtubooks.com) corresponding to some specific numeric address. The typical scenario of computer network is as shown below



**Fig. 1.1 Computer network**

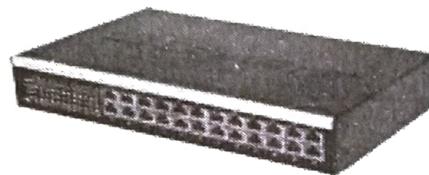
The network address is made of 4 bytes separated by dots for example 131.108.2.1  
First three bytes (either 1,2 or 3 ) correspond to network address and remaining three bytes (i.e. last 3,2 or 1) are used for host machine.

- In computer networking, a NIC provides the hardware interface between a computer and a network. A NIC basically is network adapter hardware. **Ethernet** supports data transfers at the rate of 10 Megabits per second (Mbps). To connect Ethernet cables to a computer, we normally use a network interface card (NIC).



**Fig. 1.2 Network Interface Card (NIC)**

- Switch is a device used to connect the PC to network. The switch is a device which is responsible for managing bandwidth on the network. In other words, switches are actually capable of switching the data from one port directly to the other without wasting the bandwidth of the entire network attached to that switch.



**Fig. 1.3 Switch**

- In modern computer networks the data is transferred using **packet switching**. Messages are broken into units called *packets*, and sent from one computer to the other. At the destination, data are extracted from one or more packets and used to reconstruct the original message. Each packet has a maximum size, and consists of a header and a data area. The header contains the addresses of the source and destination computers and sequencing information necessary to reassemble the message at the destination.

The typical structure of each packet is as shown by following figure –

Header	Data
10010...0	010111000010...10

### Advantages of packet switching network

1. The communication among the computers due to packet switching network is error free and there is no conversion overhead or transformation error.
2. More than one communications can be possible using the same link and hence there is no need for the other computer to wait for the ending of previous sessions. Ultimately this makes the networking cost effective.

- A protocol is a set of **rules** used for defining the communication between any two computers in the network. For example HTTP is a protocol (Hypertext transfer protocol) which defines the communication between web browser (may be Internet Explorer) and servers (may be Google server) when a request for some web page is made. Basically protocol defines two things: syntax and semantic of the message. The **syntax** defines what should be the format of message and what should be the type of the message to be exchanged. On the other hand, the **semantic** specifies the action to be taken on occurrence of particular event.

### 1.2.1 TCP/IP Model

The OSI model is most general and better suited for non TCP/IP networks. Normally there is a protocol stack which is used in network communication. This stack combines one or more layers. Such a model is called TCP/IP model. Generally the TCP/IP model is used in modern computers. Following figure shows the TCP/IP model –

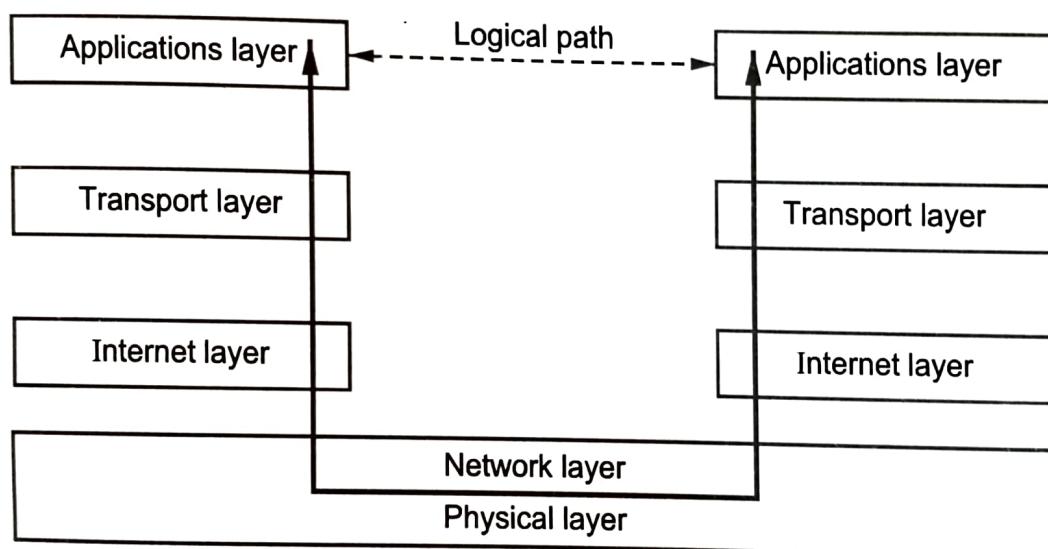


Fig. 1.4 TCP/IP model

- **Application Layer :** This layer combines the functions of OSI application, presentation and session layers. This layer makes use of protocols for simple applications such as e-mail or file transfer. The commonly used protocols are Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP).

- **Transport Layer :** Two main protocols are used in this layer and those are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The transport layer ensures that packets are received in the order they were sent. It also finds that no data is lost or corrupted. In case, if a packet is lost, the transport layer can ask the sender to retransmit the packet. The network implements this by adding an additional header to each datagram that contains more information. The TCP protocol performs the task of retransmission of lost or corrupted data. It is also responsible for transmission of data in correct sequence. the User Datagram Protocol (UDP), allows the receiver to detect corrupted packets but does not guarantee that packets are delivered in the correct order. Hence TCP protocol is called reliable protocol and UDP are called unreliable protocols. But UDP protocol is faster than TCP
- **Internet Layer :** This layer corresponds to network layer of OSI reference model. The data sent in the form of packets over the internet layer is called **datagram**. The internet layer is based on the IP protocol which makes use of **IP addressing** to determine the address of device it is communicating with.
- **Network Layer :** This layer combines the physical and data layers. The task of this layer is to route the data between devices present in the same network. This layer performs the mapping of the IP addresses and network physical addresses. In this layer the IP datagram is encapsulated in the format which is understandable by the network.
- **Physical Layer :** This layer ensures the safe and efficient transmission of data. It consists of electronic circuits for transmission of data.

### 1.2.2 TCP Data Transmission

Let us now discuss how the information flows from one machine to another using TCP/IP model –

**Step 1 :** When user makes some request from web browser to web server for retrieval of some page, then this request is first submitted to the transport layer of the local machine.

**Step 2 :** The transport layer breaks up this request into the TCP segments and some sequence number is added to these segments.

**Step 3 :** Such a request is then passed to the internet layer of the local machine. At the internet layer TCP segments are broken into the IP datagrams of necessary size. These IP datagrams are passed to network layer for transmission over the wire.

**Step 4 :** The network layer encodes the digital data into analog signal. This analog signal is transmitted over the network to the remote destination device.

**Step 5 :** The network layer of remote destination machine decodes the analog signal into the digital form. The resulting IP datagram is passed to the internet layer (of corresponding destination machine)

**Step 6 :** At the internet layer it is ensured that the IP datagrams have not lost the data. If there is no loss of data then these IP datagrams are passed to the transport layer.

**Step 7 :** The transport layer ensures that all the data has come up correctly and there is no request for retransmission (Note that it is the transport layer who performs the task of retransmission of data) When the transport layer receives the datagram into a correct sequence, then all the received information is properly reassembled in order to transfer it to application layer.

Thus the web server serves the request made by the web browser of the client machine.

## 1.3 TCP,IP and UDP Protocols

### 1.3.1 Internet Protocol

Internet Protocol (IP) is a network layer protocol which consists of addressing information. Using this information the communication between uniquely addressed computers is possible.

There are two functionalities provided by Internet Protocol–

- **Decomposition** of the initial information flow into packets of standardized size, and reassembling of data at the destination.
- The internet protocol **routes** the packet through successive networks, from the source machine to the destination which can be identified by its IP address.

- **Fragment Offset** indicates the position of the fragment's data relative to the beginning of the data in the original datagram
- **Time-to-Live(TTL)** it maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This helps in avoiding infinite loop.
- **Protocol** indicates which protocol receives incoming packets after IP processing is complete. If the protocol is TCP then it denotes 6 and if the protocol is UDP then it denotes 17.
- **Header Checksum** helps ensure IP header integrity.
- **Source Address** specifies the IP address of sending node.
- **Destination Address** specifies the IP address of receiving node.
- **Options** allows IP to support various options, such as security.
- **Data** contains information.

### 1.3.1.1 IP Addressing

Each host on a TCP/IP network is assigned a unique 32-bit logical address that is divided into two main parts: the network number and the host number. This address is called IP address. The IP address is grouped four into 8 bits separated by dots. Each bit in the octet has binary weight

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
↑	↑	↑	↑	↑	↑	↑	↑
128	64	32	16	8	4	2	1

As we know the IP address is divided in two categories network number and host number.



There are 5 classes based on two categories viz. A,B,C,D and E.

IP Address Class	Format	Range	Purpose
Class A	N.H.H.H	1 to 126	Very few large organisations use this class addressing.
Class B	N.N.H.H	128 to 191	Medium size organisations use this addressing.
Class C	N.N.N.H	192 to 223	Relatively small organisations use this class.
Class D	-	224 to 239	This class address is used for multicast groups.
Class E	-	240 to 254	This class addressing is reserved for experimental purpose.

## Web Technology

Here N stands for network number and H stands for host number. For instance in class C first three octets are reserved for network address and last 8 bits denote host address.

### 1.3.1.2 Domain Names

As we have discussed that each computer is assigned by a unique numerical address called IP address. This is because computer can remember the numerical addresses but humans can not remember numerical addresses, but they can remember names in better way. Hence the concept of domain name has come up. Domain names are the unique names of host computers that can be used instead of numerical addresses. For example instead of 192.168.1.92 we can use [www.mycomp.com](http://www.mycomp.com)

Typically servers have the fixed domain names (Domain name servers-DNS) but there may be a situation in which the clients get different addresses each time when the machine boots up. And such addresses are provided by Dynamic Host Configuration Protocol (DHCP).

### 1.3.2 TCP

The Transmission Control Protocol is used for

1. Safe delivery of data
2. Error detection
3. Assurance of the correct sequencing of data being received.

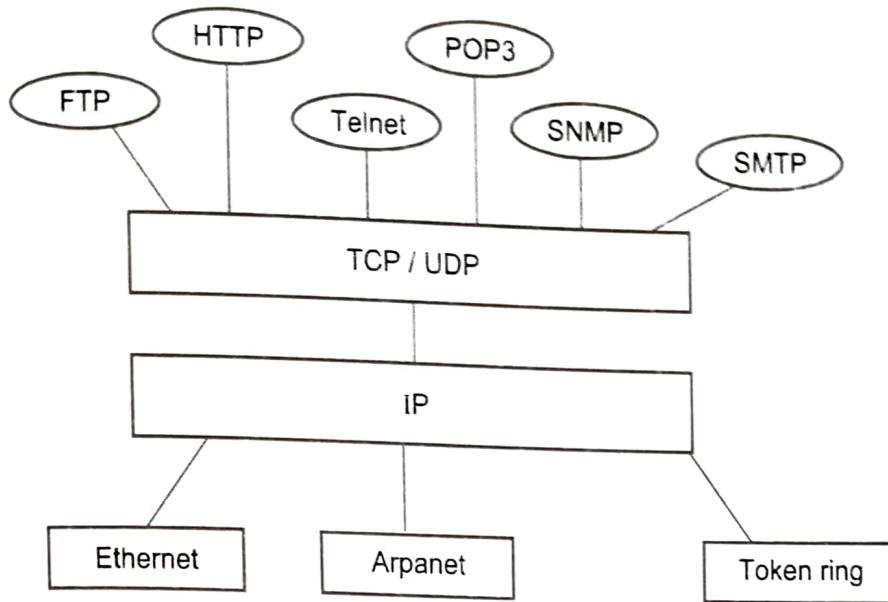
This protocol is called **connection oriented protocol** because before sending the data this protocol requires that two computers have established connections.

The TCP allows the transmission of arbitrary amount of data by breaking it into stream of separate IP packets. These IP packets are numbered so that it could be reassembled properly at arrivals. Along with the data an acknowledgement is also sent/received in order to know whether the reliable connection has occurred or not.

### 1.3.3 UDP

The user datagram protocol is a connectionless protocol without any error detection facility. This protocol is used for simply transmission of data. The UDP is known as an unreliable protocol however this is much faster than TCP.

## 1.4 Protocols Governing The Web



**Fig. 1.6 Internet Application Protocol**

- **FTP:** File Transfer protocol used for transfer of files.
- **HTTP:** Hyper Text Transfer Protocol is a communication protocol used to transfer the information on local area network and world wide web (WWW).
- **Telnet:** Allows user to connect remote host using simple terminal command.
- **POP3:** Post office Protocol of version 3 is used by the local e-mail clients to retrieve e-mails from remote server over TCP/IP connection. The remote internet server holds the e-mails and periodically clients can retrieve all the e-mails.
- **SMTP :** Simple Mail Transfer Protocol is a simple protocol which is extensively used for transfer of e-mails to remote servers.

For example to send an email we can use some specific address such as:

*technical@vtubooks.com* where *vtubooks.com* is a domain name server to which your mail can be sent.

- **SNMP:** Simple network management protocol is a set of protocols used for managing complex networks.

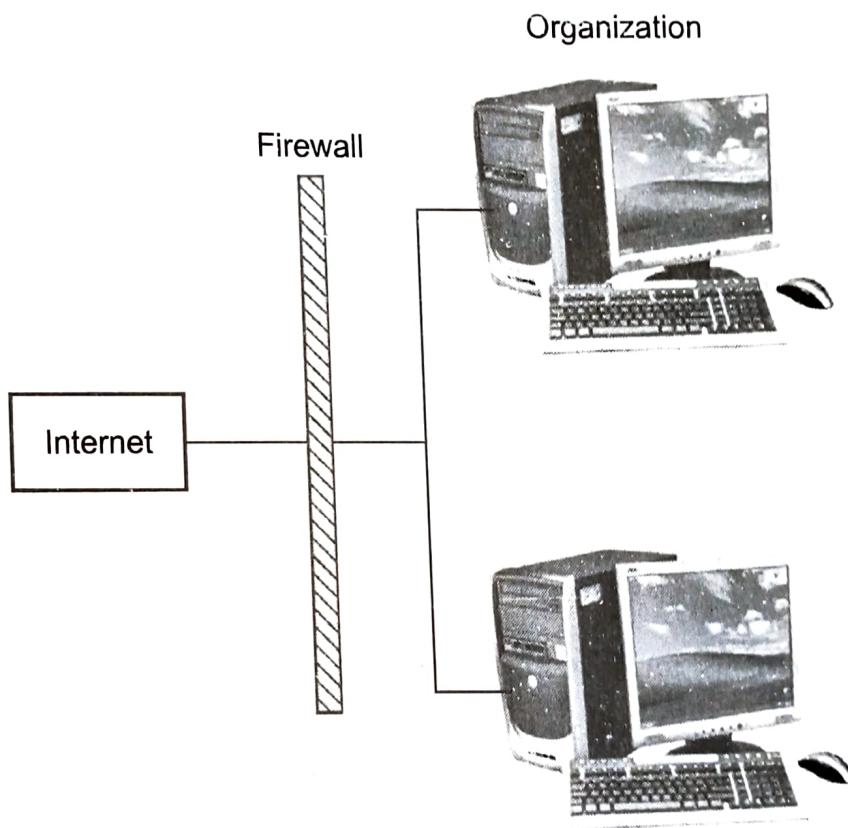
## 1.5 Some Commonly used Concepts in WWW

Internet is a linking of many computers all over the world. Using such linking user can share the information or communicate with each other. Internet is not a single centralised system to which we can pay for, in fact there are some internet service providers (ISP) to whom we pay for getting the internet services. In past most of the ISP's were run by telephone companies but now a days even some private groups who have sufficient money and expertise can be the ISPs. The typical set up required for getting the internetworking at your place is as shown below –

The **web page** is a document on the world wide web. And using **web browser** we can access various web pages. The most popularly used web browsers are *internet explorer*, *Mozilla Firefox* and *Netscape navigator*. The web pages are stored on a computer called **web server**. The **web site** is a collection of many web pages. We can browse these web pages using web browsers. The first web page of any web site is typically called as **home page**. On the web pages the **hyperlinks** may appear. The hyperlinks are underlined and colored phrases which is associated with some control link. We click at these links and can access to further web pages.

### 1.5.1 Firewalls

Firewall is a security systems which is designed to prevent unauthorized access to or from a private network. There are two types of firewalls *software firewall* and *hardware firewall*. All messages entering or leaving the intranet (private local area network) go through the firewall. The firewall system scans all the passing messages and blocks those messages that do not meet the security criteria.



**Fig. 1.8 Fierwall**

Using Firewalls the filtering can be done using following conditions

**IP address** : All the IP addresses other than the IP address class which is used in the organisation which is reading lot many files from the server is suspected one. And such IP address has to be blocked.

**Web Technology**

**Domain names :** Particular organisation can block some domain names or it may allow access to specific domains.

**Port :** The organisation can block some ports and prevent the access to server.

**1.5.2 Proxy Servers**

The functionality of firewall is combined with the proxy servers. The proxy server is a server which serves the requests made by the clients by forwarding those requests to the corresponding servers. The proxy servers sometimes can alter the clients' requests. The proxy servers are mostly used to control, monitor or unbound the network traffic. Some proxy servers cache the requested data. This helps in quick serving of the request when the demand for the previously read data is made.

Using Proxy server setting an organisation can block certain sites so that controlling of network traffic can be possible.

**1.5.3 Web Browsers**

- Web browser is a kind of software which is basically used to use resources on the web.
- Over the networks, two computers communicate with each other. In this communication, when request is made by one computer then that computer is called a **client** and when the request gets served by another computer then that computer is called **server**. Thus exchange of information takes place via Client-Server communication.
- When user wants some web document then he makes the request for it using the web browser. The browsers are the programs that are running on the clients' machines. The request then gets served by the server and the requested page is then returned to the client. It is getting displayed to the client on the web browser.
- The web browsers can browse the information on the server and hence is the name.
- Various web browsers that are commonly used are
  - Internet explorer
  - Mozilla Firefox
  - Netscape Navigator.

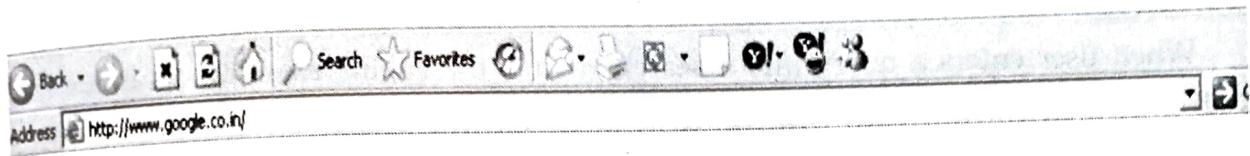
Some other web browsers are

- Opera
- Google Chrome
- Safari.

- Web browser supports variety of protocols but the most commonly used protocol on the web browser is Hyper Text Transfer Protocol(HTTP). This protocol is typically used when browser communicates with the server.

### 1.5.4 URLs

The Uniform Resource Locator (URL) is unique address for the file that has to be accessed over the internet. When we want to access some website we enter its URL in the address bar of the web browser. For example if we want to access www.google.com then we must specify its URL in the address bar as shown below -



Here

http://www.google.com

name of the protocol

domain name server

However any other file such as some text file or image file or some HTML file can also be specified. The URL contains name of the protocol such as **http://**

The URL may contain the name of the protocol as such as **ftp**. For example

**ftp://ftp.funet.fi/pub/standards/RFC/rfc2166.txt**

The protocol identifier and the resource name are separated by a colon and two forward slashes. The syntax of writing URL is as given below,

**protocol://username@hostname/path/filename**

Sometimes instead of domain name servers IP addresses can also be used, for example,

**http://192.168.0.1**

But use of IP address as URL is not preferred because human cannot remember numbers very easily but they can remember names easily.

### 1.5.5 Search Engines

Search engines is an information retrieval system designed to search the required information from world wide web (WWW). To retrieve the desired information user has to submit his demand in words or in phrases. This process is known as "firing of query". The search engines then present the user list of desired information.

#### 1.5.5.1 How Search Engines Work ?

The search engines performs following three tasks for finding the required information -

Some examples of web applications are

1. Airline reservation systems
2. Search engines
3. Message boards
4. Chat engine
5. Shopping cart
6. Net-banking

### 1.8.1 E-commerce

E-commerce is the most widely used web application. It consists of buying and selling of products over the electronic systems such as internet. The concept E-commerce encourages internet marketing, online transaction processing and electronic fund transfer.

Mainly on-line shopping is conducted using E-commerce. On line shopping is the process in which the consumers go through purchase products or services over the internet. The on line catalogue can be viewed for purchase of certain item.

#### Working with E-commerce applications

- First of all user goes to a web site for purchasing the product. He reviews the information about the product, then adds the desired item to a shopping cart. Then he clicks the **checkout** button.
- The web application calculates the total number of items and price, then it deducts the relative taxes and charges and finally presents the bill/invoice to its customer.
- The customer then reviews the bill and click on the **purchase** button.
- This purchase request is then sent to the web server in order to process for secure web service. A secure web page is then prepared to accept the credit card and personal information. When the customer enters the credit card and personal information, this data is then verified by the corresponding bank. On receiving the valid information a customer gets a notification about acceptance of his order and its processing.

### Design strategies for E-commerce

Implementing E-commerce solution for a web site is the most demanding activity. Such type of applications has a very long project design phase. The **design strategy** involves following phases—

- **Requirement gathering and Analysis**

In this phase, various objects that can be identified those are as given below –

- A product or a catalogue of products
- Customers or the users who view this product on the web site
- The place where the product is being sold.
- A way by which client information can be accepted . For example On-line form filling or email facility
- The manner by which the product will be accepted
- The way by which money can be accepted. It requires secure page and connection to a bank.
- shipping facility to fulfil the product
- A way by which the warranty claim can be accepted
- A display by which customer may come to know the status of his order

- **Project planning**

The project manager has to keep the E-commerce project on schedule and ensure that the scope of the project will not grow. He must identify the required tasks for deciding the schedule and budget of the project. Then he should recruit the appropriate resources.

The project manager generally writes the discovery document in which milestones must be decided with the deliverables. There should be some change control process for handling multiple changes that may sprung up during design and implementation phases.

The project manager is supposed to conduct regular meetings with the full team members to keep the project on schedule and to keep all members communicating.

- **Design**

During the design phase the system architecture is prepared. The system architecture should be designed in such a manner that it should handle expected load, specified hardware and software. The network engineer must prepare a security plan which will ensure the security of the application. During this phase, an application developer must prepare data models with the help of DFD,ER diagrams, and state charts. He also codes the rough HTML template pages. The design must be properly refined so that it could be directly mapped to its implementation. The application developer must also decide the

file-naming convention. Based on the design the QA manager can write the test plan and prepares the test suit so that shopping cart application can be properly tested.

- **Implementation**

After completion of design phase the project enters in the implementation phase. During this phase actual application is built, graphical assets get implemented, integrated with the application and finally get deployed with the server.

- **Testing**

Finally the project enters in a testing phase. In this phase, bugs are reported and fixed. After thorough testing the web application can be launched.

## 1.9 Writing Web Projects

Developing web projects is a crucial activity and web project development differs from traditional software project development activity because of some web project characteristics. Following are such characteristics –

### 1. Development schedules of web projects are dramatically short

As web is fairly new concept both the developers and clients underestimate the time and resources required to develop the web project. Hence it becomes impossible to complete the project on the predetermined schedule.

### 2. Scope changes occur even at implementation stage

This is most common happening in any web project. During the implementation stage many times client demand for something else and changing the scope at this stage of web development is very expensive and must be tracked and managed.

### 3. Project managers are not always client managers

Many times project manager is a person who is not a person responsible for managing the client's expectations.

### 4 New technologies are used often without any technical support

In web development process, new software packages are used which are in beta stage because such packages are easily available with new and enhanced features.

### 5. Pricing models for web projects does not exist

Costing a web project is very difficult because there is no defined cost module for various types of web applications.

## 6. Standards for web production do not exist

There are no predefined set of rules or standards defined for the web projects. If there exists some standards for developing the web project then Web Company can gain efficiency in product development.

## 7. Clients are unwilling to bear the web development cost

Clients are more eager to see the creative concept and do not want to spend during the planning stage.

## 8. Team roles are less specialized

There are various ways to develop a web page one can use Flash, HTML, Java and so on. The server side developer may also be applet designer and can design applets at client side. Hence web designing activity is not specialised for specific category of work.

## 9. The project manager's responsibilities are very broad

In web project development a project manager can be producer developer or simply a project manager. He has to do variety of jobs such as planning for cost, schedule or he may have to take part in the development process.

## 1.10 Identifying Objectives of Web Project

According to James Lewis following must be the objectives of the web projects –

1. Web projects must be **Specific**
2. The web projects should be **Measurable**
3. The schedule of the web projects must be **Attainable**
4. There should be **Realistic** approach for deciding the web project
5. It should be **Time-limited** or should not be continued for a very long period.

These objectives are also called as SMART objectives.

## 1.11 Target users

The contents of web are directly dependant upon the kind of users who are visiting the web sites. But it is observed that many web developers and designers do not make a research on identifying the target users accurately. For identifying the web site's success only checking the site traffic is not sufficient, it is equally essential to determine whether the objective of the web site is fulfilled or not. The web designers must find out what the target users want to see on the web site, or what will make the users to purchase the product or use the service, Following are some ways to find it out –

## 1. Market Research

Market research will help you to understand exactly what kind of audience is accessing your web page and what is its demand. There are many firms which provide these kinds of services to web companies. Hence your web company can give a contract to such firms for making the market research.

## 2. Focus Groups

Getting the service for market research from some private firms is costly. Hence we can use another approach of identifying target users. The focus group is a group of people who represent the target users or target audience. The web companies can make use of focus group for getting the feedback on proposed ideas on the web site. This group can also give the comment on how target audience think on the ideas that are put on the website.

## 3. Understand Intranet audience

If your web site is created for an intranet then intranet access is for several departments and every department has different needs. Hence becomes very easy and manageable to understand the target audience for intranet.

## 1.12 Web Team

Web team consists of group of people who are responsible for producing and managing the website. There are different roles of a typical web team and those are –

- **Project Manager**

Various responsibilities of project manager are – planning for schedule, calculating the budget and cost of the project, allocating resources and managing the team. The project manager deals with administrative issues. He is the person who communicates with other team members and with the clients.

- **Account Manager**

Account manager is responsible for selling different products. The account manager interacts with the client regularly.

- **Technical leader**

Technical leader observes the project from technical point of view. He is the person who communicates the project manager and tells him whether or not the background of the project is technically strong. Technical leader chooses programmer, database programmer, system integrators and other skilled members for the team. He is also responsible for preparing technical briefs/documents for the project.

- **Programmer**

Programmer is a person who writes the source code for the application. A programmer must be a skilled person in specialised area. The specialisation may vary from server side scripting, database programming, applet designing and so on.

- **Network engineer**

A network engineer configures and sets up the web server, registers the domain names. Sometimes he also sets up the E-mail server. He should also be database administrator and security expert.

- **Security expert**

A Security expert is a person who decides the security strategies for the web site. He knows and implements various encryption techniques for the website. He should discuss security strategies that could be applicable to the web site. For the commerce based sites, network engineer or programmer is often a security expert.

- **Creative lead**

The creative concepts and web site's design is managed by the creative lead. He is the person who is like art director and provides the guidance on look and feel of website. There should be interaction among the creative lead, technical leader, programmer and web specialist for determining the technical feasibility of the concept.

- **Web Designer**

Designer is a person who actually creates the web sites design. Popularly used tool in this areas is Adobe Photoshop. The web designers should have adequate knowledge of designing principles, information modelling and interaction designing. Placing appropriate images on the web page is the most important skill in this area. There should be proper communication between creative lead and web designer. In fact, creative lead himself is web designer also.

- **Information Architect**

The information architects arrange the information neatly on the web site. He should arrange the information in such a manner that user can understand and handle it quite easily. He is also responsible for information architect, information navigation, data retrieval. Sometimes user needs to be informed about some key messages regarding errors, technical needs and privacy messages. Such information must be highlighted properly.

- ***Copywriter***

Copywriter is a person who owes for the information and representation of the web page design. The information architect, client who supplies information or a web designer can be a copywriter.

- ***Quality assurance leader***

The QA leader is a core person who ensures that product satisfies the scope and functional specification. A QA person interacts with web developers and ensures that all the bugs are identified and already resolved while delivering the product. A quality assurance lead is responsible for quality of the web site. The QA leader also prepares the test plan.

- ***Tester***

The tester tests the web project using the test plan. For small projects the two roles – QA leader and tester are played by a single person.

- ***Audio and video engineer***

The audio and video engineer is responsible for designing sounds and videos for the web site.

- ***Graphics designer***

The graphics designer creates the artwork for the website usually in 3-D format. This person works with creative lead.

- ***Web cast specialist***

The web cast specialist is a person who transmits or broadcasts the web over the internet.

- ***Media Buyer***

The media buyers are the advertising agencies who buy the place on the web site for advertisements.

- ***Strategic planner***

This person provides the customer insight to the web development team. This person has to do a market research for understanding the customers' views and demands.

### 1.12.1 Technological Advances and Impact on Web Team

#### What are Technological Advances?

People started using web for more and more consumer purpose such as shopping, advertising or information retrieving and so on. The web has become like television from which the information can be passed to the users in an interactive form. Following are some distinct technical advances -

**Web as Interactive Television :** The broadband internet access is an important development in the internetworking environment. Due to this development the information in the audio and video format can also be sent over the internet. WebTV allows user to watch TV as well as to access web pages simultaneously on one screen. Many websites offer training for creating contents for interactive television. People with experience in production, scriptwriting videography must be the part of a web team.

**Web as information Store :** Web has become central repository of information for national and international organisations. Such type of web projects require the ability to search the data effectively. The web team should consists of the people who are skilled programmer and are able to create software for efficient data retrieval. The web team should necessarily consists of security experts and database programmers.

**Web and E-Commerce :** Web is also popular for its ability to sell the products on-line. User can view a wide range of products and purchase various brands on-line. Web teams that provide E-commerce applications must consists of security experts, database programmer and network engineers. The team members should able to handle, ordering, shipping, updating inventory, invoicing and so on.

#### How does web team cope-up with the changes?

Technological changes influence the web team to a large extent. As technology changes, new tools will be created and new skills will be essential. Hence web team must include some specialised members within it. Finding the right person for the specific task is an important responsibility of the project manager. Following fig represents how web team can be modified by adding the specialised members within it to manage the technological changes.

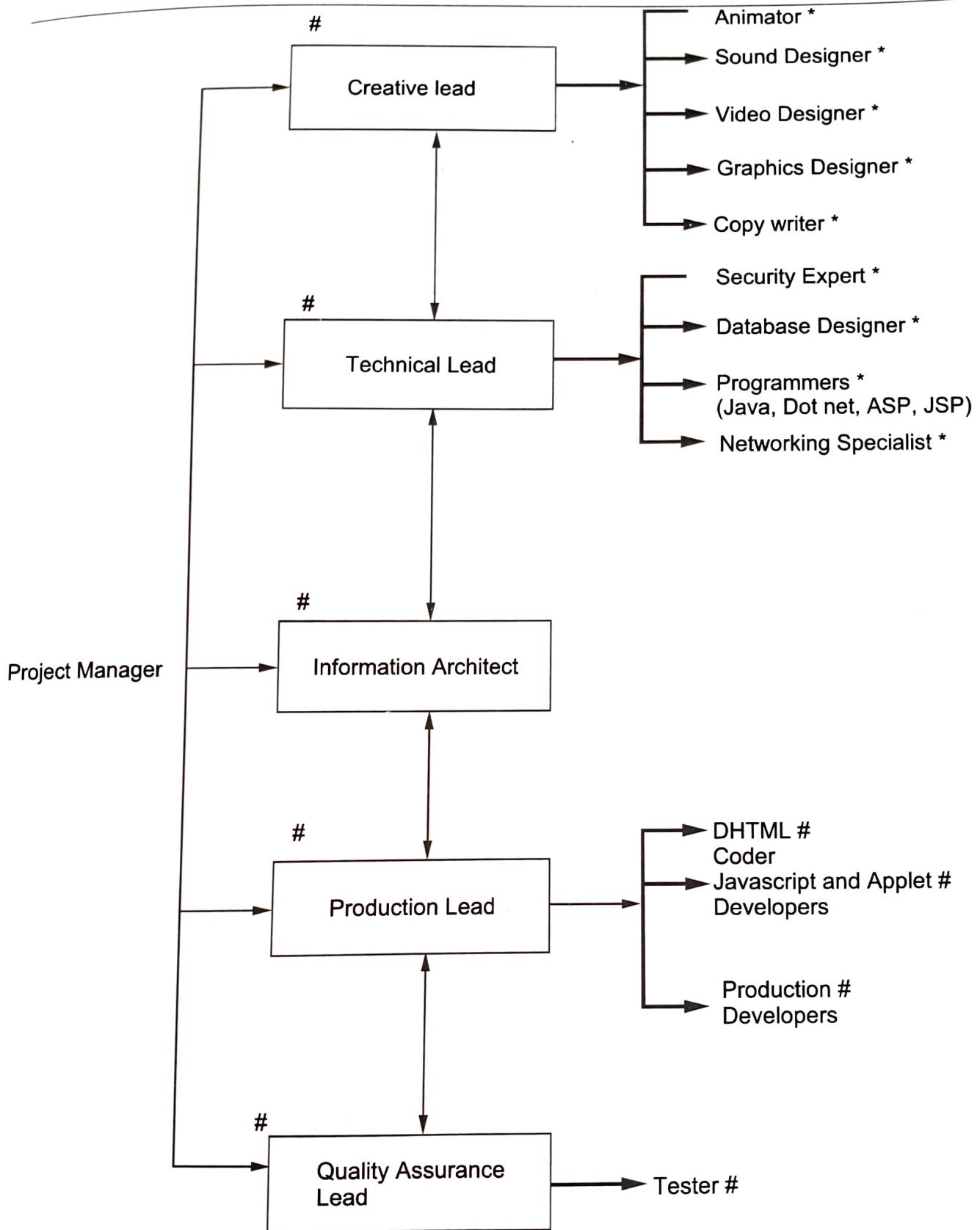


Fig. 1.10 Web Team preparing for technical change

In Fig. 1.10 the thick line denotes the line of management, the \* denotes specialised team and # denotes core team members.

## **1.13 Planning and Process Development**

For creating a web project both planning and process development are important activities. Web design should always start with planning. Planning is an important stage for ensuring the business goals.

### **1.13.1 Early Planning**

Early planning is required to know the project's objectives, client's expectations and user's need. Various techniques that can be adopted to know the audience are

- (i) Interviewing      (ii) Focus groups and Market research

#### **(i) Interviewing**

A well planned interview between the team members and client helps to understand client's needs. While conducting interviews following points must be remembered –

- Frame the questions in such a manner that objectives of the web project can be clearly understood. Do not ask narrow questions for which answers will be only "yes" and "no" type.
- Clearly explain the ideas about the web site to the user. But remember, that feature is associated with the cost and weigh each feature against the clients' expectation and the amount of money client wishes to spend.
- Bring tape recorder or assistant for taking the notes so that complete concentration will be on listening to the client.
- Use proper set of questionnaire to interview the users.

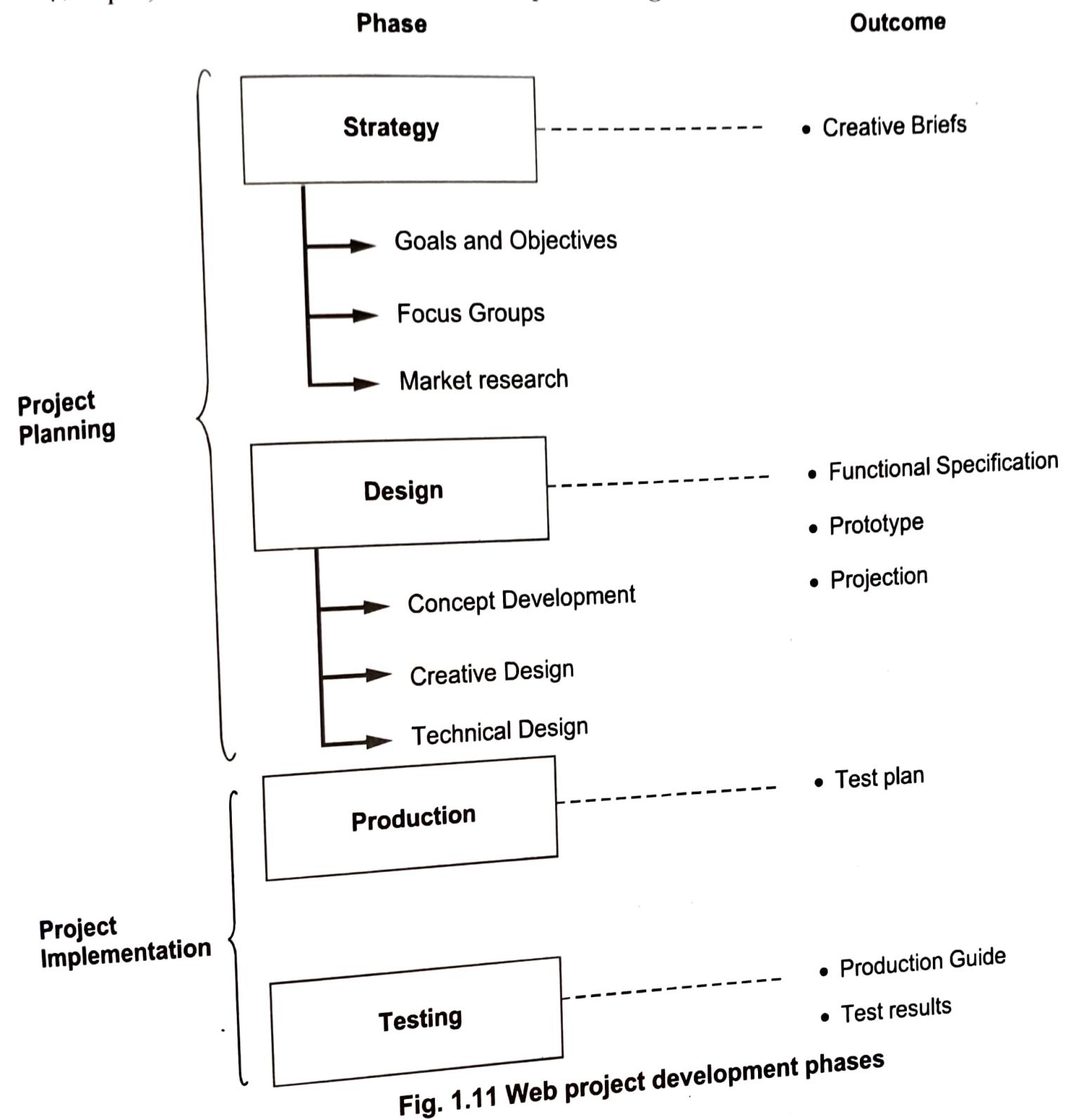
#### **(ii) Focus groups and Market Research**

Focus group is a group of people who act as the representative target audience. Market research can be made on for finding the target user for the web site.

Early planning helps to gather the end user requirements. In early planning focus groups helps to understand the mind set of target audience. Gathering end-user requirements is important parameters that help the web designer to follow visitors' expectations. There are some consistent requirements for any web project and those are regarding the speed, browsers and browser version.

### 1.13.2 Development Stages and Strategies

Web project can be designed in the four phases as given below –



### Phase I : Strategy

In this phase, a strategic planner or project manager along with the client determines the objective of the site. As an output of this phase **creative briefs** are prepared. The creative brief is a kind of document in which project objectives, requirements and key insights are clearly mentioned. Every team member makes use of creative brief as a guideline for the development.

### Phase II: Design

In this phase actual design of the web site is done with the help of creative and technical team members. The front end is designed by the creative team in which user interface and interactions are designed. The back end is designed by the technical team which is responsible for designing the database architecture. As an outcome of this phase functional and technical specifications, site architecture are prepared.

### Phase III: Production

During this phase actual site is built using the source code. Functionalities and features of the web site are closely examined. If the client demands for a change in any functionality then a change order is issued. At the end of this phase a production guide is prepared.

### Phase IV: Testing

At this phase all the functionalities and features of the web site are tested, bugs are identified and resolved before launching the web site. The QA manager develops the test plan. The test suit mentioned in it used to test the product thoroughly.

## 1.14 Template for Creative Brief

The creative brief serves as a basis for web development. The layout of creative brief is as shown below –

## Creative Briefs

### Project Details

Date:

Prepared by:

Project name:

Design Lead:

Phone:

Email:

Product Manager:

Product Release: (mm/dd/yy)

### Project concept

A short description on the idea of the project

### Objectives

This section lists the business and product objectives

### User Value/ Benefits

This section describes the benefits of the product

### Target Audience

Use this section to describe the audiences (primary and secondary) for this project/product. Include any information that you have about the audience (demographics and psychology)

### Personality/Tone/Guidance

This section describes the overall tone or personality for communication

### Current Mind-set

In this section how target thinks relative to the brand or application

### Target Audience Insight

This section describes what the target audience must think after viewing the product?

## **Web Technology**

### **Unit-I**

#### **JAVA**

##### **Introduction:**

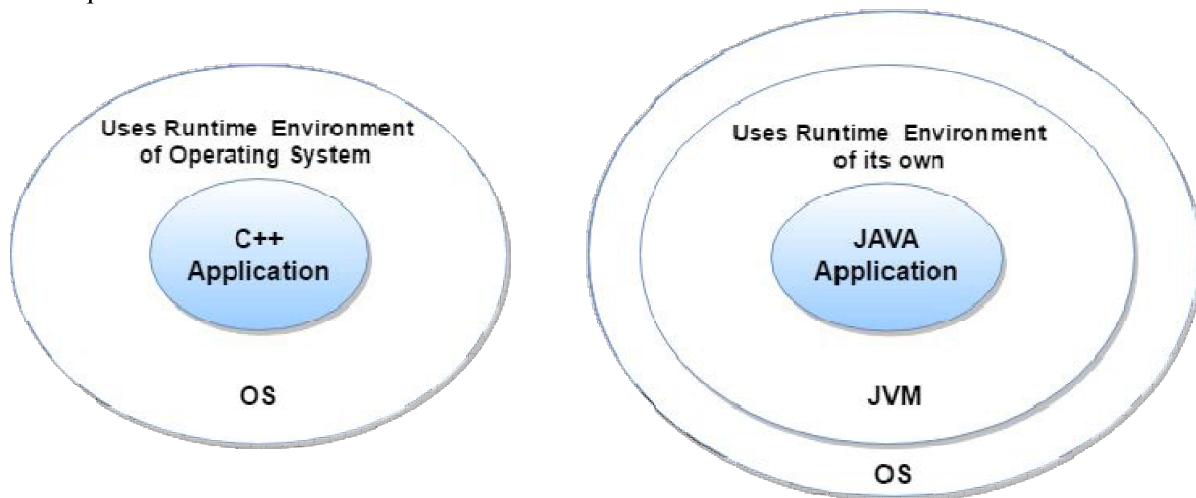
You use word processors to write documents, Web browsers to explore the Internet, and email programs to send email. These are all examples of software that runs on computers. Software is developed using programming languages. There are many programming languages—so *why Java?*

The answer is that Java enables users to develop and deploy applications on the

- Internet for servers,
- desktop computers, and
- small hand-held devices

The future of computing is being profoundly influenced by the Internet, and Java promises to remain a big part of that future. Java is *the Internet programming language*.

Java is a programming language and a platform. Java is a high level, robust, secured and object-oriented programming language. Any hardware or software environment in which a program runs is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.



## History of Java

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of java starts from Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions etc. But, it was suited for internet programming. Later, Java technology was incorporated by Netscape.

Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc. There are given the major points that describe the history of java.

1) **James Gosling, Mike Sheridan, and Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers called **Green Team**.

2) Originally designed for small, embedded systems in electronic appliances like set-top boxes.

3) Firstly, it was called "**Greentalk**" by James Gosling and file extension was .gt.

4) After that, it was called **Oak** and was developed as a part of the Green project.

5) Why Oak? Oak is a symbol of strength and chosen as a national tree of many countries like U.S.A., France, Germany, Romania etc.

6) In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.

7) Why had they chosen java name for java language? The team gathered to choose a new name. The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA" etc. They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell and fun to say.

According to James Gosling "Java was one of the top choices along with Silk". Since java was so unique, most of the team members preferred java.

8) Java is an island of Indonesia where first coffee was produced (called java coffee).

9) Notice that Java is just a name not an acronym.

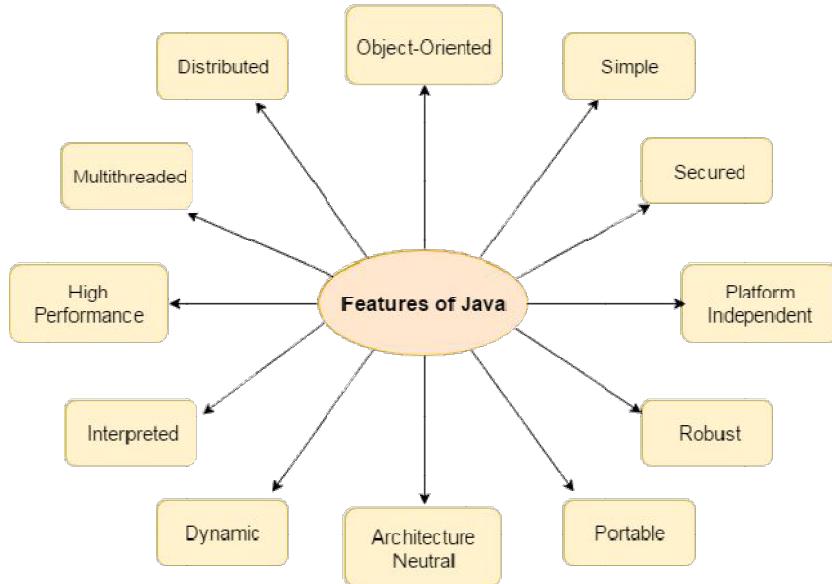
10) Originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.

11) In 1995, Time magazine called Java one of the Ten Best Products of 1995.

12) JDK 1.0 released in(January 23, 1996).

## Features of Java

The key considerations were summed up by the Java team in the following list of buzzwords:



- Simple:** Java is very easy to learn and its syntax is simple, clean and easy to understand. Java syntax is based on C++. Java has removed many confusing and rarely-used features e.g. explicit pointers, operator overloading etc.
- Object-Oriented:** Java is object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior. Basic concepts of OOPs are: Object, Class, Inheritance, Polymorphism, Abstraction and Encapsulation.
- Portable:** Java is platform independent because it is different from other languages like C, C++ etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

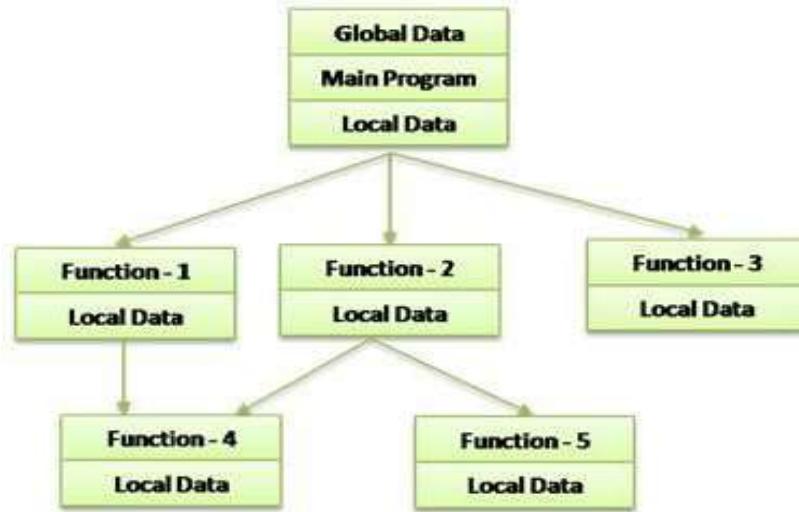
Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA)

- Secured:** Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because: *No explicit pointer, Java Programs run inside virtual machine sandbox*

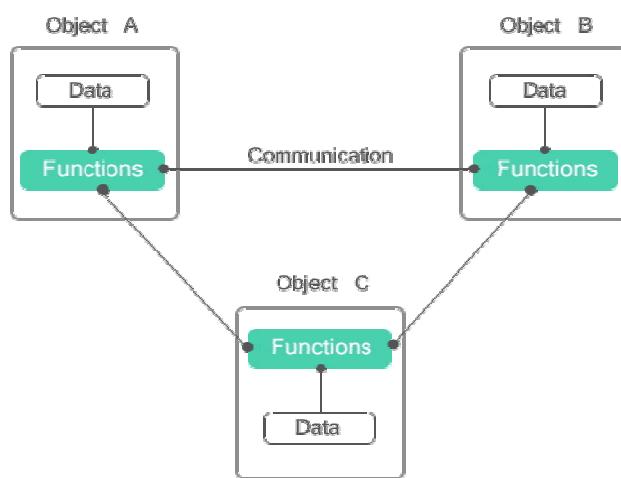
5. **Robust:** Robust simply means strong. Java is robust because: It uses strong memory management. There are lack of pointers that avoids security problems. There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
6. **Architecture neutral:** Java is architecture neutral because there is no implementation dependent features e.g. size of primitive types is fixed. In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.
7. **Interpreted:** the Java bytecode was carefully designed so that it would be easy to translate directly into native machine code for very high performance by using a just-in-time compiler. Java run-time systems that provide this feature lose none of the benefits of the platform-independent code.
8. **Multithreaded:** A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications etc.
9. **Distributed:** It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages i.e. C and C++.

## Object Oriented Programming:

Before going in detail of OOP we have to understand **procedure oriented programming language**: in the procedural languages the problem is divided in to number of function and each function accomplish a particular task. The primary focus of procedure oriented languages is functions.



**Object Oriented Program:** OOPs treat data as a critical element in th program development and does not allow to flow freely around the system. OOPs allow decomposition of problem in to number of entity called object and build data and method around the object.



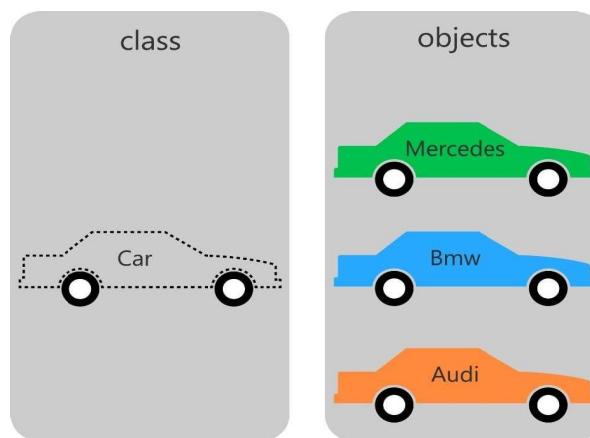
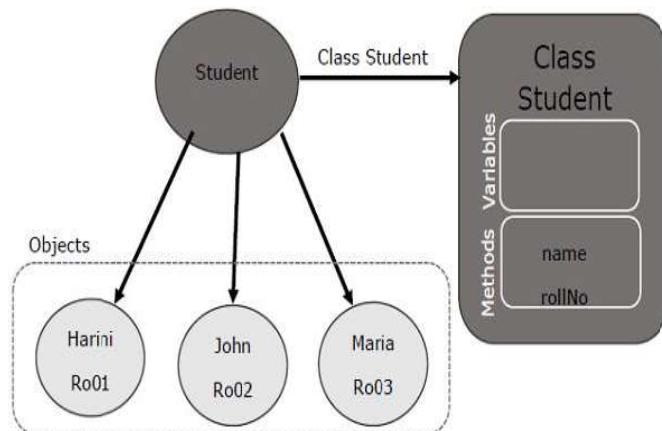
## Basic Concept of OOP:

- Class and Object
- Abstraction
- Encapsulation
- Polymorphism
- Inheritance

### Class and Object

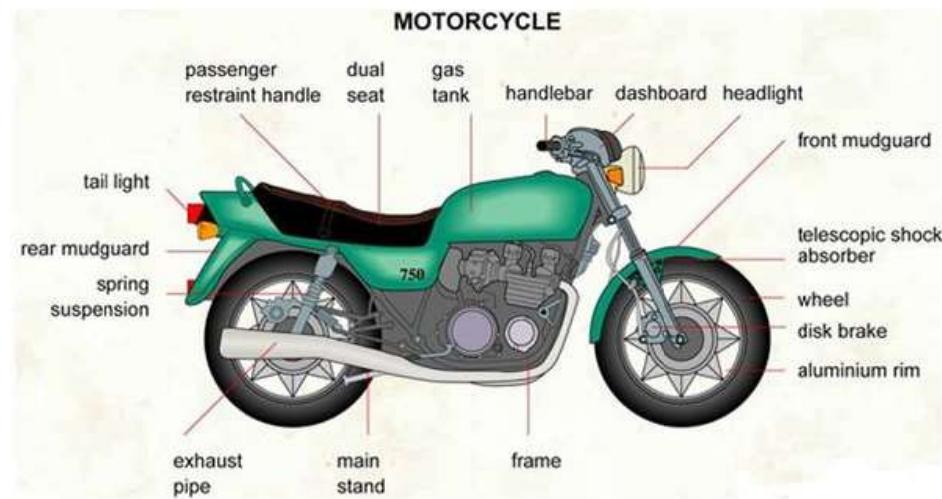
The class is at the core of Java. Class define the shape and nature of object. A class is a *template* for an object, and an object is an *instance* of a class. A class is declared by use of the **class** keyword.

Objects are basic run time entity. It take space in memory. When program is executed the one object interact other object by sending message.



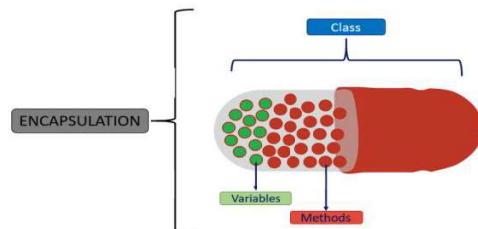
**Abstraction:** Hiding internal detail and showing functionality is known as abstraction. Eg. Phone Call. In java we can use Abstract Class and Interface to achieve abstraction

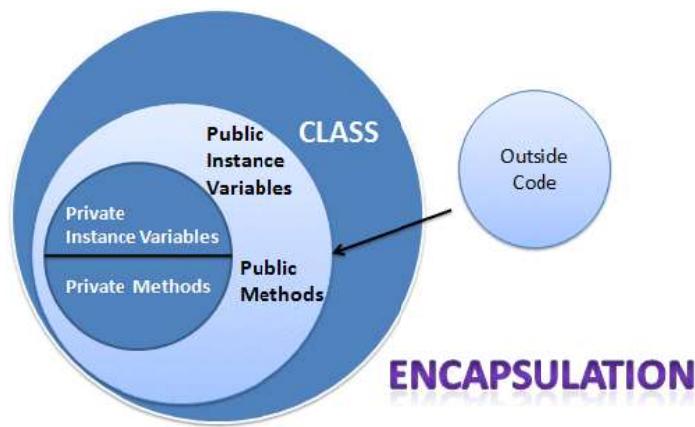
## Abstraction



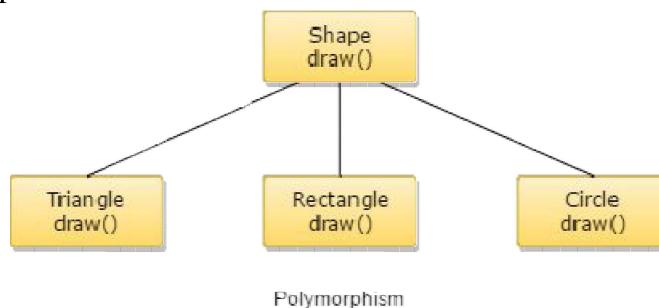
Real Life Example of Abstraction

**Encapsulation:** Binding or wrapping code and data together into single unit is known as encapsulation eg. Capsule

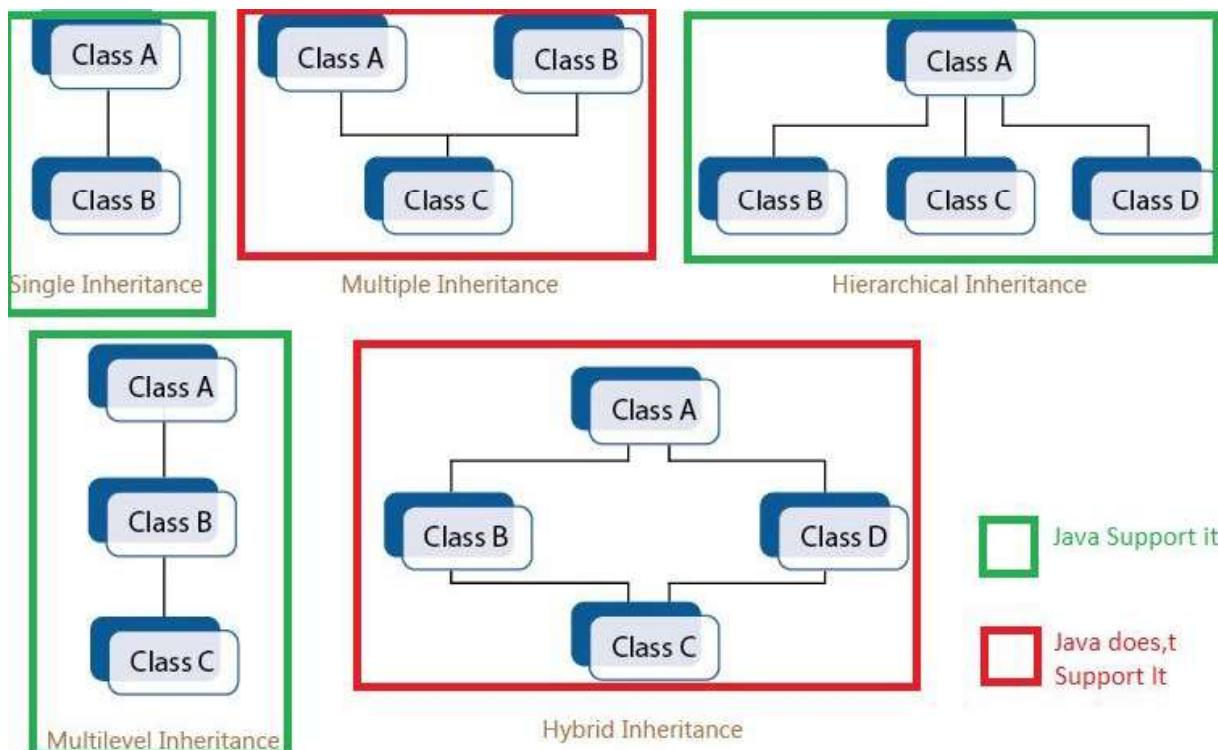




**Polymorphism:** Polymorphism in java is a concept by which we can perform a single action by different ways. Polymorphism is derived from 2 greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms. Method overloading and Method overriding achieved Polymorphism.



**Inheritance:** The process by which object of one class acquires the properties and functionalities of another parent object of class is called inheritance. It provides code reusability.



## A First Simple Program

For executing any java program, you need to

1. Install the JDK if you don't have installed it, download the JDK and install it.
2. Set path of the jdk/bin directory.
3. create the java program
4. compile and run the java program

```
/*
This is a simple Java program.
Call this file "Hello.java".
*/
Comment class Hello{
    // Your program begins with a call to main().
    public static void main(String args[]) {
        System.out.println("Hello Java.");
    }
}
```

```
C:\Users\Sandeep>cd..

C:\Users>cd..

C:\>cd java
|
C:\Java>javac Hello.java

C:\Java>java Hello
Hello Java.

C:\Java>
```

### Closer Look at the First Sample Program

**Comment:** Multiline, Single Line, Documentation

**public:** keyword is an access specifier, which allows the programmer to control the visibility of class members

**static:** The keyword static allows main( ) to be called without having to instantiate a particular instance of the class. This is necessary since main( ) is called by the Java Virtual Machine before any objects are made

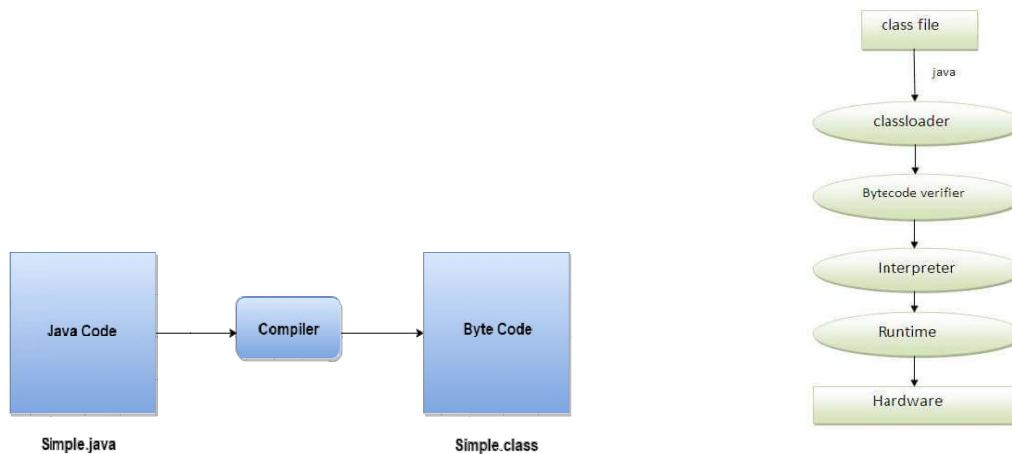
**void:** The keyword void simply tells the compiler that main( ) does not return a value.



**main:** main( ) is the method called when a Java application begins. Keep in mind that Java is case-sensitive. Thus, Main is different from main. It

**String args[ ]:** In main( ), there is only one parameter, albeit a complicated one. String args[ ] declares a parameter named args, which is an array of instances of the class String. (Arrays are collections of similar objects.) Objects of type String store character strings. In this case, args receives any command-line arguments present when the program is executed.

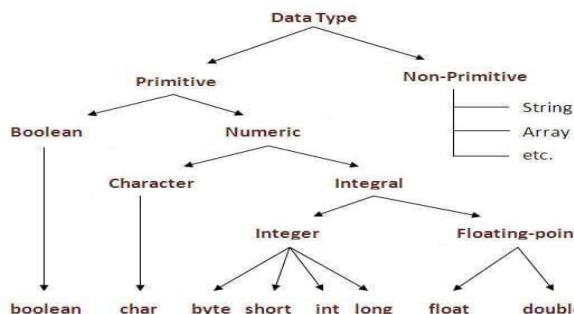
**System.out.println:** System is a predefined class that provides access to the system, and out is the output stream that is connected to the console. println( ) displays the string which is passed to it.



## Data Types in Java

Primitive data types: The primitive data types include Integer, Floating Point, Character and Boolean.

Non-primitive data types: The non-primitive data types include Classes, Interfaces, and Arrays.



Data Type	Default Value	Default size	Range	Used As
boolean	false	1 bit	true or false	boolean b;
char	'\u0000'	2 byte	0 to 65,536	char ch1, ch2;
byte	0	1 byte	-128 to 127	byte b, c;
short	0	2 byte	-32,768 to 32,767	short t;
int	0	4 byte	-2,147,483,648 to 2,147,483,647	int a;
long	0L	8 byte	- 9,223,372,036,854,775, 808 to 9,223,372,036,854,775, 807	long days;
float	0.0f	4 byte	1.4e-045 to 3.4e+038	float avg, sum;
double	0.0d	8 byte	4.9e-324 to 1.8e+308	double pi, r, a;

## Java Variables

Variable is name of reserved area allocated in memory. In other words, it is a name of memory location. It is a combination of "vary + able" that means its value can be changed. A variable is defined by the combination of an identifier, a type, and an optional initializer. all variables have a scope, which defines their visibility, and a lifetime.

```
Data_type identifier [= value][, variable [= value] ...] ;
```

Examples:

```
int a, b, c;           // Declares three ints, a, b, and c.  
int a = 10, b = 10;   // Example of initialization  
byte B = 22;          // initializes a byte type variable B.  
double pi = 3.14159; // declares and assigns a value of PI.  
char a = 'a';          // the char variable a is initialized with value 'a'
```

There are three types of variables in java:

- o local variable: A variable declared inside the body of the method
- o instance variable: A variable declared inside the class but outside the body of the method
- o static variable: A variable which is declared as static is called static variable. It cannot be local

```
class A{  
    int data=50;//instance variable  
    static int m=100;//static variable  
    void method(){  
        int n=90;//local variable  
    }  
}
```

## Identifiers in Java

All Java components require names. Name used for classes, methods, interfaces and variables are called **Identifier**. Identifier must follow some rules. Here are the rules:

- All identifiers must start with either a letter( a to z or A to Z ) or currency character(\$) or an underscore.
- After the first character, an identifier can have any combination of characters.
- A Java **keyword** cannot be used as an identifier.
- Identifiers cannot be true, false or null.
- Identifiers in Java are case sensitive; foo and Foo are two different identifiers.

Samples of acceptable variable names: YES	Samples of unacceptable variable names: NO
Grade	Grade(Test)
GradeOnTest	GradeTest#1
Grade_On_Test	3rd_Test_Grade
GradeTest	Grade Test (has a space)

## Operators in java

**Operator** in java is a symbol that is used to perform operations. For example: +, -, \*, / etc.

There are many types of operators in java which are given below:

- o Unary Operator,
- o Arithmetic Operator,
- o Relational Operator,
- o Bitwise Operator,
- o Logical Operator,
- o Assignment Operator.

### The Unary Operators

The unary operators require only one operand; they perform various operations such as incrementing/decrementing a value by one, negating an expression, or inverting the value of a boolean.

Operator	Description
+	Unary plus operator; indicates positive value (numbers are positive without this, however)
-	Unary minus operator; negates an expression
++	Increment operator; increments a value by 1
--	Decrement operator; decrements a value by 1
!	Logical complement operator; inverts the value of a boolean

### Arithmetic Operators

The Java programming language provides operators that perform addition, subtraction, multiplication, and division. There's a good chance you'll recognize them by their counterparts in basic mathematics. The only symbol that might look new to you is "%", which divides one operand by another and returns the remainder as its result.

Operator	Description
+	Additive operator (also used for String concatenation)
-	Subtraction operator

*	Multiplication operator
/	Division operator
%	Remainder operator

## Relational Operators

The equality and relational operators determine if one operand is greater than, less than, equal to, or not equal to another operand. The majority of these operators will probably look familiar to you as well. Keep in mind that you must use "==" , not "=", when testing if two primitive values are equal.

==	equal to
!=	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

## Logical Operator

&	Conditional-AND
	Conditional-OR
?:	Ternary (shorthand for if-then-else statement)

## Assignment Operator

=	Simple assignment operator
---	----------------------------

## Bitwise Operators

The bitwise & operator performs a bitwise AND operation.

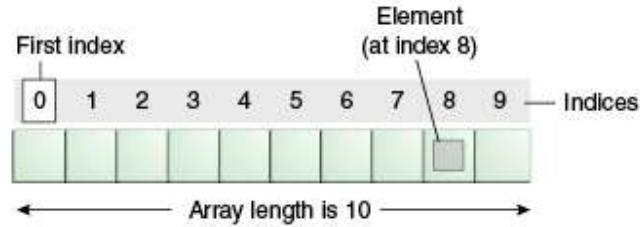
The bitwise ^ operator performs a bitwise exclusive OR operation.

The bitwise | operator performs a bitwise inclusive OR operation

shift operator "<<" shifts a bit pattern to the left, and the signed right shift operator ">>" shifts a bit pattern to the right.

## Arrays

An *array* is a group of like-typed variables that are referred to by a common name. Arrays of any type can be created and may have one or more dimensions. Array in java is index based, first element of the array is stored at 0 index.



### I. One-Dimensional Arrays

To create an array, you first must create an array variable of the desired type. The general form of a one-dimensional array declaration is

```
type array-var[];  
array-var = new type[size];
```

- declaration

- instantiation

*array-var* is the array variable that is linked to the array

The elements in the array allocated by **new** will automatically be

initialized to zero

Example:

```
int month_days[];  
month_days = new int[12];
```

Or

```
int month_days[] = new int[12];
```

Or

```
int a[] = {33,3,4,5};
```

*Example1:*

```
class Array {  
    public static void main(String args[]) {  
        int month_days[]; //Declaration  
        month_days = new int[12]; // instantiation  
        month_days[0] = 31; // initialization  
        month_days[1] = 28;  
        month_days[2] = 31;  
        month_days[3] = 30;  
        month_days[4] = 31;  
        month_days[5] = 30;  
        month_days[6] = 31;  
        month_days[7] = 31;  
        month_days[8] = 30;  
        month_days[9] = 31;  
        month_days[10] = 30;  
        month_days[11] = 31;
```

```

        System.out.println("April has " + month_days[3] + " days.");
    }
}

```

Output:

```

C:\Java>javac Array.java
C:\Java>java Array
April has 30 days.

```

*Example2:*

```

class Testarray1{
public static void main(String args[]){

int a[]={33,3,4,5}//declaration, instantiation and initialization

//printing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);

}}

```

Output

```

C:\Java>javac Testarray1.java
C:\Java>java Testarray1
33
3
4
5

```

## II. Multidimensional Arrays

Data is stored in row and column based index (also known as matrix form). *multidimensional arrays* are actually arrays of arrays

```
int arr[][] =new int[3][4];//3 row and 4 column
```

Example:

```

// Demonstrate a two-dimensional array.
class TwoDArray {
public static void main(String args[]) {
int twoD[][]= new int[4][5];
int i, j, k = 0;
for(i=0; i<4; i++)
for(j=0; j<5; j++) {
twoD[i][j] = k;
k++;
}

```

```

for(i=0; i<4; i++) {
    for(j=0; j<5; j++)
        System.out.print(twoD[i][j] + " ");
    System.out.println();
}
}
}
}

```

**Output:**

```

C:\Java>javac TwoDArray.java

C:\Java>java TwoDArray
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19

```

**Classes And Method**

The class is at the core of Java. It is the logical construct upon which the entire Java language is built because it defines an object. The class forms the basis for object-oriented programming in Java. Thus, a class is a *template* for an object, and an object is an *instance* of a class A class is declared by use of the **class** keyword. A simplified general form of a class definition is shown here

```

class classname {
    type instance-variable1;
    type instance-variable2;
    // ...
    type instance-variableN;
    type methodname1(parameter-list) {
        // body of method
    }
    type methodname2(parameter-list) {
        // body of method
    }
    // ...
    type methodnameN(parameter-list) {
        // body of method
    }
}

```

The methods and variables defined within a class are called *members* of the class. All methods have the same general form as **main( )**, most methods will not be specified as **static** or **public**. Java classes do not need to have a **main( )** method. You only specify one if that class is the starting point for your program. Further, applets don't require a **main( )** method at all.

**Example:**

```

class Box {
    double width;
}

```

```
    double height;
    double depth;
}
```

a class defines a new type of data the new data type is called **Box**. You will use this name to declare **objects** of type **Box**. you will use a statement like the following:

```
Box mybox = new Box(); // create a Box object called mybox
```

Every Box object will contain its own copies of the instance variables width, height, and depth. To access these variables, you will use the dot (.) operator.

```
mybox.width = 100;
```

Program:

```
/* A program that uses the Box class.
Call this file BoxDemo.java
*/
class Box {
    double width;
    double height;
    double depth;
}
// This class declares an object of type Box.
class BoxDemo {
    public static void main(String args[]) {
        Box mybox = new Box();
        double vol;
        // assign values to mybox's instance variables
        mybox.width = 10;
        mybox.height = 20;
        mybox.depth = 15;
        // compute volume of box
        vol = mybox.width * mybox.height * mybox.depth;
        System.out.println("Volume is " + vol);
    }
}
```

Output

```
C:\Java>javac BoxDemo.java
C:\Java>java BoxDemo
Volume is 3000.0
```

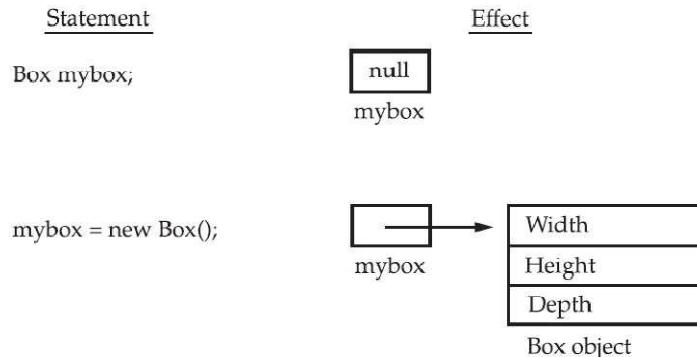
---

**Declaration of Object:** The **new** operator dynamically allocates (that is, allocates at run time) memory for an object and returns a reference to it.

```
Box mybox = new Box(); // declare reference to object and allocate a Box object
```

Or

```
Box mybox; // declare reference to object
mybox = new Box(); // allocate a Box object
```



### **Program:**

```
// This program uses a parameterized method.
class Box {
    double width;
    double height;
    double depth;
    // compute and return volume
    double volume() {
        return width * height * depth;
    }
    // sets dimensions of box
    void setDim(double w, double h, double d) {
        width = w;
        height = h;
        depth = d;
    }
}
class BoxDemo5 {
    public static void main(String args[]) {
        Box mybox1 = new Box();
        Box mybox2 = new Box();
        double vol;
        // initialize each box
        mybox1.setDim(10, 20, 15);
        mybox2.setDim(3, 6, 9);
    }
}
```

```
// get volume of first box
vol = mybox1.volume();
System.out.println("Volume is " + vol);
// get volume of second box
vol = mybox2.volume();
System.out.println("Volume is " + vol);
}
}
```

## Output

```
C:\Java>javac BoxDemo5.java
C:\Java>java BoxDemo5
Volume is 3000.0
Volume is 162.0
```

## Introducing Methods:

lasses usually consist of two things: instance variables and methods

This is the general form of a method:

```
type name(parameter-list) {
    // body of method
}
```

*type* specifies the type of data returned by the method

The *name* of the method is specified by name

The *parameter-list* is a sequence of type and identifier pairs separated by commas

Methods that have a return type other than void return a value to the calling routine using the following form of the return statement:

```
return value;
```

Example:

```
class Box {
    double width;
    double height;
    double depth;
    // display volume of a box
    void volume() {
        System.out.print("Volume is ");
        System.out.println(width * height * depth);
    }
}
```

```
}

class BoxDemo3
{
    public static void main(String args[]) {
        Box mybox1 = new Box();
        Box mybox2 = new Box();
        // assign values to mybox1's instance variables
        mybox1.width = 10;
        mybox1.height = 20;
        mybox1.depth = 15;
        /* assign different values to mybox2's
        instance variables */
        mybox2.width = 3;
        mybox2.height = 6;
        mybox2.depth = 9;
        // display volume of first box
        mybox1.volume();
        // display volume of second box
        mybox2.volume();
    }
}
```

#### Method That Takes Parameters

```
class Box {
    double width;
    double height;
    double depth;
    // compute and return volume
    double volume() {
        return width * height * depth;
    }
    // sets dimensions of box
    void setDim(double w, double h, double d) {
        width = w;
        height = h;
        depth = d;
    }
}

class BoxDemo5 {
    public static void main(String args[]) {
        Box mybox1 = new Box();
        Box mybox2 = new Box();
        double vol;
        // initialize each box
        mybox1.setDim(10, 20, 15);
        mybox2.setDim(3, 6, 9);
        // get volume of first box
        vol = mybox1.volume();
```

---

```
System.out.println("Volume is " + vol);
// get volume of second box
vol = mybox2.volume();
System.out.println("Volume is " + vol);
}
}
```

## Inheritance in Java

**Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes.

### Terms used in Inheritance

- o **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- o **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- o **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- o **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

### The syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```

The **extends keyword** indicates that you are making a new class that derives from an existing class

- **Single Inheritance Example**

```
class Animal{
void eat()
{
System.out.println("eating...");}
}

class Dog extends Animal{
void bark()
{
System.out.println("barking...");}
}

class TestInheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark();
d.eat();
}
}
```

- **Multilevel Inheritance Example**

```
class Animal{
void eat(){System.out.println("eating...");}
}

class Dog extends Animal{
void bark(){System.out.println("barking...");}
}

class BabyDog extends Dog{
void weep(){System.out.println("weeping...");}
}

class TestInheritance2{
public static void main(String args[]){
BabyDog d=new BabyDog();
d.weep();
d.bark();
d.eat();
}
}
```

---

- **Hierarchical Inheritance Example**

```
class Animal{  
    void eat(){System.out.println("eating...");}  
}  
class Dog extends Animal{  
    void bark(){System.out.println("barking...");}  
}  
class Cat extends Animal{  
    void meow(){System.out.println("meowing...");}  
}  
class TestInheritance3{  
    public static void main(String args[]){  
        Cat c=new Cat();  
        c.meow();  
        c.eat();  
        //c.bark()//C.T.Error  
    }  
}
```

# Summary

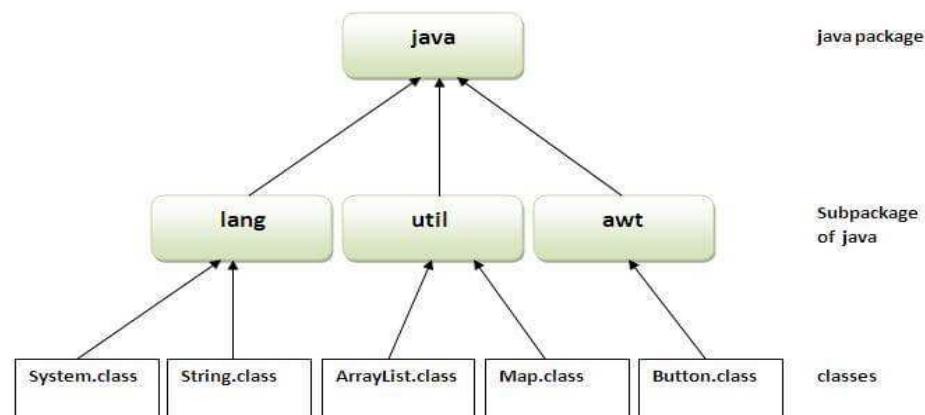
- Packages Introduction
- Advantage
- Defining a Package
- Package Example
- Importing Packages
- Access Specifiers

# Packages Introduction

- A java package is a group of similar types of classes, interfaces and sub-packages.
- Package in java can be categorized in two form,
  - built-in package and
  - user-defined package
- built-in packages such as java,lang, awt, javax, swing, net, io, util, sql etc.

# Advantage

- Advantage of Java Package
  - Java package is used to categorize the classes and interfaces so that they can be easily maintained.
  - Java package provides access protection.
  - Java package removes naming collision.



# Defining a Package

- Simply include a package command as the first statement in a Java source file. This is the general form of the package statement:

*package pkg;*

- Eg: *package MyPack;*
- You can create a hierarchy of packages For example, a package declared as

*package java.awt.image;*

- In order for a program to find MyPack, one of three things must be true.
  - Either the program can be executed from a directory immediately above MyPack,
  - or the CLASSPATH must be set to include the path to MyPack,
  - or the -classpath option must specify the path to MyPack when the program is run via java.

# Package Example

```
// A simple package
package MyPack;
class Balance {
    String name;
    double bal;
    Balance(String n, double b) {
        name = n;
        bal = b;
    }
    void show() {
        if(bal<0)
            System.out.print("--> ");
        System.out.println(name + ": $" + bal);
    }
}
```

```
class AccountBalance {
    public static void main(String args[]) {
        Balance current[] = new Balance[3];
        current[0] = new Balance("K. J. Fielding",
                               123.23);
        current[1] = new Balance("Will Tell", 157.02);
        current[2] = new Balance("Tom Jackson", -
                               12.33);
        for(int i=0; i<3; i++) current[i].show();
    }
}
```

# Importing Packages

- There are three ways to Import package.
  - import package.\*;
  - import package.classname;
  - fully qualified name.

```
// save by A.java
package Pack;
public class A{
    public void
msg(){System.out.println("Hello");}
}
```

```
// save by B.java
import Pack.*;
class B{
    public static void main(String args[]){
        A obj = new A();
        obj.msg();
    }
}
```

# Access Specifiers

TABLE 9-1  
Class Member Access

	Private	No Modifier	Protected	Public
Same class	Yes	Yes	Yes	Yes
Same package subclass	No	Yes	Yes	Yes
Same package non-subclass	No	Yes	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

# Summary

- **Introduction**
- **Why interface**
- **Defining a Interfaces**
- **Example**

# Introduction

- An **interface in java** is a blueprint of a class. It has static constants and abstract methods.
- Interface in Java is *a mechanism to achieve abstraction*. There can be only abstract methods in the Java interface, not method body.
- It is used to achieve abstraction and multiple inheritance in Java
- Interfaces are syntactically similar to classes, but they lack instance variables, and their methods are declared without any body

# Why interface

- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.
- This is the general form of an interface:

```
access interface name {  
    return-type method-name1(parameter-list);  
    return-type method-name2(parameter-list);  
    type final-varname1 = value;  
    type final-varname2 = value;  
    // ...  
    return-type method-nameN(parameter-list);  
    type final-varnameN = value;  
}
```

# Example

```
interface printable{  
    void print();  
}  
  
class A6 implements printable {  
    public void print() {System.out.println("Hello");}  
    public void show() {System.out.println("How r you");}  
    public static void main(String args[]){  
        printable obj=new A6();  
        A6 obj1 = new A6();  
        obj.print();  
        obj1.show();  
    }  
}
```

# Multiple inheritance in Java by interface

```
interface Printable{  
    void print();  
}  
interface Showable{  
    void show();  
}  
class A7 implements Printable,Showable{  
    public void print(){System.out.println("Hello");}  
    public void show(){System.out.println("Welcome");}  
    public static void main(String args[]){  
        A7 obj = new A7();  
        obj.print();  
        obj.show(); }  
}
```

# Summary

- **Introduction**
- **Exception-Handling**
- **Syntax**
- **Exception classes**
- **Types of Java Exceptions**
- **Examples**

# Introduction

- Exception is an unusual condition that can occur in the program
- This condition is cause due to run time error in the program.
- Exception handling mechanism allow the programmer to handle run time error situation gracefully without crashing the program.
- Eg divide by zero, array access out of bound etc.

```
Statement 1;  
Statement 2;  
Statement 3;  
Statement 4;  
Statement 5; //Exception Err0r  
Statement 6;  
Statement 7;  
Statement 8;
```

# Exception-Handling

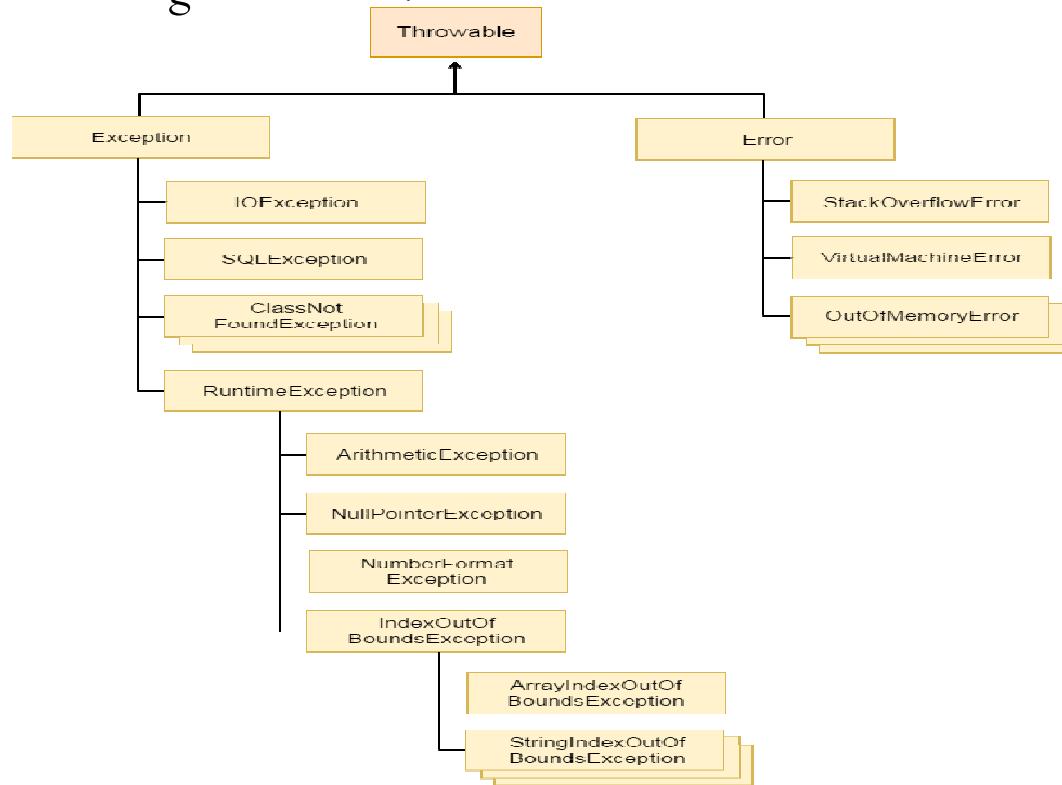
- Java exception handling is managed via five keywords: **try, catch, throw, throws, and finally**.
- **Try** :The "try" keyword is used to specify a block where we should place exception code.
- **Catch**: The "catch" block is used to handle the exception. It must be preceded by try block
- **Finally**: The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not.
- **Throw** :The "throw" keyword is used to throw an exception.
- **Throws**: The "throws" keyword is used to declare exceptions. It doesn't throw an exception.

# Syntax

```
try {  
    // block of code to monitor for errors  
}  
catch (ExceptionType1 exOb) {  
    // exception handler for ExceptionType1  
}  
catch (ExceptionType2 exOb) {  
    // exception handler for ExceptionType2  
}  
// ...  
finally {  
    // block of code to be executed after try block ends  
}
```

# A hierarchy of Java Exception classes

- The `java.lang.Throwable` class is the root class of Java Exception hierarchy which is inherited by two subclasses: `Exception` and `Error`. A hierarchy of Java Exception classes are given below:



# Types of Java Exceptions

- There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:
- Checked Exception: The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.
- Unchecked Exception: The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.
- Error: Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

# Examples

```
public class JavaExceptionExample{  
    public static void main(String args[]){  
        try{  
            //code that may raise exception  
            int data=100/0;  
        }catch(ArithmaticException e){System.out.println(e);}  
        //rest code of the program  
        System.out.println("rest of the code...");  
    }  
}
```

# Examples

```
class JavaException {
    public static void main(String args[]) {
        int d = 0;
        int n = 20;
        try {
            int fraction = n / d;
            System.out.println("This line will not be Executed");
        }
        catch (ArithmaticException e) {
            System.out.println("In the catch Block due to Exception = " + e);
        }
        System.out.println("End Of Main");
    }
}
```

# Summary

- **Introduction**
- **Advantage**
- **Life cycle of a Thread**
- **Creating a Thread**
- **Examples**

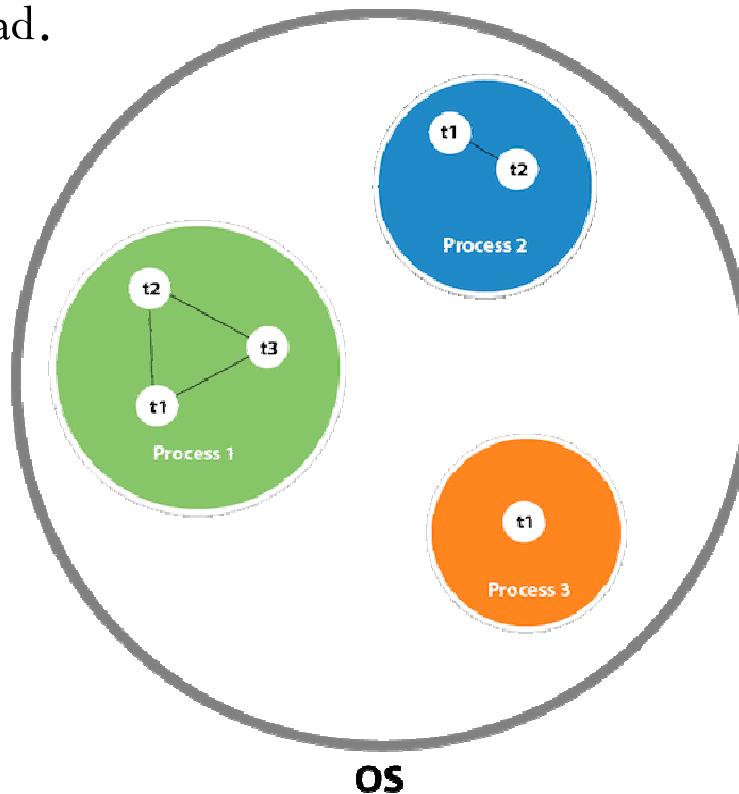
# Introduction

- **Multithreading in java** is a process of executing multiple threads simultaneously.
- A thread is a lightweight sub-process, the smallest unit of processing.
- Multiprocessing and multithreading, both are used to achieve multitasking.

Multitasking	Multithreading	Multiprocessing
<ul style="list-style-type: none"><li>• Capacity to run more than one process concurrently in a single CPU environment.</li><li>• Multitasking may be Process-based or Thread-based</li></ul>	<ul style="list-style-type: none"><li>• A single process that contains more than one thread (sub-processes) running simultaneously is called as a Multithread.</li><li>• It is a thread-based multitasking</li><li>• use a shared memory area</li></ul>	<ul style="list-style-type: none"><li>• It is much like multitasking doing with more than one CPU.</li><li>• It is a process-based multitasking</li><li>• separate memory area</li></ul>

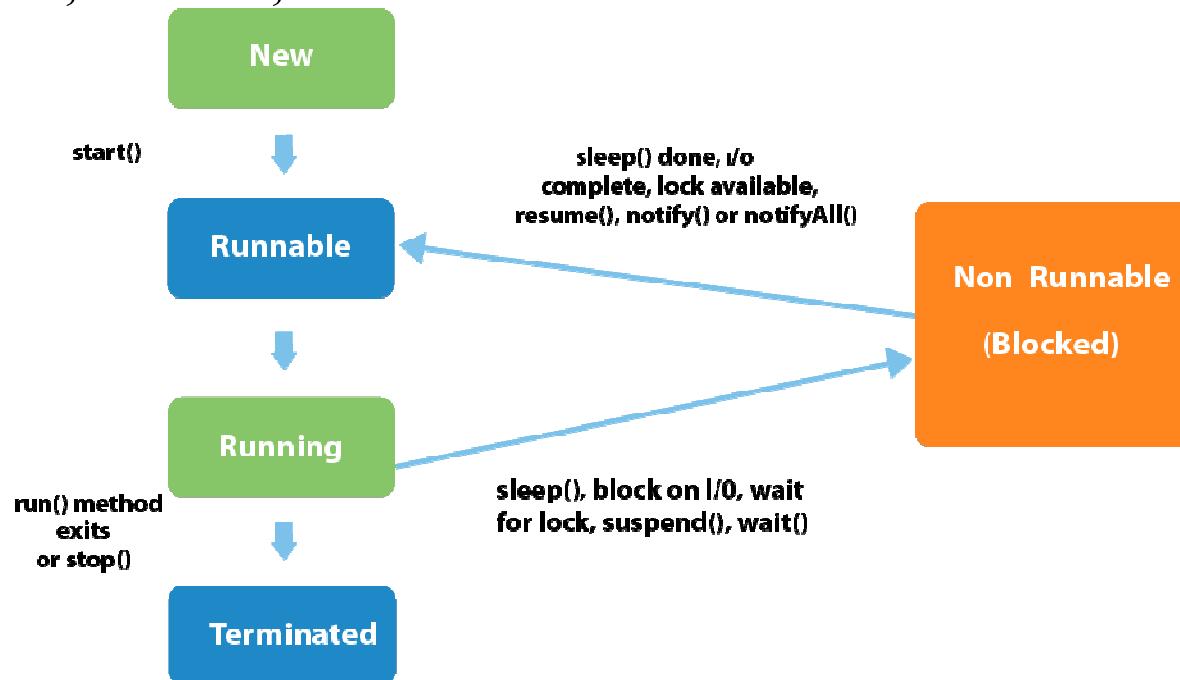
# Advantage

- **doesn't block the user** because threads are independent and you can perform multiple operations at the same time.
- You **can perform many operations together, so it saves time.**
- Threads are **independent**, so it doesn't affect other threads if an exception occurs in a single thread.



# Life cycle of a Thread

- A thread can be in one of the five states: *New, Runnable, Running, Non-Runnable (Blocked), Terminated.*
- According to sun, there is only 4 states in thread life cycle in java *new, runnable, non-runnable and terminated.*



# Creating Thread

- Threads can be created by using two mechanisms :
  - **Extending the Thread class**
  - **Implementing the Runnable Interface**
- Commonly used Constructors of Thread class:
  - Thread()
  - Thread(String name)
  - Thread(Runnable r)
  - Thread(Runnable r, String name)
- Commonly used methods of Thread class:
  - public void run()
  - public void start()
  - public void sleep(long miliseconds)
  - public int getId()
  - public void resume()
  - public void stop()
  - public String getName()

# Extending the Thread class

- The Class which extends the *thread* class will become a sub class.
- The thread class is contained in the package *java.lang.Thread*
  - STEP 1: Create a class by extending a Thread class
  - STEP 2: Override the run() method of a Thread class and place it inside the class which extends the Thread class.
  - STEP 3: Create an object for the thread inside the main() method.
  - STEP 4: Invoke the start() method through the object of the Thread class to execute the run() method of a thread.

# Extending the Thread class

```
class Demo extends Thread  
{  
    public void run()  
    {  
        System.out.println("Thread started execution");  
        for(int i=1;i<=5;i++)  
        {  
            System.out.println("Multithreading");  
        }  
        System.out.println("Thread is completed");  
    }  
}
```

```
public class Firstthread  
{  
    public static void main(String args[])  
    {  
        Demo t1=new Demo();  
        Demo t2=new Demo();  
        t1.start();  
        t2.start();  
    }  
}
```

# Implementing the Runnable Interface

- The simplest way of creating a thread is by implementing a *Runnable* interface.
- At a time any number of threads can be created in a program using *Runnable* interface.
- We create a new class which implements *java.lang.Runnable*
  - STEP 1: As a first step, the thread should be created through implementing a *Runnable* interface.
  - STEP 2: Define a *run()* method inside the thread. The *run()* method can be used only after implementing the *Runnable* interface.
  - STEP 3: Place the *run()* method inside the class which implements a *Runnable* interface.
  - STEP 4: Next is to instantiate a *Thread* class inside the *main()* method and use that object to execute a thread.

# Implementing the Runnable Interface

## Program 10.2

```
class RT implements Runnable
{
    public void run()
    {
        System.out.println("Thread started its
execution");
        for(int i=1;i<=10;i++)
        {
            System.out.println(i+" * 6 = "+i*6);
        }
        System.out.println("Thread completed its
execution");
    }
    public class Firstthread
    {
        public static void main(String[] args)
        {
            RT r=new RT();
            Thread t=new Thread(r);
            t.start();
        }
    }
}
```

## Output 10.2

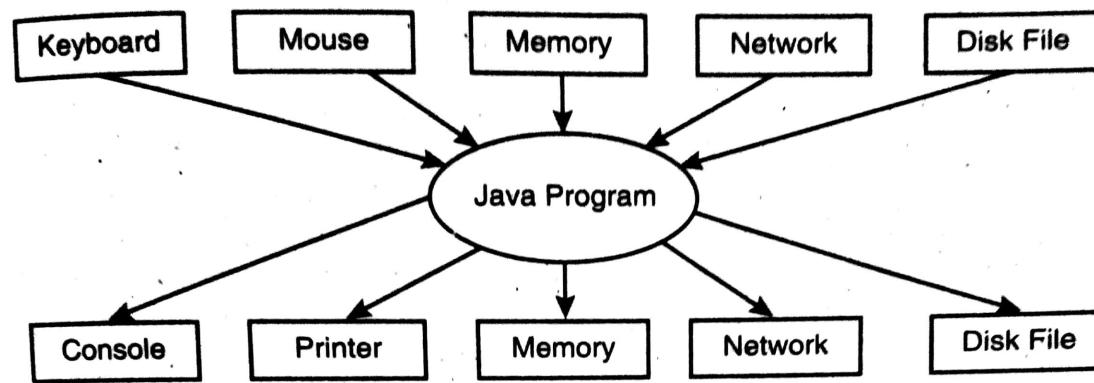
```
Thread started its
execution
1 * 6 = 6
2 * 6 = 12
3 * 6 = 18
4 * 6 = 24
5 * 6 = 30
6 * 6 = 36
7 * 6 = 42
8 * 6 = 48
9 * 6 = 54
10 * 6 = 60
Thread completed its
execution
```

# Summary

- **Introduction**
- **Stream**
- **Java Stream Classes**
- **Examples**

# Introduction

- Java I/O (Input and Output) is used *to process the input and produce the output.*
- Java uses the concept of a *stream* to make I/O operation fast.
- The `java.io` package contains all the classes required for input and output operations.
- We can perform **file handling in Java** by Java I/O API.

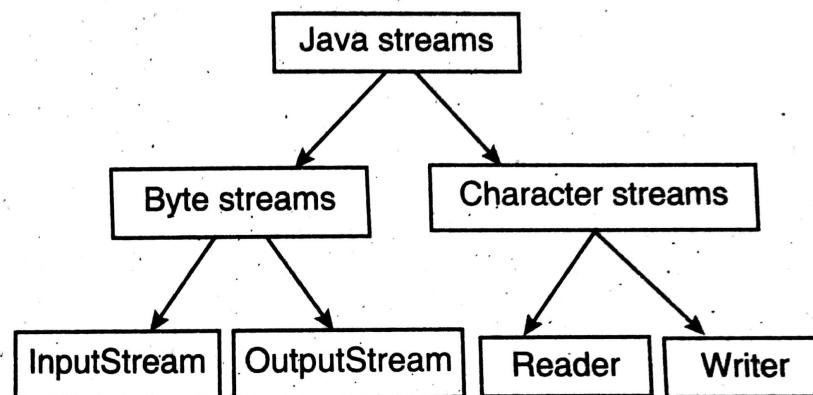


# Stream

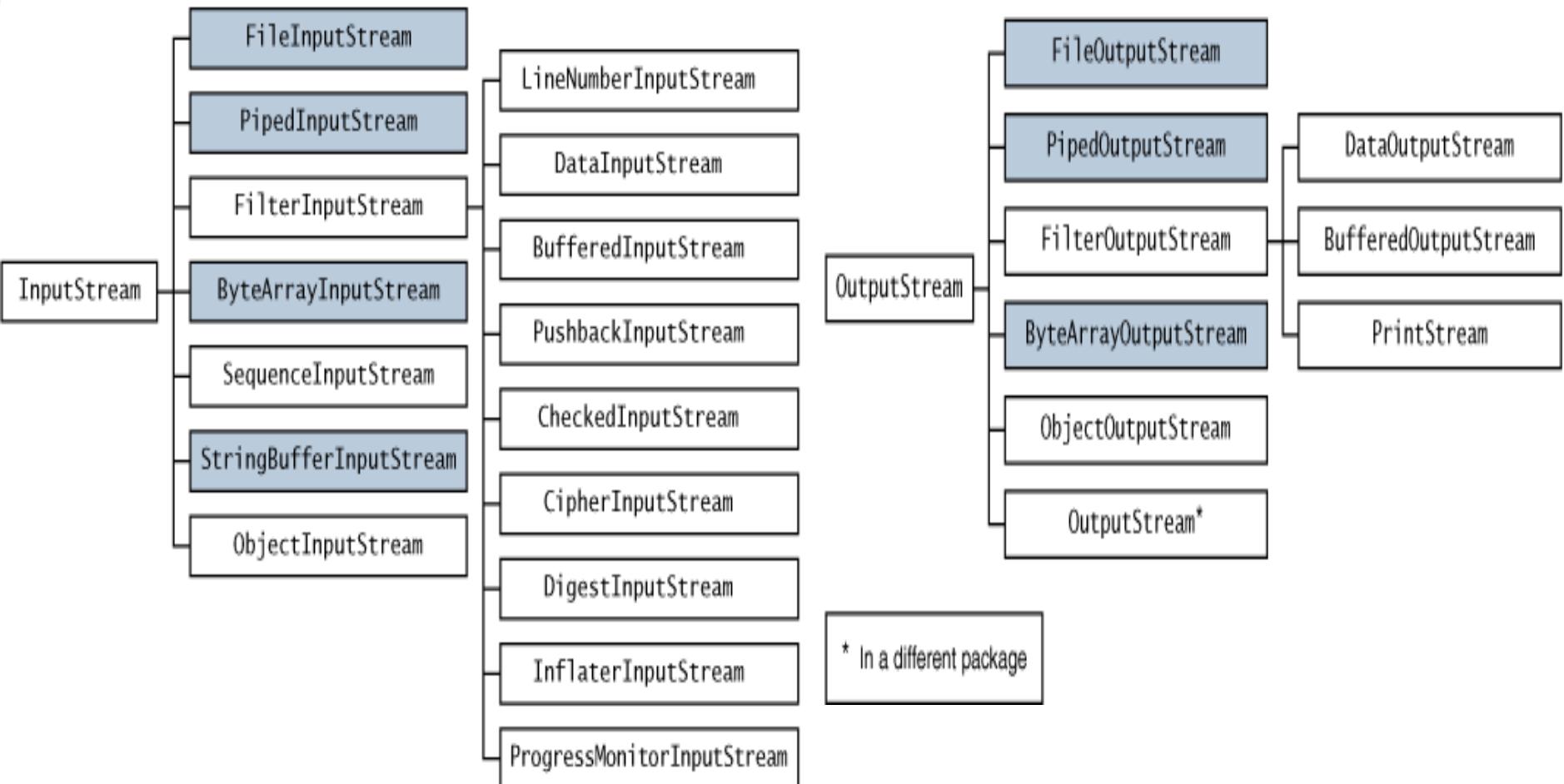
- A Stream is an ordered sequence of bytes that has an input stream and an output stream.
- In Java, 3 streams are created for us automatically. All these streams are attached with the console.
  - **System.out:** standard output stream
  - **System.in:** standard input stream
  - **System.err:** standard error stream
- Java implements *streams* within class hierarchies defined in the **java.io package**.

# Java Stream Classes

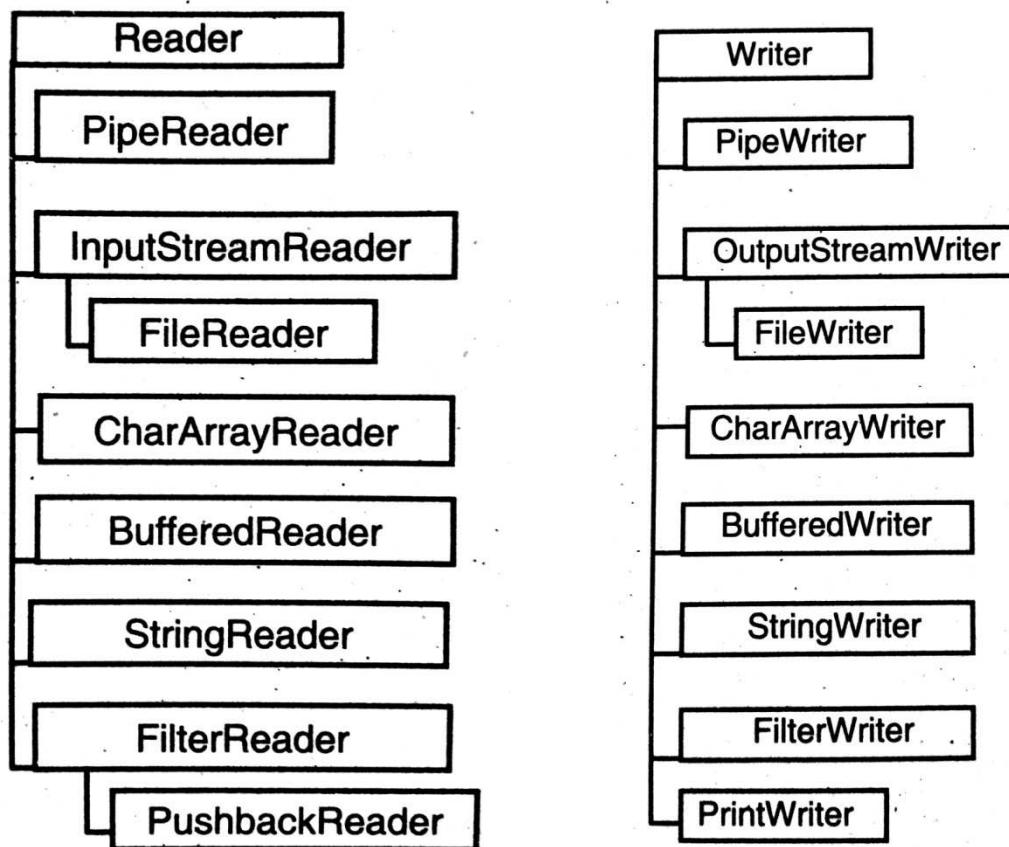
- Java defines two types of streams
  - **Byte Streams** : provide a convenient means for handling input and output of bytes. Byte streams are used, for example, when reading or writing binary data.
  - **Character Streams** : provide a convenient means for handling input and output of characters. They use Unicode.



# Byte Stream Classes



# Character Streams Classes



# Byte Stream Example

```
import java.io.FileOutputStream;
public class FileOutputStreamExample {
    public static void main(String args[]){
        try{
            FileOutputStream fout=new FileOutputStream("D:\\testout.txt");
            fout.write(65);
            fout.close();
            System.out.println("success...");
        }catch(Exception e){System.out.println(e);}
    }
}
```

# Byte Stream Example

```
import java.io.FileOutputStream;
public class FileOutputStreamExample1 {
    public static void main(String args[]){
        try{
            FileOutputStream fout=new FileOutputStream("D:\\testout1.txt");
            String s="Welcome to java.";
            byte b[]={s.getBytes()}; //converting string into byte array
            fout.write(b);
            fout.close();
            System.out.println("success...");
        }catch(Exception e){System.out.println(e);}
    }
}
```

# Byte Stream Example

```
import java.io.FileInputStream;
public class DataStreamExample {
    public static void main(String args[]){
        try{
            FileInputStream fin=new FileInputStream("D:\\testout.txt");
            int i=fin.read();
            System.out.print((char)i);

            fin.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

# Character Streams Examples

```
import java.io.*;
public class WriterExample {
    public static void main(String[] args) {
        try {
            Writer w = new FileWriter("D:\\testout2.txt");
            String content = "I love my country";
            w.write(content);
            w.close();
            System.out.println("Done");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

# Character Streams Examples

```
import java.io.*;
public class ReaderExample {
    public static void main(String[] args) {
        try {
            Reader reader = new FileReader("D:\\testout2.txt");
            int data = reader.read();
            while (data != -1) {
                System.out.print((char) data);
                data = reader.read();
            }
            reader.close();
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

# Summary

- **Introduction**
- **Advantage**
- **Lifecycle of Java Applet**
- **Example**
- **Importing Packages**
- **Access Specifiers**

# Introduction

- Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.
- applet does not have a **main( ) method**
- An **applet** is a Java program that runs in a Java-compatible Web browser or standard tool, **appletviewer**.
- This applet begins with two **import statements**
  - import java.awt.\*; // Abstract Window Toolkit
  - import java.applet.\*;

# Advantage

- There are many advantages of applet. They are as follows:
  - It works at client side so less response time.
  - Secured
  - It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.
- Drawback of Applet
  - Plugin is required at client browser to execute applet.

# Lifecycle of Java Applet

## **java.applet.Applet**

- **Applet is initialized:** `public void init()` is used to initialize the Applet. It is invoked only once
- **Applet is started:** `public void start()` is invoked after the `init()` method or browser is maximized. It is used to start the Applet.
- **Applet is stopped:** `public void stop()`: is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
- **Applet is destroyed:** `public void destroy()`: is used to destroy the Applet. It is invoked only once.

## **java.awt.Component**

- **Applet is painted:** `public void paint(Graphics g)` invoked immediately after the `start()` method, and also any time the applet needs to repaint itself in the browser

# Applet Example

- Simple example of Applet by html file:

```
//First.java  
import java.applet.Applet;  
import java.awt.Graphics;  
public class First extends Applet{  
  
public void paint(Graphics g){  
g.drawString("welcome",150,150);  
}  
}
```

```
<html>  
<body>  
<applet code="First.class" width="300" hei  
ght="300">  
</applet>  
</body>  
</html>
```

- To execute the applet by html file, create an applet and compile it.
- After that create an html file and place the applet code in html file

```
c:\>javac First.java  
c:\>appletviewer myapplet.html
```

# Applet Example

- Simple example of Applet by appletviewer tool:

```
//Second.java
import java.applet.Applet;
import java.awt.Graphics;
public class Second extends Applet{
    public void paint(Graphics g){
        g.drawString("welcome to applet",150,150);
    }
}
/*
<applet code="Second.class" width="300" height="300">
</applet>
*/
```

**c:\>javac Second.java  
c:\>appletviewer Second.java**

# Summary

- **Introduction**
- **Components of Event Handling**
- **Types of Event Handling**
- **Examples**

# Introduction

- Event describes the change in state of any object. It interrupt the current ongoing activity. Eg. When user clicks mouse or press keyboard during some processing, then it generate an activity.
- Event represent all activity that carried out between user and application.
- Java abstract window toolkit(AWT) conveys these message to the program.
- Events are supported by a number of Java packages, like **java.util**, **java.awt** and **java.awt.event**.

# Components of Event Handling

- Event handling has three main components,
  - **Events** : An event is a change in state of an object.
  - **Events Source** : Event source is an object that generates an event.
  - **Listeners** : A listener is an object that listens to the event. A listener gets notified when an event occurs.

# Types of Event Handling

- There are two commonly used events :
  - **Mouse Event** : While handling the mouse event we use two interface **MouseListener** and **MouseMotionListener**. These interface has certain methods such as *mouseClicked()*, *mousePressed()*, *mouseReleased()*, *mouseEntered()*, *mouseExited()*, *mouseDragged()* and *mouseMoved()*.
  - **Keyboard Event**: When Key from keyboard is pressed then it caused an event. There are three commonly method from **keyListener** interface and those are *keyPressed()*, *keyReleased()* and *keyTyped()*.

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="MouseEvents" width=300
height=300>
</applet>
*/
public class MouseEvents extends Applet
implements MouseListener,
MouseMotionListener {
String msg = "";
int mouseX = 0, mouseY = 0; // coordinates
of mouse
public void init() {
addMouseListener(this);
addMouseMotionListener(this);
}
// Handle mouse clicked.
public void mouseClicked(MouseEvent me) {
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}

```

```

// Handle mouse entered.
public void mouseEntered(MouseEvent me) {
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
// Handle mouse exited.
public void mouseExited(MouseEvent me) {
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
// Handle button pressed.
public void mousePressed(MouseEvent me) {
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
// Handle button released.
public void mouseReleased(MouseEvent me) {
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
// Display msg in applet window at
current X,Y location.
public void paint(Graphics g) {
g.drawString(msg, mouseX, mouseY);
}

```

```

// Demonstrate the key event handlers.

import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*
<applet code="SimpleKey" width=300
height=100>
</applet>
*/
public class SimpleKey extends Applet
implements KeyListener {
String msg = "";
int X = 10, Y = 20; // output coordinates
public void init() {
addKeyListener(this);
}

```

```

public void keyPressed(KeyEvent ke) {
showStatus("Key Down");
}
public void keyReleased(KeyEvent ke) {
showStatus("Key Up");
}
public void keyTyped(KeyEvent ke) {
msg += ke.getKeyChar();
repaint();
}
// Display keystrokes.
public void paint(Graphics g) {
g.drawString(msg, X, Y);
}
}

```

# Summary

- **Introduction**
- **String Methods**
- **Examples**

# Introduction

- String is a collection characters.
- String is basically an object that represents sequence of char values.
- An array of characters works same as Java string
- Example

```
char[] ch={'U','n','i','v','e','r','s','i','t','y';  
String s=new String(ch);  
OR  
String s="University";
```

# String Methods

Method	Description
s1.charAt(position)	Returns the character present at the index <i>position</i> .
s1.compareTo(s2)	If $s1 < s2$ then it returns positive, if $s1 > s2$ then it returns negative and if $s1 = s2$ then it returns zero.
s1.concat(s2)	It returns the concatenated string of s1 and s2.
s1.equals(s2)	If s1 and s2 are both equal then it returns true.
s1.equalsIgnoreCase(s2)	By ignoring case, if s1 and s2 are equal then it returns true.
s1.indexOf('c')	It returns the first occurrence of character 'c' in the string s1.
s1.indexOf('c',n)	It returns the position of 'c' that occur at after nth position in string s1.
s1.length()	It gives the length of string s1.
String.valueOf(var)	Converts the value of the variable passed to it into String type.

# Example

```
public class StringExample{  
    public static void main(String args[]){  
        String s1="java";//creating string by java string literal  
        char ch[]={'s','t','r','i','n','g','s'};  
        String s2=new String(ch);//converting char array to string  
        String s3=new String("example");//creating java string by new keyword  
        System.out.println(s1);  
        System.out.println(s2);  
        System.out.println(s3);  
        int x=s2.length();  
        System.out.println(x);  
        boolean b=s1.equals(s2);  
        System.out.println(b);  
        s2=s2.concat(s3);  
        System.out.println(s2);  
    }  
}
```

# Summary

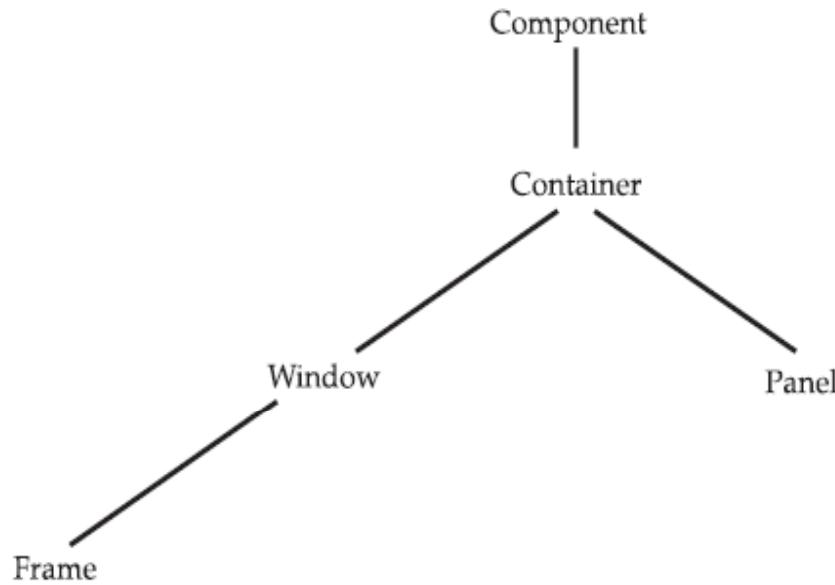
- **Introduction to AWT**
- **AWT Control**
- **AWT Layout Managers**
- **Examples**

# Introduction

- The Graphics programming in java is supported by AWT package.
- The AWT contains large number of classes which help to include various graphical component in the java program.
- These components are text box,button,labels, radio button, list item and so on.
- We need to import **java.awt** package. It is one of Java's largest packages.
- These classes are arranged in hierarchical manner.

# Window Fundamentals

- The two most common windows are those
  - derived from **Panel**, which is used by applets,
  - derived from **Frame**, which creates a standard application window.



# Window Fundamentals

- **Component:** It defines over a hundred public methods that are responsible for managing events, such as mouse and keyboard input, positioning and sizing the window, and repainting. A **Component object** is responsible for remembering the current foreground and background colors
- **Container:** The Container class is a subclass of Component. It has additional methods that allow other Component objects to be nested within it. Other Container objects can be stored inside of a Container. A container is responsible for layout and placement of graphical component.
- **Window:** the top level window without border and without menu bar is created using **window class**.
- **Panel:** The Panel class is a subclass of Container. It similar to Window without border and without any menu bar , title bar.
- **Frame:** it is top level window with border and menu bar. It support the common window event such as window open, close.

# AWT Control

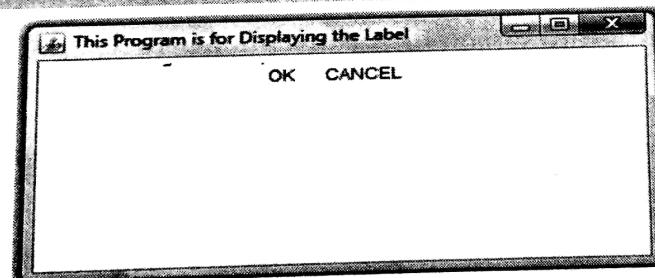
- Label
- Button
- Text Box
- Check Box
- Choice

## Java Program[Use\_Label.java]

```
import java.awt.*;
class Use_Label
{
    public static void main(String[] args)
    {
        Frame fr=new Frame("This Program is for Displaying the Label");
        fr.setSize(400,200);
        fr.setLayout(new FlowLayout());
        fr.setVisible(true);
        Label L1=new Label("OK");
        Label L2=new Label("CANCEL");
        fr.add(L1);
        fr.add(L2);
    }
}
```

## Output

```
C:\>javac Use_label.java
C:\>java Use_label
```



# Button

- Button

```
Button (String s)
```

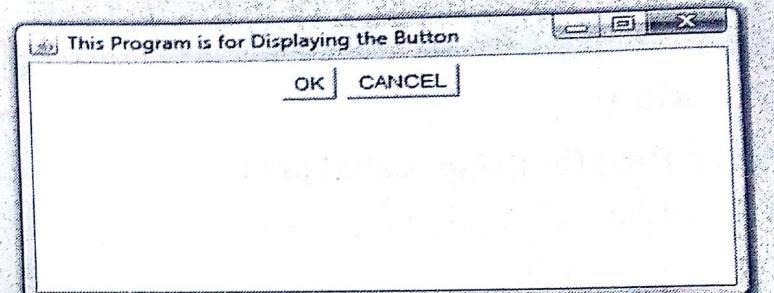
## Java Program

```
import java.awt.*;
class Use_Button
{
    public static void main(String[] args)
    {
```

```
        Frame fr=new Frame("This Program is for Displaying the Button");
        fr.setSize(400,200);
```

```
        fr.setLayout(new FlowLayout());
        fr.setVisible(true);
        Button B1 = new Button("OK");
        Button B2 = new Button("CANCEL");
        fr.add(B1);
        fr.add(B2);
    }
}
```

## Output

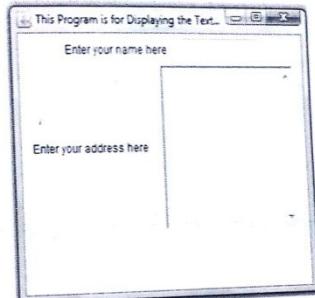


# Text Box

Java Program[Use\_TxtFld.java]

```
import java.awt.*;
class Use_TxtFld
{
    public static void main(String[] args)
    {
        int i;
        Frame fr=new Frame("This Program is for Displaying the TextField");
        fr.setSize(350,300);
        fr.setLayout(new FlowLayout());
        fr.setVisible(true);
        Label L1=new Label("Enter your name here");
        TextField input1=new TextField(10);
        Label L2=new Label("Enter your address here");
        TextArea input2=new TextArea(10,20);
        fr.add(L1);
        fr.add(input1);
        fr.add(L2);
        fr.add(input2);
    }
}
```

Output



# Check Box

**Checkbox(String label)**

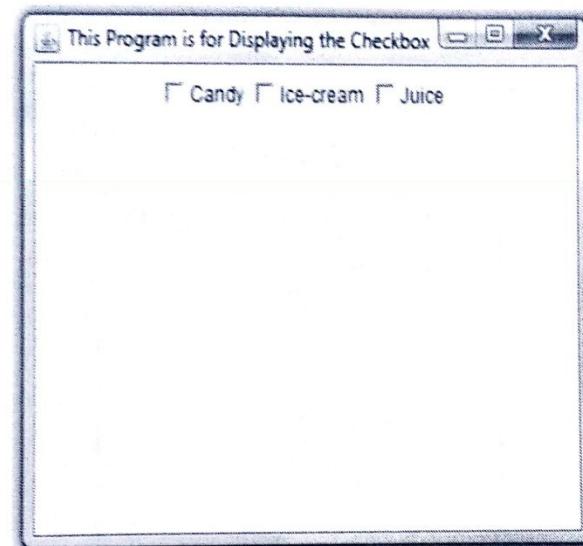
where *label* denotes the label associated with each checkbox.

To get the state of the checkbox the **getState()** method can be used.

**Java Program[Use\_ChkBox.java]**

```
import java.awt.*;
class Use_ChkBox
{
    public static void main(String[] args)
    {
        int i;
        Frame fr=new Frame("This Program is for Displaying the Checkbox");
        fr.setSize(350,300);
        fr.setLayout(new FlowLayout());
        fr.setVisible(true);
        Checkbox box1=new Checkbox("Candy");
        Checkbox box2=new Checkbox("Ice-cream");
        Checkbox box3=new Checkbox("Juice");
        fr.add(box1);
        fr.add(box2);
        fr.add(box3);
    }
}
```

**Output**



# Choice

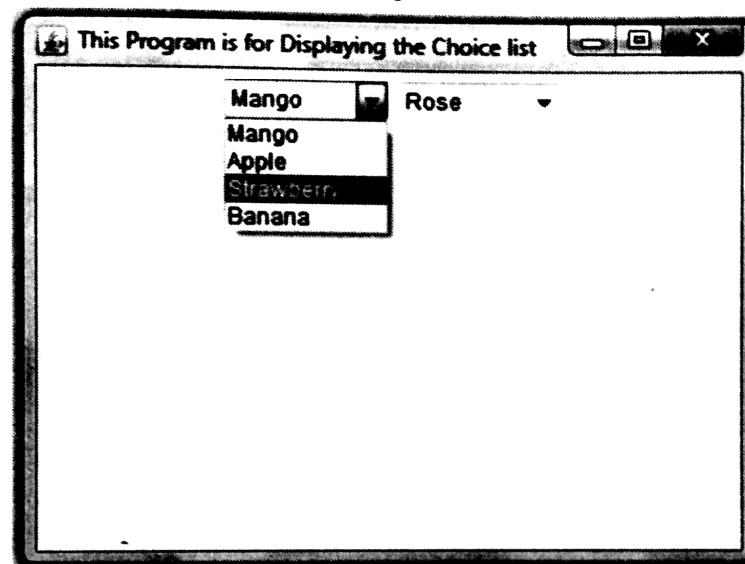
```
Choice obj=new Choice();
```

## Java Program[Use\_ChkBox.java]

```
import java.awt.*;
class Use_ChkBox
{
    public static void main(String[] args)
    {
        int i;
        Frame fr=new Frame("This Program is for Displaying the Choice list");
        fr.setSize(350,300);
        fr.setLayout(new FlowLayout());
        fr.setVisible(true);
        Choice c1=new Choice();
        Choice c2=new Choice();
        c1.add("Mango");
        c1.add("Apple");
        c1.add("Strawberry");
        c1.add("Banana");

        c2.add("Rose");
        c2.add("Lily");
        c2.add("Lotus");
        fr.add(c1);
        fr.add(c2);
    }
}
```

## Output



# AWT Layout Managers

- **FlowLayout:** FlowLayout is the default layout manager. FlowLayout implements a simple layout style, which is similar to how words flow in a text editor. The direction of the layout is governed by the container's component orientation property, which, by default, is left to right, top to bottom
- **BorderLayout:** The BorderLayout class implements a common layout style for top-level windows. It has four narrow, fixed-width components at the edges and one large area in the center. The four sides are referred to as north south, east, and west. The middle area is called the center.