Date.....

Web Technologies (BCS502)

Introduction :- Introduction and Web Development Strategies, History of Web and Internet, Protocal Governing Web, Writing Web Projects, Connecting to Internet, Introduction to Internet Services and tools, Introduction to client-server computing. Web Page Designing : HTML : List, Table, Images, Frames, forms, XML : Document type definition (DTD), XML schemes, Objects Model, presenting and using XML, Using XML Processors: DOM and SAX.

CSS :- Creating style sheet, CSS Properties, CSS Styling (Background, Text format, Controlling Fonts), Working with Lists and Tables, CSS ID and Class, Box Model (Introduction, Border properties, Padding properties, Margin properties).

CSS Advanced : (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image sprites, Attribute selector), CSS Color, Creating Page Layout and Site designs.

Scripting :- Java script, Introduction, documents, forms, statements, functions, objects, introduction to AJAX.

Networking : Internet Addressing, Inet Address, Factory Methods, Instance Methods, TCP/IP Client Sockets, URL,

Spiral

Teacher's Sign .....

 Date.....

, URL Connections , TCP/IP Server Sockets , Datagram.

(4) Enterprise Java Bean :- Creating a JavaBeans , JavaBeans Properties , Types of beans , Stateful Session bean , Stateless Session bean , Entity bean.

Node js : Introduction , Environment Setup , REPL Terminal , NPM (Node Package ~~Management~~ Manager) . Callback Concept, Events, Packaging , Express Framework , Restful API) .

Node.js with MongoDB : MongoDB Create Database , Create Collection , Insert , delete update , join , sort , query .

(5) Servlets : Servlet Overview and Architecture , Interface Servlet and the Servlet Life Cycle , Handling HTTP get Requests, Handling HTTP post Requests, Redirecting Requests to other Resources , Session ~~Tasking~~ . Tracking ; Cookies , Session Tracking with Http session .

Java Server Pages (JSP) : Introduction , Java Server Pages Overview , A First Java Server Page . Example , Implicit Objects , Scripting , Standard Actions , Directives , Custom Tag Libraries.

Date 10/08/18

UNIT-I

History of World Wide Web:-

Tim Berners Lee is known as the father of the world wide web. Tim Berners Lee was physi-ciast at the European organisation for Nuclear research CERN. A highly esteemed particle physics laboratory in Switzerland.

In early 1989, Tim Berners Lee proposed the web as a way for scientist around the world to collaboriate using a Global information system based on Hypertext

In the fall of 1990, the first text only browsers were set up and CERN scientist could access hyper text files and other information at CERN based on proposal by Tim Berners Lee, the structure of hypertext document was dyined by a new language called the hypertext markup language (HTML). HTML was based on a subset of the standard generalised markup language (SGML) already in wide use. To transfer document to remote sites, a new protocol was dvlped called http.

Protocols governing web.

Protocol.

A protocol is set of rules, that is used to communicate application to each other. In other word we can say that the protocol is the interface required for communicating the different applications.

There are different type of protocols that are used in web, these are as follows :-

In the fall of 1991, conference goes around the world ~~started~~ hearing about the promise and age of hypertext, but sparks still were not flying.

By early 1993, there were only about 50 worldwide websites.

Only browsers allowed to take advantage of the web's graphical capabilities were developed at National center for Super computing applications (NCSA) called the browser MOSAIC.

History of Internet

Floppy disk

Page 4
Date

1. HTTP (Hyper text transfer protocol)
2. ICMP (Internet control message protocol)
3. RIP (Routing Information protocol)
4. POP3 (Post office protocol)
5. UDP (User datagram Protocol)
6. TCP/IP (Transmission control protocol / Internet Protocol)
7. OSPF (Open Shortest path first) or Dynamic Protocol
8. IGRP (Interior Gateway routing protocol)
9. MIME (Multipurpose Internet Mail Extension)

1. HTTP

HTTP is an application layer protocol for distributed collaborative, hyper media information system. Its used for retrieving interlinked resources led to the establishment of the word wide web.

HTTP is a request/response standard between a client and a server. An HTTP client initiates a request, it establishes a TCP connection to a particular code on a host (port 80). An HTTP server listning on that port wait for the client to send a request message upon receiving the request the server send back a status line and a message of its phone the body of which is perhaps a request response or error

2. ICMP

ICMP is one of the core protocols of the Internet protocol suit. It is chiefly used by used by network computers operating system to send error message, indicating for instance, that are requested service is not available or that a host or router could not be reached. ICMP realise on IP to perform its task and it is an Integral part of IP

3. OSPF

OSPF is a dynamic routing protocol for used in Internet protocol (IP) networks, specifically, it is a linked state routing protocol and falls into a group of Interior gateway protocol and operating within an autonomous system. OSPF is perhaps the most widely used Interior gateway protocol in large Enterprise network.

OSPF built routing table basis on the destination. IP address found in IP packet. It was designed to support variable length subnet masking.

4. RIP

RIP is a dynamic routing protocol used in local and wide area networks. As such it is

Classified as an interior gateway protocol (IGP) using the distance vector routing algo.

5. IGRP

IGRP is a kind of IGP which is a distance vector routing protocol invented by CISCO, used by router to exchange routing data within an autonomous system

IGRP is a proprietary protocol. IGRP was created in part to overcome the limitations of RIP.

6. POP3

In computing the post office protocol version 3 is an application layer protocol used by the local email client to retrieve email from a remote server over a TCP/IP connection. POP3 and IMAP (Internet message access protocol) are the most prevalent Internet standard protocols for email retrieval

7. TCP/IP

TCP/IP is the set of communication protocol

ASCII (American standard code for information interchange.

In it. Transmission Control and Internet protocol, The IP suite like many protocol suits may be viewed as a set of layers. Each layer solves a set of problems involving the transmission of data and provides a well defined servic to the upper layer protocol based on using the services from the lower layer.

8. UDP

UDP is one of the core members of the Internet protocol shouldn, the set of network protocol used for the Internet with UDP computer applications can send messages, in the case refered to as data-gram, to other host on an Internet protocol network without requiring prior communi-cation to setup special transmission Chanel or data path. UDP provides an unreliable services and datagram may arrived out of order appear duplicated or go missing without notice.

9. MIME { Multipurpose Internet mail extension}

MIME is an Internet standard that extends the format of email to support.

1) Text in character set other than ASCII

2) Non text attachments.

3) Message body with multiple parts

4) Header information is non ASCII character set.

MIME's use however has grown beyond describing the content of email to describing content type in general including the web.

Webpage.

Webpage refers to nothing more than a world wide web document. A webpage is a document written in HTML form. It is used to build a website.

Website

A website is simply a collection of interlinked webpages. Webpages need to be related. By its varying nature web makes it possible to link any no. of items including other

pages.

Category of website :

These website could be classified into individual and corporate website. As per the application and scope of web projects, we need to have both of them.

Corporate type website :

In corporate type website there are certain no. of persons who develop there website for a particular organization. The corporate website are formed when the group of people have common interest and objective for which they come and then make use of website. The purpose of this type of website is to convey the information of organization to all over the world.

Individual website.

In the individual website is just like profile management system. In this type of website an individual want to develop website for h-projection, carreer growth etc.

Web development process.

1. Phase 1 - Strategy

In web development the developer first makes strategy how to develop a page or website for this he must decide.

a) Goals and objectives
b) Build the developing team.
c) Make the useful research related to problem.
d) Prepare a list of task and propose the project to his or her team for dveloping

2. Phase 2- Design and Specification

After making the proper streategies require to develop a proposed work, he/she must design the scheduled and prepare specifications

a) Developing concepts.
b) Plan the content require to use.
c) Make rough design.
d) Make final design if there are not major changes in rough design.
e) Build a prototype of object.

f) Test the prototype, if prototype is succeed then go to phase 3 otherwise repeat phase 2 until prototype is succeed.

Phase 3: Produce desired result

After the successful testing of Prototype the developer must proceed to develop the actual live web project i.e, requested by the client.

Phase 4. Testing and Maintinance.

After the completing the web project, the developer perform the testing checking the client's requirement if there is a mismatch found perform the appropriate action for removing the bugs occured in the testing.

Phase 5. Registered with ISP:-

After completing the phase 4, the resultant web project must be registered with their reasonal ISP to become his/her web project legal so that web project can get a valid domain space at ISP server.

Phase 6 Launch.

After getting registered with ISP the web

project is launched that is, it is publically accessible to the users of that particular domain.

Web team

Web teams begin with web masters to tell who did anything from coding the page to maintaining the web server. Web development efforts were often grass roots efforts within companies.

There are basically two types of web teams that are -

1) **Server Side :-** Server side web teams are web teams hired by a company to develop a website.

2) **Client side :-** Client side web teams are part of the company i.e, putting together the website. It is entirely possible that you will put together web teams comprised by both these subteams.

Assessment of web teams

A web team might be core, extended or

Special members, core means a member who is a expert in his work, special members of this team might be core. Extended team members are people whose skills might not always be necessary or who might have cross functional groups.

Core team members -
- Project Manager
- Technical Lead
- Web Production specialists
- Creative Lead
- Designers
- Production artist
- Quality assurance.

Extended team members -
- Account manager
- Programmer
- Network Engineer
- Information architects
- Copy writers
- Tester.

Special team members. -
- Security experts
- Audio Engineer
- Video Engineer
- 3-D Modelez
- Web cost specialist
- Media buyer
- Strategic planner

Tools for using the Internet

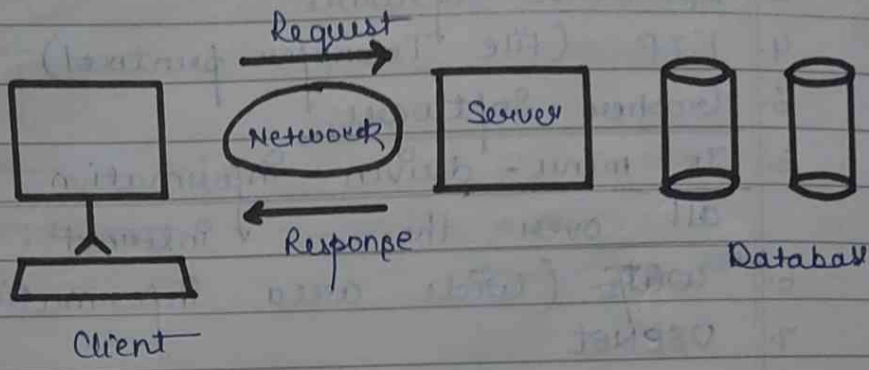
1. Email client
2. Telnet software
3. List Serve Software
4. FTP (File Transfer protocol)
5. Gopher Software
6. Is menu-driven information resources located all over the Internet.
6. WAIS (Wide area information Server)
7. USENet
8. Internet Relay chat (IRC)
9. WWW web browsers

Client Server Computing

Client Server model of computing is a distributed application structure that partitions task or work load b/w the providers of resource of service, called servers and service request called client often clients and Server communicate over a computer network on separate hardware but both client and server may reside in same system.

A server host one or more server programs which share their resources with clients. A client does not share any of its resources but request a servers

Content servic functions, client therefore initiate communication session with servers which await incomming request



Component of client server computing

A client server process usually manage the user interface portion of the application, validate data by the user dispatch request to server programs on it

On the other hand the server process fulfill the client request by performing the servic requested after the server receive request from client, it executes database retrieval, update and manages data integrity dispatch responses to client request

Middle ware allows application to transparently communicate with other programs or processes regardless of location.

UNIT - 2.

Creating list

1. Ordered list

Ordered list (OL) tag defines a sequentially numbered list of items. It is used in conjunction with the list item tag (LI).

```
<html>
<head>
<title> Ordered list </title>
</head>
<body>
<OL>
<LI> Mouse
<LI> Keyboard
<LI> Printer
<LI> CPU
</OL>
</body>
</html>
```

2) Unordered list

The unordered list or UL tag defines a bulleted list of items, the list item LI is nested inside the UL tag and defines

the each items within the list

```
<html>
<head>
<title> Unordered list </title>
</head>
<body>
<UL>
<LI> Mouse
<LI> Keyboard
<LI> Printer
<LI> CPU
</UL>
</body>
</html>
```

3) Nested list.

You can nest a list inside another list, the browser automatically indent nested list levels, you can nest the same or different kind of list.

```
<html>
<head>
<title> Nested list </title>
</head>
<body>
<UL>
<LI> Hardware
<OL>
<LI> Mouse
```

Page 20
Date

```
<LI> Keyboard
<LI> CPU
<LI> Monitor
</OL>
<LI> Software
<OL>
<LI> Device driver
<LI> Office tools
<LI> Windows
<LI> Browsers
</OL>
</OL>
</body>
</html>
```

Controlling how ordered list are displayed.

The type attribute allows you to specify the number type for an ordered list.

Specify the number type :-

```
<html>
<head>
<title> ordered list </title>
</head>
<body>
<OL>
```

```
<LI> Hardware
<OL type = "a">
<LI> CPU
<LI> Monitor
<LI> Keyboard
<LI> Mouse
</OL>
<LI> Software
<OL type = "disc">
<LI> device driver
<LI> office tool
<LI> windows
<LI> Browsers
</OL>
</UL>
</body>
</html>
```

Table

Whenever group information needs to be presented tabular representation is the best one and so is the case even on webpages tables are used

for arranging the layout of a webpage.

The content on the page look organized with the use of tables which adds to its visual appeal also

Basic table tab

<table> - - - - </table>

The table tag can have following attributes-

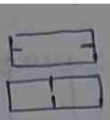
- Align
- Boreder { default 1,2,3- - }
- Cell padding (space b/w content and cell Vanshita)/default 1,2,
- Cell spacing ( default 1,2,3- - -)
- Width. (default 1,2,3- - -)

Ex 1.

```
<html>
<body>
<table>
<TR>
<TD>S:No.</TD>
<TD> Name of Boys </TD>
<TD> Name of Girls </TD>
</TR>
<TR>
<TD> 1 </TD>
<TD> Vaibhav </TD>
<TD> Vanshita </TD>
</TR>
<TR>
<TD> 2 </TD>
<TD> Ricky </TD>
<TD> Shubhi </TD>
</TR>
```

S:No	Name of Boys	Name of Girls
1	Vaibhav	Vanshita
2	Ricky	Shubhi

for width → colspan
height → rowspan



</table>
</body>
</html>

Ex 2

```
<html>
<body>
<table Align="Centre" Cell padding="2" Cell spacing="6"
       BG Color="green">
<TR>
<TH> ← <TD> S.No.</TD>
for bold. <TD> No. of Boys</TD>
<TD> No of Girls</TD>
</TR>
<TR>
<TD> 1 </TD>
<TD> Vaibhav </TD>
<TD BG Color="green" Align="left"> Vanshita </TD>
</TR>
<TR>
<TD> 2 </TD>
<TD> Ricky </TD>
<TD> Shubhi </TD>
</TR>
</table>
</body>
</html>
```

<P> paragraph change

24

Images.

The image <img> tab allows you to display image on your web page

```
<img src="abc.jpg">
```

we can use width directly

```
width="80%"
```

Forms.

A form film is a data entity field on a page. A user supplies information into a field either by typing text or by selecting a value from a list of dedicated value.

A form has a form handler which is a method of collecting information from the form

Applications areas for forms.

The application area for forms on www or forever involving, some typical areas where they can be used.

1) Educational websites
2) Online purchase
3) Collecting feedback
4) Providing interface for a chat session
5) Social media website

f) Banking
g) Registration websites
h) Jobs websites
i) Railway reservation

Creating a form

forms can be classified into three sections.

1) Form header 2) Input field
3) Action button

Syntax

```
<body>
  <form>
    Content
  </form>
</body>
```

Adding a password field.

```
<input type = "Password" Name = "salary",
       size = "10">
```

Multiple line text input

```
<text area row = "3"  column = "50"  name = "Remark">
</text area>
```

Pull down Menus :-

```
<Select Name = "list name" >
<option > option 1
<option > option 2
<option > option 3
</Select >
```

Adding submit and reset button :-

```
<input type = "Submit" Value = "Submit" >
<input type = "text" name = "student Name"
       size = "30" max length = "30"
       value = "default" >
```

Adding a check box :-

```
<input type = "Check box" value = "B. Tech" >
<input type = "Check box" value = "M. Tech" >
```

Adding Radio buttons :

```
<input type = "Radio" name = "Gender"
       value = "Male" checked >
```

CSS (Cascaded Style Sheet)

Style Sheet is a collection of formatting style, which can be applied to a Web page. The Style Sheet consists of following components:

Style rule :- A style rule is a set of html tags specifying the formatting elements. Style rules can be applied to selected contents of a Web page.

a) Selector : A selector is a string that identifies what elements the correspondig rule applies to and is the first part of the rule.

b) Declaration : This apart of the rule is enclosed within {} or curly brackets.

Basically there are four ways of incorporating style sheets in html document.

1) Including the information within HTML.
2) Embedding a style sheet

3) Linking to an externally style sheet
4) Importing a style sheet.

Frames.

Frames enables the user to divide a page into number of rectangular reasem window of various sizes. You can think of a frame as a window within another window. A page can have one or more frames

```
<html>

<frameset row="" col="">
Page content
</frameset>
</body>
</html>
```

1) Including style information within Html or in line

Inline style declaration is the most basic style rule, which can be applied to individual elements in the web page, in line styles are implemented by using style attributes with the html tags

Syntax

```
<html tag style = "Property : Value">
```

eg:

<P style = {colour : Olive} >

2) Embedding a style sheet :-

You can group more than one style rule.

```
<head>
<style>
<P {font-family : Arial} >
</style>
</head>
```

```
<head>
<style>
H1 { color : Limegreen }
H2 { font-family : Arial }
H3 { color : Cyan }
</style>
```

Grouping style rule.

```
<head>
<style>
H1, H2 { Color : red ; font-family : Arial }
</style>
</head>
```

3) Linking to an External style sheet :-

For constructing a css, first style rules must be written in a document and save in separate with an extension of css. The syntax for linking to an external style sheet is -

```
<html>
<head>
<link REL = stylesheet "href" = "Dictionary Path where the style sheet is saved">
</head>
<body>
</body>
</html>
```

4) Importing a style sheet.

Importing a style sheet automatically pulls the style rules into the document for use. Once imported, changes made to the style sheet will not be reflected in the web page into which it has been imported. This problem can be avoided by linking the style sheet to the main document rather than importing it.

```
<html>
<head>
<style type = "Text/css">
@ import URL(The path);
```

</style>
</head>
<body>
</html>

Style sheet property

1) Font property
   1. Font-family
   2. Font-size
   3. Font-style
   4. Font-weight

2) Text properties
   1. Letter spacing
   2. Word spacing
   3. Vertical-Align
   4. Text-Align
   5. Text-indent
   6. Text-transform
   7. Text-decorate

3) Colour and BG properties

4) Box properties:-
   Margin and Border.

5) Padding properties
   Cell padding

XML
(Extensible Markup Language)

XML stands for Extensible Markup Language
XML is a markup much like html.
XML was designed to describe data.
XML tags are not pre-defined, you must define your own tags.

XML is self describing.
XML uses a DTD (Document type definition) to formally describes the data

Difference b/w XML and HTML.

1) XML is not a replacement of html.
2) XML and html work design with different goods
3) XML was designed to describe data and to focus on what data is.

* HTML was designed to display data and to focus on how data looks.
* HTML is about displaying information, XML is about describing information

```
<? XML    Version = "1.0" ?>
<home>
<head>
<Title>
My home page
</Title>
<banner source = "abc.gif" />
</head>
<body>
<main.title>
Welcome to my home page
</main.title>
<rule />
<Text>
<Para>
Sorry, this home page is still under construction
</Para>
</Text>
</body>
<footer source : "abb.gif" />
</home>
```

XML Schema.

XML Schema is an XML based alternative to DTD. An XML Schema describes the structure of a XML document. The XML schema language is also referred to as XSD (XML schema definition)

An XML is a.

* Defines elements that can appear in a document.

* Defines attributes that can appear in a document.

* Defines which elements are child elements.
* Defines the order of the child element.
* Defines the no. of child element

XML Schema are the successor of DTDs :-
Very soon XML Schema will be used in most applications as a replacement of DTDs reasons

a) XML Schema's are extensible of future additions.
b) XML Schema's are richer and more powerful than DTDs
c) XML Schema's are written in XML.
d) XML Schema's support data types
e) XML Schema's support name spacing.

DTD (Document type Definition | Declaration)

The DTD describes a model of the structure of the content of an XML document. This model says what must be present which one are optional, what their attributes are and how they can be structured with relation to each other. While HTML has own link on DTD, XML allows you to create own DTD's for

your applications. This gives you complete control over the process of checking the content and structure of the XML document created for that application. This checking process starts validation

Document object Model (DOM)

The DOM is a W3C (Consortium) standard. DOM defines a standa-rd form for accessing documents like XML and HTML (world wide web)

"W3C document object model is a platform and language neutral interface that allows programs and script to dynamically access and update the content, structure and style of a document."

The document object model is separated into three different parts.

* Core DOM
* XML DOM
* HTML DOM

→ Core DOM

Standard model for any structured document

→ XML DOM

Standard model for XML document

→ HTML DOM

Standard model for HTML document

SAX { Simple API for XML }

SAX is a serial access parser API for XML. SAX provide a mechanism for reading data from an XML document. It is a popular alternative to the document object model.

XML processing with SAX parser.

A parser which elements SAX (i.e a SAX parser) functions as a stream parser, with an event deriven API. The user defines a no. of call back method that will be called when events occur during parsing. The SAX event include-

* XML text node
* XML elment node
* XML processing instructions
* XML comments

Note

SAX parsing is uni-directional, previously parsed data cannot be re-read without starting the parsing operating again

Syntax

```
<?XML version="1.0" encoding = "UTF-8" ?>
<Root Element param="value">
<First Element>
  - - -
  - - -
</First Element>
<Second Element param = "Something">
Pretext <Inline> Inlined text
</Inline text>
Post-text
</Second Element>
</Root Element>
```

Advantages

SAX parsers have certain benifits over DOM style parsers. The quantity of memory that SAX parser must use in order to function is typically much smaller then that of a DOM parser because of the event driven nature of SAX, processing document can often be faster than DOM style parser.

Memory allocation takes time, so the larger memory footprint of the DOM is also a performance issue.

Introduction to Java.

Java is a general purpose object oriented programming language developed by SUN Microsystems of USA in 1991 originally called OAK by James Gosling later on Patrick Naughton

In 1995 OAK was renamed java. In 1996 java development kit JDK 1.0 were introduced.

Features of Java.

1. Compiled and Interpreted
2. Platform independent and portable.
3. Object oriented
4. Robust and secure
5. Distributed
6. Simple, small and familier
7. Multithreading and ~~interact~~
8. Multithreaded and interactive

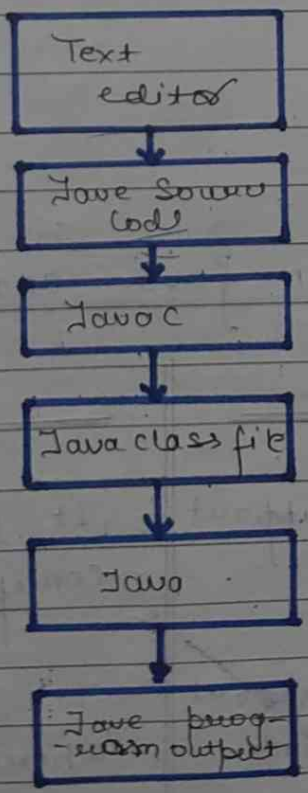
9. High Performance.

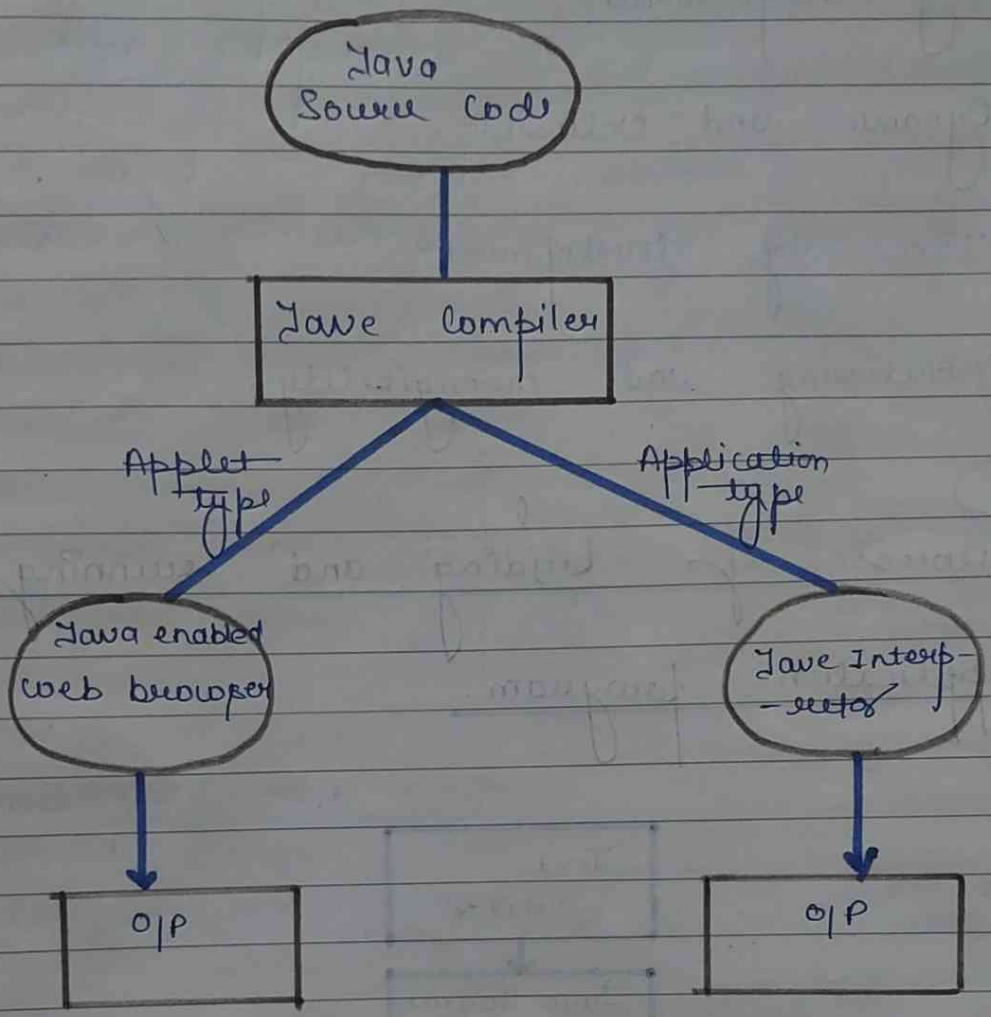
10. Dynamic and extensible.

4. Ease of development.

12. Monitoring and managibility.

Process of buiding and running java application program.





Difference between Java and C++

	Java	C++
1.	Java does not support pointer concept.	It support pointer concept
2.	It does not support multiple inheritance	It support multiple inheritance
3.	Java does not include structures and unions	It have structure and union concept

4. Java include automatic garbage collection.	C++ required explicit memory management.
5. Java has method overloading but no operator overloading	C++ support both method overloading and operator overloading
6. It is platform independent	It is platform dependent
7. It is mainly use design web based application but also use for develop desktop application	It is used for design only desktop application OS, compiler, driver etc
8. Java uses compiler and interpreter both	It use only compiler.

Operators

Operators are special symbols that perform specific operations on one, (1), 2, 3 operands and then return a result

1. Arithmetic operators Eg: +, -, *, /.
2. Assignment operators Eg: +=, -=, *=, /=

3. Conditional operators Eg: ?, :
4. Relational operators Eg: ==, !=, >
5. Boolean logical operators Eg: &, &&, |, ||
6. Increment and decrement operators Eg. i++, j--
7. Bitwise operator. Eg: !, ^, <<, >>

Control flow.

1. Selection statements
   - if statement
   - if else statement
   - if else if else statement
   - switch statement
2. Repetition statements
   - while loop
   - dowhile loop
   - for loop
3. Branching statement
   - Break statement
   - Continue statement
   - Return statement

Array

An array is a collection of variables of the same type that are referred to through a common name. A specific element in an array is accessed by a index. An array type variable is denoted by a base type followed by the empty bracket []

Class

```
Class test
{
public static void main(String args[])
{
System.out.println("Java is better than c");
}
}
```

Set path

C:\Program Files\Java\Jdk1.6.0_27\bin;

>Javac test.java
>Java test(Classname)
Java is better than C

For permanent →

Computer > System Properties > Advanced System Setting >

Environment variables > System variables > Path

* Define a method name main, Java application program must include the main method. This is the starting point for the interpreter to begin the execution of program.

* The key word public is an access specifier that declare the main method as unprotected and therefore making it accessible to all other classes.

* static: which declares this method as one that belongs to the entire class & not a part of any objects of the class main must declare static

* void void state that the main method does not return any value

```
Import java.lang.Math
class SquareRoot
{ Public Static void main (String args[])
  {
     double X=5;
     double y;
       y= Math.Sqrt(X);
       System.out.println("y," +y);
  }
}
```

Eg.

```
Class Array
{
  public static void main(String args[])
  {
    double num[] = {10.1, 20.1, 21.3, 24.3, 20.4};
     double result=0;
       int i;
       for(i=0; i<5; i++)
        result = result + nums[i];
      System.out.println("Arrays is " + result);
  }
}
```

Variables.

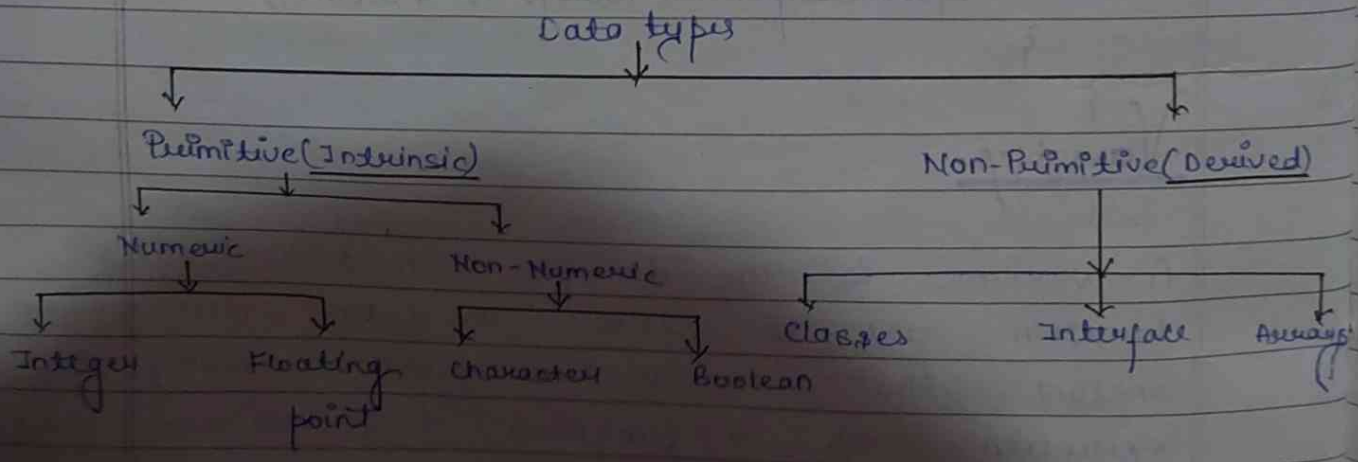
A variable is an identifier that denotes location use to store a data value unlike constant that remain unchanged during the execution of a program, a variable may

take different values at different time during the execution of the program.

Subject to following condition.

1. They must not began with a digit.
2. Upper case and lower case are distinct.
3. It should not be keyword.
4. White space is not allowed.
5. Variable name can be of any length.

Data types



Page
Date

Scope of Variables.

1. Instance Variables
2. Class Variables
3. Local Variables

Method of Class

OR

Method

A class with only data field has no life. The objects created by such a class cannot respond to any messages. We must therefore add methods that are necessary for manipulating the data contained in the class.

Methods are declared inside the body of the class but immediately after the declaration of Instance variable

Syntax.

```
Method name( Parameter)
{
   Method body
}
```

Method Overloading

In java it is possible to create methods that have the same name but different parameter list and different definitions. This is called method overloading.

Method overloading is used when objects are required to perform similar task but using different parameters

```
Class Room
{
  float length;
  float breadth;
  Room(float x, float y)
  {
    length = x;
    breadth = y;
  }
Room(float x)
{
  length = breadth = x;
}
int area()
{ return (length * breadth);
```

Method overriding

Overriding means having two methods with the same method name and parameter (method signature). One of the method is in the parent class and other is in child class

```
Class dog
{
  public void bark()
  {
    System.out.println("Woof");
  }
}
Class Hound extends dog   (extends: Keyword)
{
public void sniff()
  {
    System.out.println("Sniff");
  }
Public void bark()
{
  System.out.println("bowl");
}
}
```

Difference b/w overloading and overriding

Method overloading	Method overriding
1. Method overloading is used to increase the readability of the program	Overriding provide the specific implementation of the method that is already provided by superclass.
2. Method overloading performed within the class	OR occurs in two classes that have same argument (Inheritance)
3. Parameter must be different	Parameter must be same.
4. Ex. of compile time polymorphism	Ex. of Run time polymorphism

Inheritance.

The mechanism of deriving a new class from an old class is called inheritance. The old class is known as the base class or super class and parent class and the one is called the sub class or derived class or

child class

There are four types of inheritance.

1. Single inheritance (only one super class) class → class
2. Multiple inheritance (several superclass) class has a
3. Multilevel inheritance (derived from a derived class)
4. Hybrid inheritance (one superclass and many sub classes)
Hierarchical

Ex.

```
Class cd
{
 int z;
 public void addition(int x, int y)
 {
  z=x+y;
  System.out.println("The sum of two no. "+z);
 }
 public void substraction(int x, int y)
 {
  z=x+y;
  System.out.println("The difference of two no. "+z);
 }
}
Public class my_cal extends cal
{
 int z;
 public void multiplication(int x, int y)
 {
```

```
z = x * y;
System.out.println("The product of two no. " +z);
}
public static void main(String args[])
{
Int a=20 , b=10;
Inheri demo=new Inheri();
demo.addition(a,b);
demo.substraction(a,b);
demo.multiplication(a,b);
}
}
```

Ex.

```
class Box
{
Int w;
Int h;
Int d;
}
class BoxV
{
public class static void main(string[args])
{
Box b = new Box();
Int volume;
b.w=5;
b.h=10;
b.d=15;
```

```
volume = b.w * b.h * b.d;
System.out.println("Volume is " + volume);
    }
}
```

Packages in Java.

Packages are Java's way of grouping a variety of class and / or interfaces together. The grouping is usually done according to functionality; packages act as containers for classes.

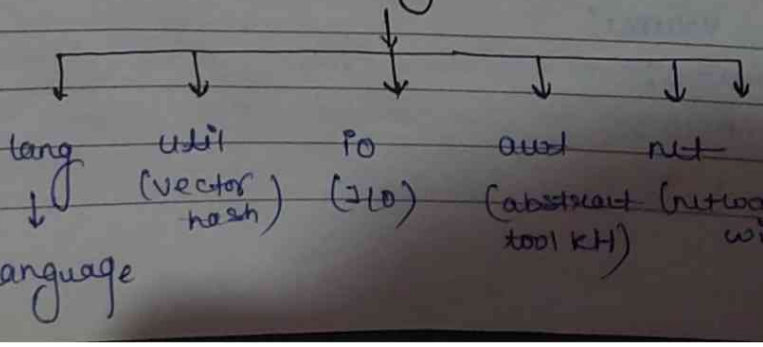
Benefits of packages.

The classes contained in the packages of other program can be easily reuse.

Packages provides a way to hide thus preventing other program or packages.

Packages also provide a way for separating design and coding

Java API Package.



Creating Package

You must first declare the name of the package keyword followed by a package name. This must be the first statement in a Java source file.

```
Package package name;
Class class name
{
- - - -
- - - -
}
```

Package Pack1;
Public class classA
{
Public void display()
{
System.out.println("Class A");
}
}
Import Pack1.ClassA;
Class Pack2
{
Public static void main(String arg[])
{
Class A objectA = new classA();
ObjectA.displayA();
}
}

Interface in Java.

An Interface in Java is a mechanism to achieve fully abstraction. It is a collection of abstract method.

* An Interface is not a class

```
Interface Printable
{
  void print();
}
Interface showable
{
  void show();
}
class A implements Printable, showable
{
 Public void print()
 {
  System.out.println("Hello");
 }
 Public void show()
 {
 System.out.println("welcome");
 }
Public static void main(String args[])
 {
  A obj = new obj();
```

```
        obj.print();
        obj.show();
    }
}
```

Multithreading

Thread. A thread is a similar to a program that has a single flow of control. It is basically a light weight subprocess. Threads are independent, if there occurs exception in one thread, it does not affect other thread, it share a common memory area.

Multithreading

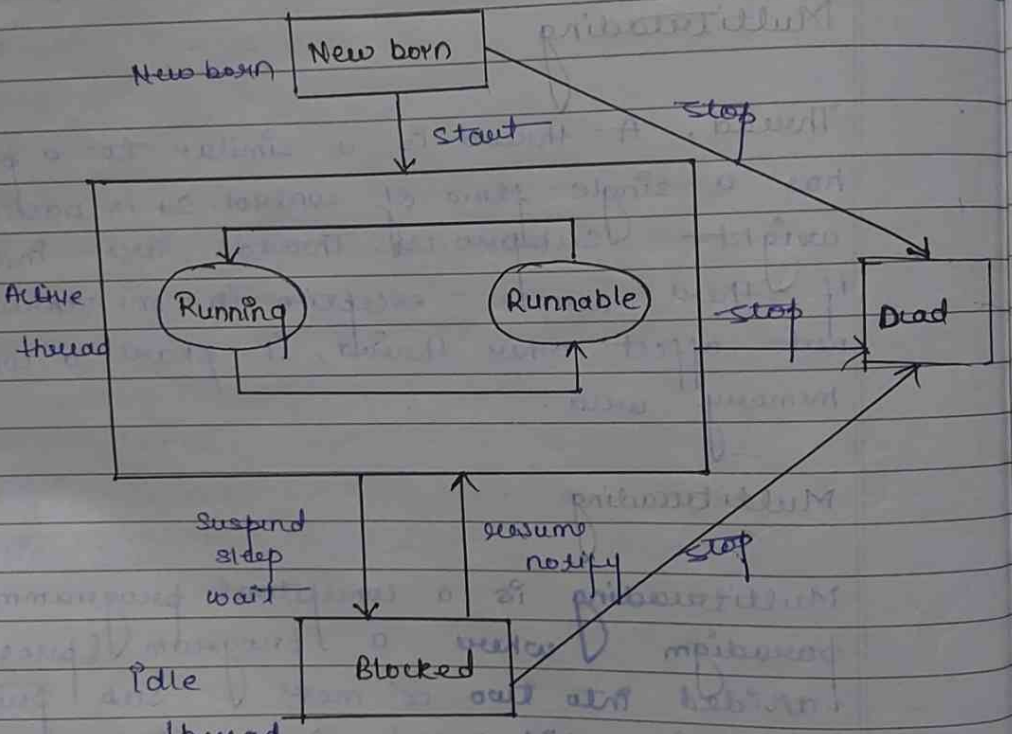
Multithreading is a conceptual programming paradigm where a program (process) is divided into two or more subprograms (process), which can be implemented at the same time in parallel.

Creating threads

Threads are implemented in the form of object that contain a method called run().

```
Public void run()
{
Statement for implementing threads
}
```

Life cycle of a thread.



New born state

When we create a thread object, the thread is born and said to be in new born state the thread is not yet scheduled for running.

Runable state

Runable state means that the threads is

ready for execution and waiting for the availability of the processor.

Running

Running means that the processor has given its time to the thread for its execution. Running thread may be its control in one of the following situation -

1. suspend()
2. wait()
3. sleep()

Killed threads

Blocked state

A thread is said to be blocked when it is prevented from entering into the runnable state & subsequently the running state.

Dead.

A running thread ends its life when it has completed executing its run method run(). It is a natural dead. However we can kill it by sending the stop method. Thus causing a premature death of it.

EXCEPTION HANDLING

Error → Compile time
Error → Run time.

Compile time error All syntax error will be detected and displayed by the Java compiler at missing {semicolon, spelling, codes etc}.

Run time error Some time a program may compile successfully creating the class file but may not run properly such program may produce wrong result due to wrong logic or may terminate due to errors such as stack overflow - an

a) Dividing a integer by zero.

b) Accessing an element that is out of bound of array

c) Passing parameter that is not a valid range.

d) Negative size for an array

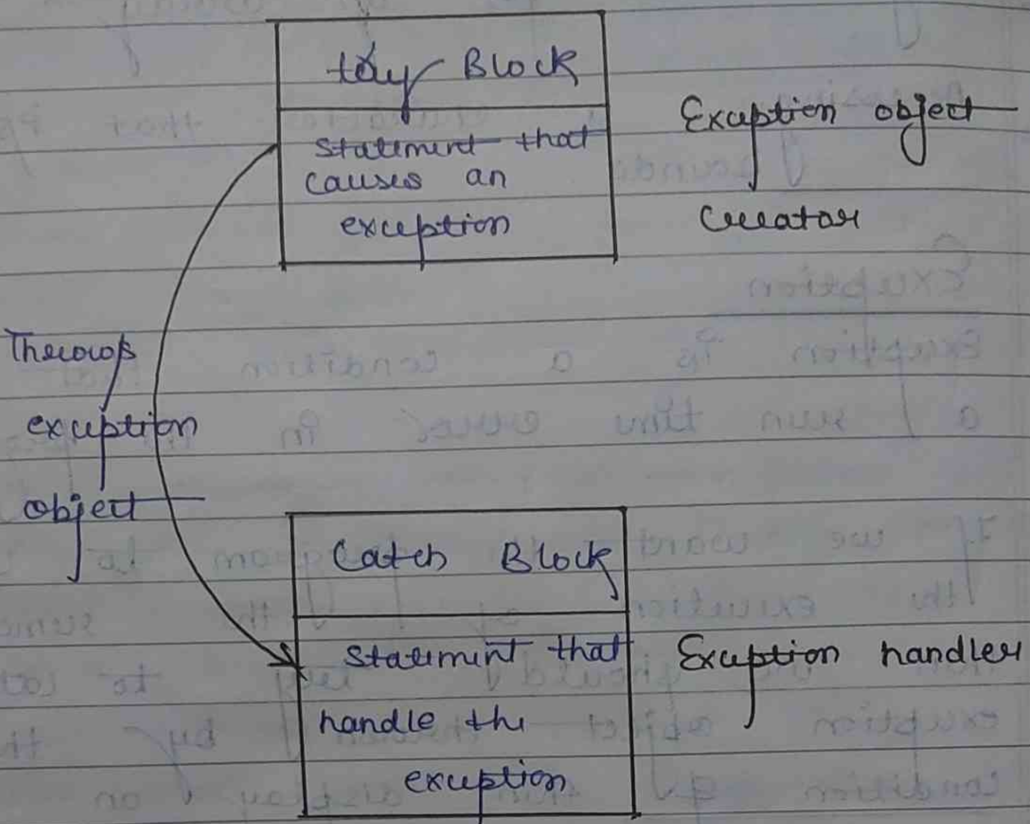
e) Accessing a character that is out of bounds.

Exception

Exception is a condition that caused by a run time error in the program.

If we want the program to continue with the execution of the remaining code then we should try to catch the exception object thrown by the error condition & then display an appropiate message for taking corrective action this task is known as Exception handling.

1. Find the problem (hit the exception)
2. Inform that an error has occurred {Through the exception}
3. Receive the error information {catch the exception}
4. Take corrective actions {Handle the exception}




```
class error
{
 public static void main(String args[])
 {
   int a=10;
   int b = 5;
   int c = 5;
   int x, y;

 try
  {
   x = a/b-c ;   // Exception
  }
```

```
Catch (Arithmetic Exception e)
{
  System.out.println("Division by zero");
}
}
y = a/b+c;
  System.out.println("y = " +y);
}
}
```

Applet Programming

Applets are small java programs that are primarily used in internet computing. They can be transported over the internet from one computer to another computer and run using applet viewer or any browser that support java. An applet developed locally and stored in a local system is known as local applet.

A remote applet is that which is developed by someone else & stored on a remote computer connected to the internet.

* How applet differ from other application (java class).

1) Applet do not use the main() method. for initiating the execution of the code applets when loaded, automatically call certain method of applet class to start and execute the applet code.
2) Applet cannot read from / write to the files in the local computer.
3) Applet cannot communicate with other services on the network
4) Applet cannot run any program from local computer

Preparing to write applet

1. Building an applet code (.java file)
2. Creating an executable applet (.class file)
3. Designing a web page using html tags (.html).
4. Preparing applet tag <applet>
5. Incorporating applet tag into the

web page.

6. Creating html file.

7. Testing the applet code

Example.

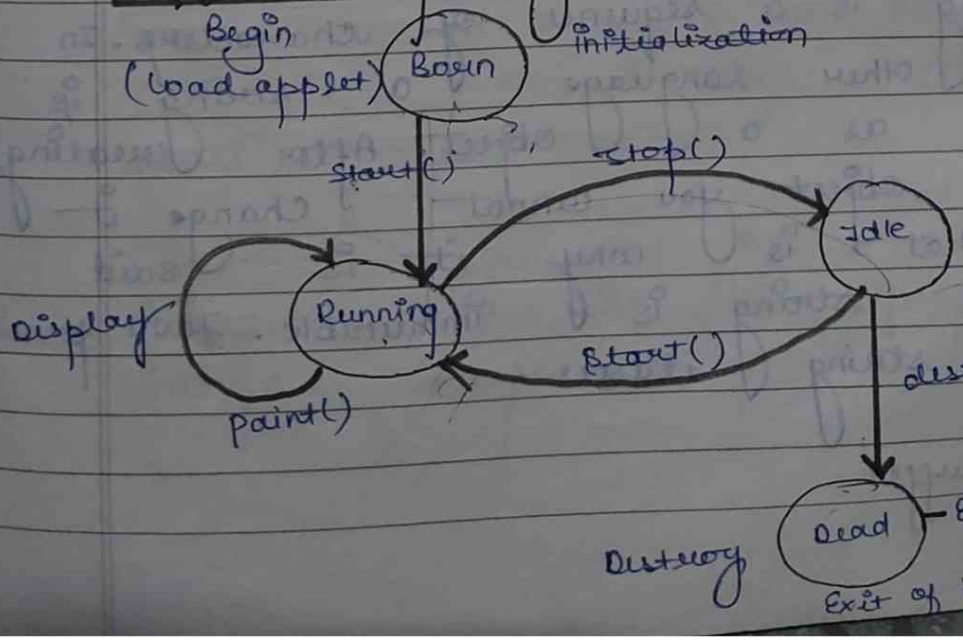
```
import java.awt.*;
import java.applet.*;

Public class HelloJava extends applet
{
  public void paint(Graphic g)
  {
    g.drawString("HelloJava", 10, 100);
  }

}
```

Keyword

* Applet life cycle



Adding applet to html file.

```
<html>
<head>
<title> welcome to java applet </title>
</head>
<body>
<center>
<H1> welcome to the world of applet </h1>
</center>
<BR>
<center>
<Applet code=Hellojava.class width=400 Height=200>
</Applet>
</center>
</body>
</html>
```

Ski String handling

A String is a sequence of characters. In many other language a string is treated as a object. After creating String object you cannot change it BO that is why it is said that a string is immutable. Java provides three string classes -

1. String
2. String buffer

3. String builder.

Two types of creating string object -

1. By string literal
2. By new keyword.

```
public class Exp
{
  public static void main(String args[])
  {
    String S = "Hello";
    S = S.Concat("World");
    System.out.println(S);
  }
}
```

Event Handling.

Event ⇒ Change in state of an object is known as event i.e, event describes the change in state of source.

Event are generated as result of user interaction with the graphical user interface component.

Types of event

1. Foreground Event
2. Background Event

Require the direct interaction of user.

Require the interaction of end user.

Event handling

Event handling is the mechanism that controls the event and ~~decided~~ decides what should happen if an event occur.

This mechanism have the code which is known as event handler that is executed when an event occurs.

Java uses the delegation event model to handle the event.

Delegation event model has the following key participants mainly.—

a) Source

The source is an object on which event occurs. Source is responsible for providing information of the occurred event to its handler. Java provides as with classes for source object.

b) ~~Listner~~ Listner.

It is also known as event handler. Listner is responsible for generating response to an event. Listner wait until it recives an event. Once the event is recived, the listner ~~proces~~ process the event and then returns.

Steps involved in event handling

a) The user clicks the button and then event is generated.

b) Now the object of an concern event class is created automatically and information about the source of event get populated within same object.

c) Event object is forwarded to the method of registered listner class

d) The method is now get executed and returns

* AWT {Abstract window toolkit}

AWT is an API to develop GUI ~~interface~~ based or window based application in Java.

Class {component} that provide a platform Independent API but that make use of platform specific implementation (peers) ~~to~~ be specific, every AWT component class has an equivalent peer class and every component object has a peer object that controls the objects look and feel.

AWT hierarchy

Class:

- Border_layout
- Card_layout
- Checkbox_Group
- Color
- Component
  - Button
  - Canvas
  - Checkbox
  - Choice
  - Container

Container

* Panel
* window
  - Text Area
  - Text Field

- Dimension
- Event
- File_dialog
- Flow_layout
- Font
- Font_metrics
- Graphics
- Grid_Layout
- Image
- Media_Tracker
- Menu_component
  - Menu_bar
  - Menu_Item
- Point
- Rectangle
- Toolkit
- LayoutManager
  - FlowLayout
  - GridLayout
  - BorderLayout
  - CardLayout
  - SpringLayout
  - BoxLayout
  - GridBagLayout

Layout managers are used to arrange components in a particular manner. Layout managers is an interface i.e, implemented by all classes of layout managers.

Layout Manager
- Flow Layout
- Grid "
- Border "
- Card "
- Spring "
- Box "
- Group "

What can a java Script do?

* Java script give html designers programming tool.
* Java script can put dynamic text into a html page.
* Java script can read and write html elements.
* Java script can be used to validate data.
* Java script can be used to create cookies

UNIT-3.

JAVA Scripting

Java Script

Java Script can be thought of as an extensi to html which allows authors to incorporate some sanctionality in their web pages. So, now whenever the user presses the submit button you don't necessary have to invoke, a CGI script to do the processing.

- Java Script can also be used in a no. of ways to spice up your web page.
- It can be used for both client and server side applications.
- It is a cross platform scripting language developed by Netscape.
- Client side Java Script is typically executed in a browser that support the language.
- It is an interpreted object based language that can be included in <html>

- Helps in creation of html pages capable of responding to user events like mouse click, key press, selection of elments in the form
- JavaScript is a scripting language and is a light weight programming language.
- Java script was designed to add intelladively to html pages.
- Java Script is a scripting language which is an interpreted language
- A Javascript consist of lines of executable computer code

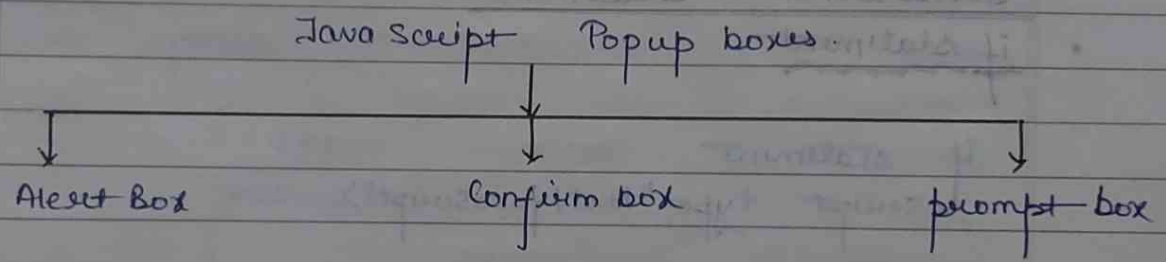
* How to a java script into an html page.

```
<html>
<body>
<script type="text/javascript">
document.write("Hello world")
</script>
</body>
</html>
```

* Variables : Variables are case sensitive in java script

* Operators

* Java Script Pop-up boxes.



Java Script functions.

A function contains code that will be executed by an event or by a call to that function. you may call a function from anywhere within the page.

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello world")
}
```

</script>
</head>
<body> <form>
<input type="button" value="click Me!" onclick="displaymessage()">
</form>
</body>
</html>

* Conditional statement

- if statement

```
if statement
<script type="text/javaScript">

    var d = new Date()
    var time = d.get hours()
    if(time<10)
    {
      document.write("<b> Load moving to all B.Tech
                    Student of AKTU</b>")
    }
</script>
```

- if-else statement

```
<Script type="text/javaScript>

    Var d=new Date()
    Var time dgetHours()
    if(time<10)
```

```
}
document.write("Good morning to all B.Tech Student")
}
else
{
document.write("Good afternoon to all B.Tech Student")
}
</script>
```

Return statement

```
function proc(a,b)
{
x = a*b
return x
}
```

when a call pass product = proc(2,3) = 6

* Java Script loops

```
<html>
<body>
<script type="text/javascript">
var i=0
for(i=0; i<=10; i++)
{
document.write("The number is "+i)
document.write("<br>")
}
```

</script>
</body>
</html>

* Event

By using java script, we have the ability to creat dynamic web pages. Events are actions that can be detected by java script. Every element on a page has certain events which can trigger java script functions.

Examples of Event:

1. A mouse click.
2. A webpage or an image loading.
3. Mousing over hotspot on the webpage.
4. Selecting an input box in an html form.
5. Submitting an html forms.
6. Key stoke.
7. Onload and onunload
8. Onfocus, Onblur and Onchange
9. Onmouse over and Onmouse out.

* Java Script Catching error.

When browsing webpages on the internet

we all have seen a java script ~~colo~~ alert box telling us there is a run time error and asking "Do you wish to debug".

Error message like this may be useful for developers but not for users. When users see errors, they often leave the webpage. There are two ways of catching error in a webpage.

1. By using the ~~try catch~~ try---catch statement
2. By using the on error event

Example.

```
<html>
<head>
<script type="text/javascript">

~~var = "~~
 var text=" "
function message()
 {
  try
  {
  ~~addalext~~ addalert("Welcome guest!")
  }
  catch(err)
  {
```

```
txt="There was an error on this page \n\n"
txt+="Error description: " + err.description + "\n\n"
txt+="Click OK to continue \n\n"
alert(txt)
}
}
</script>
</head>
<body>
<input type="button" value="view message" onclick="message()" />
</body>
</html>
```

* Throw Statement

```
<html>
<body>
<script type="text/javascript">
var x = prompt("Enter a number b/w 0 and 10", " ")
try
{
if(x>10)
throw "Err1"
else if (x<0)
throw "Err2"
}
```

```
Catch(er)
{
if(er=="Error 1")
alert("Error! The value is too high")
if(er== "Error 2")
alert("Error! The value is too less")
}
</Script>
</body>
</html>
```

() "/>

* Object Oriented Programming

Java Script is an object oriented programming language. OOP language allow you to define your own object make your own variable type.

Ajax * Java form Validation

Java Script can be used to validate input data in html form before sending of the content to a server. Form data that typically are check by a Java Script could be -

* Has the user left required field empty.
* Has the user entered a valid email address.
* Has the user entered valid data.
* Has the user entered a text in a numeric field.

```
<html>
<head>
<script type = "text/javascript">
function validate_required(field, alerttxt)
{
with(field)
{
if(value==null||value=="")
{
alert(alerttxt);
return false
}
else
{
return true
}
}
function validate_form(thisform)
{
with(thisform)
```

```
}
if(validate_required(email, "E-mail must befilled out!")==false)
  {
   email.focus(); return false}
  }
}
</script>
</head>
<body>
<form action="Submitpage.html" onsubmit="return
                validate_form(this form)"
                method="post">
Email:<input type="text" name="email" size="30">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Introduction to Ajax. (Asynchronous javascript and XML)

AJAX is a technique for creating fast and dynamic webpages.

AJAX allows web pages to be updated asynchronously by exchanging small amount of data with the server behind the screen. This means that

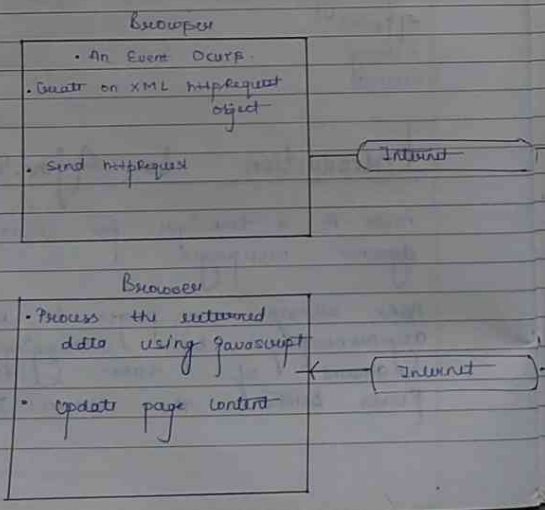
it is possible to update parts of webpage, without reloading the whole page.

Classic web pages (which do not use AJAX) must reload the entire page if the content should change.

Examples of applications using AJAX.

Google maps, Gmail, youtube, facebook.

How AJAX work



AJAX is based on Internet standard.

AJAX is based on Internet standard and uses a combination of -

XML HttpRequest object (To exchange data asynchronously with a server)

JavaScript, DOM {to display/Interact with the Information.

CSS {to style the data}

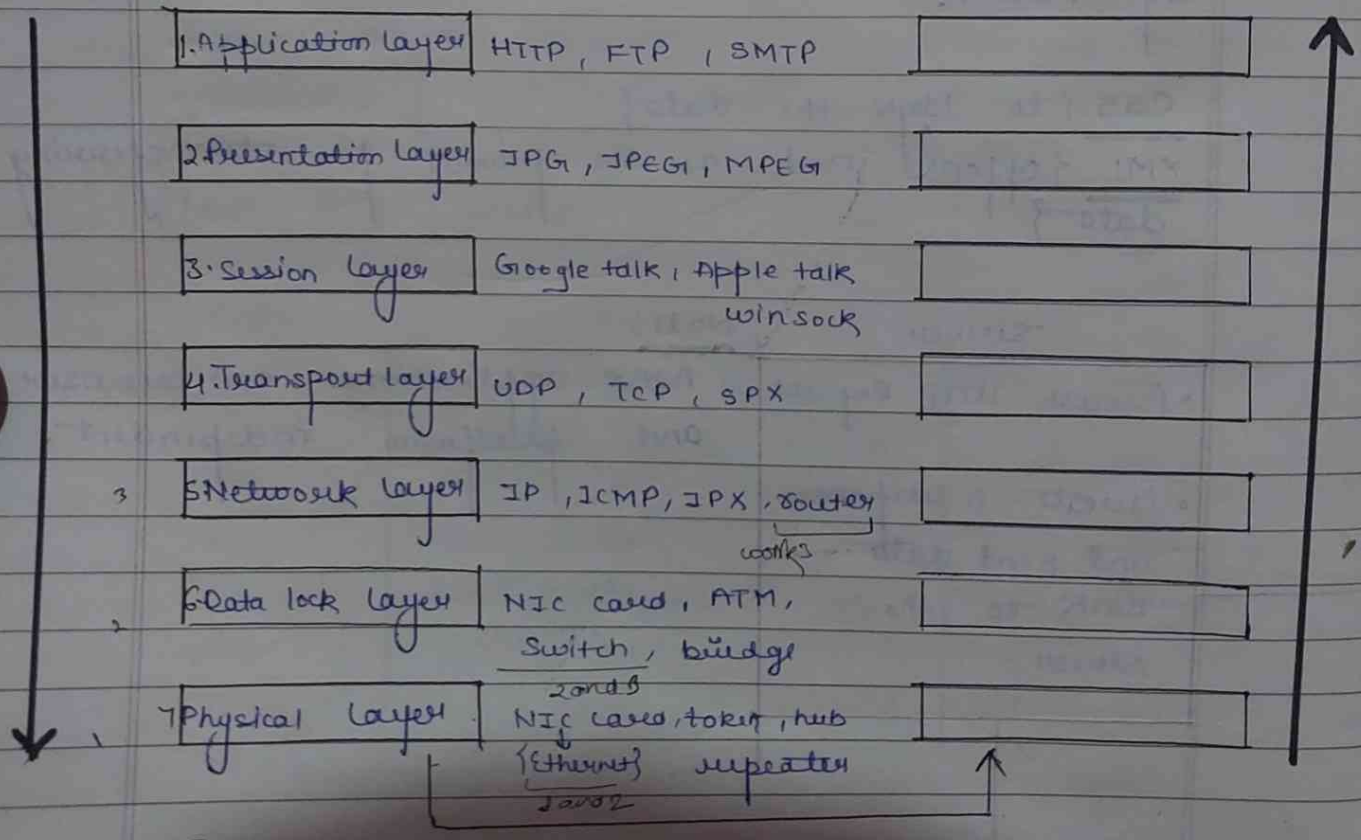
XML {often used as the format for transferring data}

Note:
AJAX applications are browsers and platform Independent.

UNIT-3.

Internet Addressing

OSI Model. (Open system Interconnection)



Addressing.

Class	bits	series.	IP (H)
Class A	N H H H	1-126	(16,777,214)
Class B	N N H H	128-191	(65,534)
Class C	N N N H	192-223	(254)
Multi-casting Class D	N H H H	224-239	N/A
Experim-ental Class E	N H H H	240-254	N/A

172.16.20.137

00010000. 00010100.

Internet Addressing

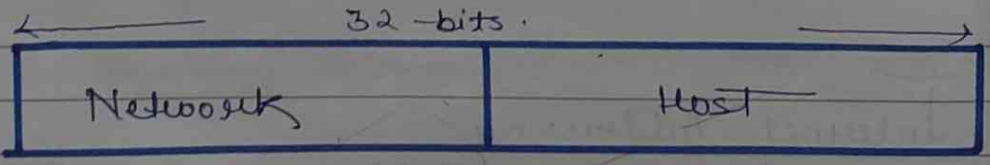
Each host on TCP/IP network is assigned a unique 32-bit IP address consisting of a network no. & a host no.

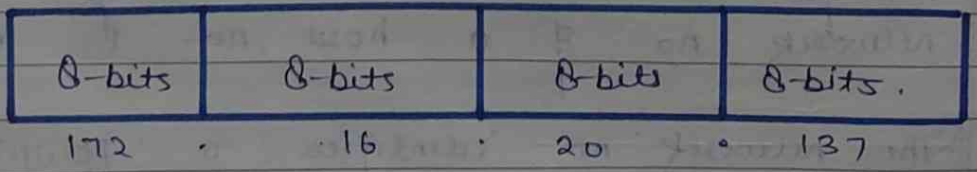
The network no. identifies a specific network and must be assigned by the Internet network informatic center (NIC) or an accredited registerar an ISP can obtain blocks of network address from InternetNIC & can assign addresses as necessary.

The host no. identifies a host on a network and is assigned by the network administrator.

The IP address is grouped into four binary octate (an octatt is a group of 8-bits) and is represented using dotted decimal notation number. The minimum value for an octate is zero & the maximum value is 255.

The basic format is illustrated below:-





Dotted decimal number/notation

IP address support five address classes (A, B, C, D, E) of which only classes

A, B, C are available for commercial use

Class	bits	series	IP(H)
Class A	NHHH	1-126	(16777,214)
Class B	NN HH	128-191	(65,534)
Class C	NNNH	192-223	254
Class D	NHHH	224-239	N/A
Class E	NHHH	240-254	N/A

Class	First Octate	High-Order bit
A	1-126	0
B	128-191	10
C	192-223	110
D	224-239	1110
E	240-254	1111

IPv_4 → 32-bit {Octate में separate करते हैं
IPv_6 → 128-bit Hexadecimal
Dynamic (DHCP)

Inet Address.

The platform on which TCP/UDP are build is called internet protocol(IP) address. An IP address comprises of either a 32-bit or 128-bit unsigned number used by IP and uses host name with which either it may be constructed or may have done a reverse host name resolution. Going by that logic the designer of Java could have simple represented IP addresses as long numbers but they creat a powerfull class called Inet Address that represint an IP address.

The various addresses types are as follows-

1. **Unicast :-**
As the name suggest, it is an Identifier for a single interface. A packet send to a unicast address is delivered to the interface identified by that address. The unicast address can be the following type -

a) **Unspecified address**

b) **loop back address**

Factory method OR static method
Dynamic OR Instance method

2. Multicasting

In contrast to unicast, multicast is an identifier for a set of interfaces (typically belonging to different nodes).

A packet send to a multicast address is delivered to all interfaces identified by that address

* IP Addresses Scope

The IPAddress scope can be classified into -

1. Link local address

They are designed to use for addressing on a single link for purpose such as auto address configuration neighbouring discovery or no routers are present

2. Site local address

They are designed to used for addressing inside A site without the need for a global prefer.

3. Global addresses

They are unique across the internet.

* Host name resolution

Host name to IPAddress resolution is accomplished through the use of a combination of local machine configuration information & network naming service such as domain name system (DNS) and network information service (NIS).

The particular naming service being used is by default the local machine configured one. For any host name its corresponding IP address is returned.

* Reverse name resolution means that for any associated IPAddress the host with the IPAddress is returned.

The InetAddress class provides method to resolve host name to their IP address & viu-versa.

* factory methor or Static method. IP fixed

Since the InetAddress class has no

Visible constructors, factory methods have to be used to create an Inet address object. Factory methods are a convention where by static methods in class return on instance of that class. Instead of overloading a constructor with various parameter list. and It is better to have unique method names, The commonly used Inet address factory methods are

Static InetAddress getbyName(String Host()) throws UnknownHostException

Static InetAddress getLocalHost() throws UnknownHostException

Static InetAddress getAllbyName(Stringhost) throws UnknownHostException

The get local host method simply returns the InetAddress object that represent the local host. The get_By_Name() method returns on InetAddress for a host name passed to it, if these method are unable to resolve the host name they through an unknown host exception

Instance method.
The InetAddress class also has several other method called instance method, which can be

used on the object returned by the method, the most commonly used methods-

- boolean equals(Object obj)
  byte[] getAddress()
  String getHostAddress()
  String getHostName()
  boolean isMulticastAddress()
  String toString()

Internet address are looked up in a series of hierarchical cached server. This means that your local computer might know a particular name-to-IP address mapping automatically, such as for itself & for near by servers.

For other means it may ask a local DNS server for IP address information. If that server does not have a particular address it can to a remote side & ask for it this can continue all the way upto the route server.

Socket programming

Socket provide the communication mechanism b/w two computer using TCP. A client program create a socket on its end of the communication and attempt to connect that socket to a server. When the connection is made the server create a socket object method on its end of the communication. The client & server can now communicate by writing

To start with the Socket programming you should know how to open a socket. If you are programming a client, then you need to open a socket like this

```
Socket myclient;  myclient=newSocket("My machine name"; PortNumber);
```

where my machine name is the machine you are trying to open a connection to, and port no. is the port (a number) on which the server you are trying to connect is running to create an input stream using data stream (DataInputStream) class to receive response from the server that allows you to read lines of text & Java primitive datatype in a portable way by having method such as read, ReadChar, ReadInt, ReadDouble and ReadLine

```
DataInputStream inputstr;
try
{
 inputstr = new DataInputStream(myclient.getInputStream());
}  IOExceptione
catch(e)
{
 System.out.println(e);
}
```

```
try
{
    output.close();
    input.close();
    ServiceSocket.close();
    MyService.close();
}
catch(IOException e)
{
    System.out.println(e);
}
```

URL (Uniform resource locator)

Class URL represents a uniform resource locator, a pointer to a "Resource" on the WWW.

A resource can be something as simple as a file or a directory, or it can be a reference to a more complicated object such as a query to a database or to a search engine.

The abstract class URL_Connection is the superclass of all classes that represent a communications link b/w the application and a URL.

URL

Instances of this class can be used both to read from and to write to the resource referenced by the URL. To retrieve the information, process it and sending the results back to the server or just display the required information retrieved from the server is what we are aiming at.

Consider, Ex a small application which ask for a movie, movie name from user & in turn returns the "imdb" rating of the movie or return all the links related to that movie.

All of this can be achieved using the URL connection class "URLConnection is an abstract class whose sub classes form the link b/w the user application and any resource on the web.

We can use it to read/write from/to any resource referenced by an URL object.

There are mainly two sub classes that extends the URL connection class

1) HttpURLConnection
2) JarURLConnection

1) HttpURLConnection

If we are connecting to any URL which uses "Http" as its protocol, then URLConnection class is used.

2) JarURLConnection.

If however we are trying to establish a connection to a jar file on a web, then ~~JarCon~~ JarURLConnection is used.

Conclusion

Once the connection is established and we have a ~~URL connn~~ URLConnection object, we can use it to read or write or get further information about when was the page/file last modified, content length etc.

Important Methods.

1. ~~URL~~ to URLConnection openConnection()
2. Object getContent()
3. Map<String, List> getHeaderFields()
4. getContentEncoding()
5. getContentLength()
6. getDate()

Step involved in the above process

1. URL Creation: Create a URL object using any of

the Constructor given it.

2) Create object: Invoke the openConnection() call to create the object of URLConnection.

3) Display the Content: Either use the above created object to display the information about the resource to read/write contents of the file to console using BufferedReader and InputStream of the open connection using get_InputStream() method.

4) Close stream: Close the InputStream when done.

DATA GRAM

A datagram is a basic transfer unit associated with a packet switched network. Data grams are typically structured in header & payload section. Data grams provide a connectionless communication service across a packet switched network. The delivery, arival time & order of arival of datagram need not be guaranteed by the network.

A datagram need to be self contained without reliance on earlier exchanges because there is no connection of fixed duration b/w the two communicating points.

as there is.

Datagrams in Java.

TCP/IP style networking provides a serialized, predictable, reliable stream of packet data. This is not without its cost, however. TCP includes algorithm for dealing with congestion control on crowded network, as well as pessimistic expectations about packet loss.

This leads to inefficient way to transport data.

"A datagram is an independent, self contained message send over the network whose arrival, arival time and content (data) are not guaranteed."

- Data gram plays a vital role as an alternative.

- Datagrams are bundles of information passed b/w machines. Once the datagram has been released to its intended target, there is no assurance, that it will arrive or even that someone will be there to catch it

Java object.

Features
1) Managing persistent data
2) Uniquely identify by a primary key
3) Performing complex logic involving dependent object.

- Likewise, when the datagram is received, there is no assurance that it hasn't been damaged in transit or that however send it is still there to receive a response & it is crucial point to note.

Java implements datagram on top of the UDP protocol by using two classes

1. DatagramPacket {object is the data container}
2. DatagramSocket {is the mechanism used to send or receive the datagram packets}

DatagramSocket defines four public constructor -

1. DatagramSocket() throws Exception
2. DatagramSocket(int port) throws Exception
3. DatagramSocket(int port, InetAddress ipAddress) throws SocketException
4. DatagramSocket(SocketAddress address) throws SocketException

SocketAddress is an abstract class that is implemented by the concrete class InetSocketAddress.

Entity bean

An entity bean is a remote object that manages persistent data, perform complex business logic, potentially uses several dependent java objects & can be uniquely identified by a primary key. Entity beans are normally coarse grained persistent object, in that they utilize persistent data stored within several find grained persistent

UNIT-4

JAVA BEAN / JDBC

Java Bean.

A bean is a reusable and self contained software component. Beans created using Java take advantages of all the security and platform independent features of Java. Reusable software components are designed to apply the power & benefits of reusable, interchangeable parts from other industries to the field of software constructions.

Reusable software components can be simple like families push buttons, text field list boxes, scroll boxes, dialogue

Java beans development phases.

- Construction phase bean बनाना
- Build phase
- Execution phase

The construction phase involves creation of java bean and gets its user interface. A java bean cannot exist by its. In order to access its functionality, it must be placed inside some other elements of a running application called the container.

A container can be a frame window or a web page.

Build phase.

Building of a new phase and reuse components.

Execution phase

Execution take place in this phase.

"In computing based on the java platform, java beans are class that encapsulate many objects into a single object the bean. They are serializable have a zero argument constructor & allow access to properties (getter/setter)."

BDK (Bean development kit)

A wide variety of application development environment such as forte for java IBM, visual age, Kawa, BDK and

Semantic Visual Cafe are available in the market that support the creation, building & testing of Java beans, BDK is used as tool to -

(A) Build applications that consist of one or more beans. BDK contains a variety of reusable example java beans.

The beans that make up an application can either be a pre-existing example bean of BDK or a bean that you create

(B) Test the beans that you create for their intended functionality.

There are several component windows, we get after starting the BDK, which are as follows -

1) Tool Box : Contain the a list of sample beans supplied by BDK
2) Bean Box
3) Properties
4) Method Tracer

uyt

JAR File. {Java archieve}

A Jar file is similar to the zip file & contain all class files of an application. A Java bean is packaged into a JAR file for distribution.

JAR file contains the manifest file & all other file such as the class file & all pictures file of the bean.

Create a login bean.jar file

jar cfm loginBean jarloginBean.mft
loginBean.class

Enterprise Java Bean {EJB}

Sun Microsystem introduce the J2EE application server & EJB specification into the multitier server side component architecture market.

It is important to note that through EJB & javabeans are both component models, they are not same.

EJB's are interprocess component &

java beans are intra process component.
EJB is a specification for creation of server side component that enables & simplifies the task of creating distributed object.

The Key factor of EJB are as follows:-

EJB components are server side component written using java.

EJB component implement the business logic only.

EJB components provide services such as transaction & security management & can be customized during deployment

EJB can maintain straight into access various method.-

→ The whole bussiness logic takes place in EJB container which is apart EJB server.

→ EJB server which contains the EJB container which contains enterprise means. bean.

→ Enterprise bean which contain method

that Implement the business logic

EJB Architecture.

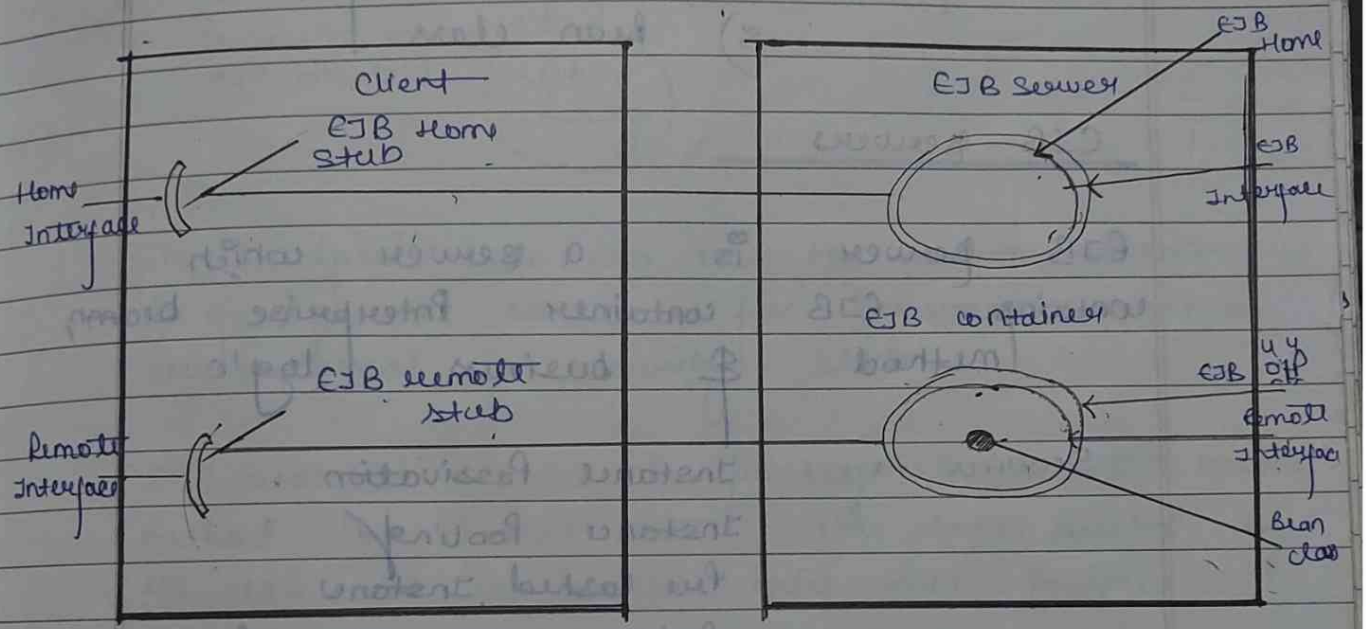


fig: EJB container acts as a interface b/w client & EJB server

EJB architecture consist of two side i.e., client side & server side. At the client side the following components are present.

1) Home Interface
2) Remote "
3) EJB Home Stub
4) " Remote stub

At server side the following components are there - 1) EJB home.
2) Home Interface
3) EJB object
4) Remote Interface
5) Bean class

EJB Servers

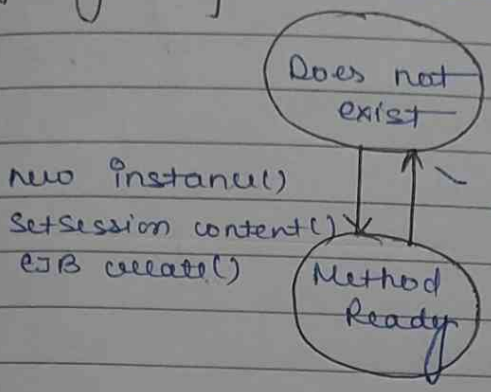
EJB server is a server which contains EJB container Enterprise beans, method & business logic

Services → Instance Passivation.
Instance Pooling
Pre Cashed Instance
Database connection Pooling

EJB Container.

Services
* Security
* Transaction management
* Persistence.
* Life cycle management {entity bean, session bean, message driven EJB}
  - Stateless
  - Stateful
*
*

Life cycle of stateless session bean.



The enterprise bean can use the session context to query the container for information such as transactional & security state.

EJB create() method is finally invoked, EJB create() method is similar to the constructor of the class. It is invoked only once in the lifecycle of stateless session bean.

Now the enterprise bean instance is said to be in method ready state. In this state bean is ready to service the clients request. The EJB remove() method end the life cycle of stateless session bean.

Entity bean

An Entity bean is a remote object that manages persistent data perform complex business logic, potentially uses several dependent java objects & can be uncqually identified by a primary key. Entity beans are normally

Difference b/w session bean & entity bean.

Session bean.

* Session bean represents a process of flow
* Each session bean is associated with one EJB client at a time.
* Session bean have type stateless & stateful
* Stateless bean dont maintain state while statefull bean maintain state
* Session bean does not survive in middle ware system shutdown

Entity bean

* Entity bean represent row in a database
* Each entity bean associated with more than one client at a time.
* It has two type.
  CMP container manage persistent
  BMP Bean manage persistent
* Entity bean do survive in System shut-down.

Establishing JDBC connection

Establishing a connection is accomplished in two steps :-

Firstly we need to load the JDBC driver for the RDBMS (Relational DBMS)

Secondly we open the connection to specifically database managed by the system

loading JDBC drivers.

loading a JDBC driver is very simple & involves just a single line of java code.

Loading a JDBC driver

```
try{
    class.forName("com.mysql.jdbc.Driver");
  }catch(ClassNotFoundException exc)
    {
     exc.printStackTrace();
    }
```

This line of code will tell the driver manager which java class to be load as a JDBC driver class.

Loading a JDBC Driver connecting to a database.

```
- - - - -
Connection Connection;
- - -
try {
Connection = DriverManager.getConnection("jdbc:mysql://"+host+"/"+db,username,password);
    }
catch(SQLException exc)
{
exc.printStackTrace();
}
```

To create a particular connection to a database we call getConnection method of the driver manager class. This method takes the three argument. A JDBC URL which follows the standard URL syntax the protocol part start always with (jdbc:) followed by the name of the rdbms (mysql) & the terminated with ://. The host part identifies the host where the rdbms is running. In the case that the java application connecting to the rdbms & the system itself & running on the same machine "localhost" can be used to identify host.

* A registered user privilege for

* corresponding

JDBC statement

JDBC statem

instance that

instances of th

```
- - -
try
{
statement stat
}
catch(SQL Excepti
{
exc printStack
}
```

* A registered username that has the proper privilege for manipulating the database.
* Corresponding password for the username.

JDBC statement

JDBC statements are created by an instance of the connection class. Thus, the connection instance that we created in the to obtain instances of the statement class

```
ost+"/"+db,username,password);
```

```
try
{
 Statement statment = Connection.createStatment();
}
catch(SQLException exc)
{
 exc printStackTrace();
}
```

INSERT DATA INTO DATABASE

```
---
try
{
String insert_sql_start = "INSERT INTO" + table + " values(" + values + ")";
Into row = statement execute update(insert_sql_start, Statement
                                    - Return_generated_keys);
ResultSet Keys = statement.getGeneratedKeys();
{ Catch SqlException ex()
  {
   exc.print Stack Trace();
  }
}
;
;
```

EXAMPLE

```
import java.net.*;
import java.sql.*;        for JDBC
import java.io.*;

Class MakeDB
{ Public static Void main(String args[])
  {
    try{
      Class forName("Spcial db Driver");
```

```
String dbaseURL = "jdbc".mysubprotocol ://dbasehost /dbasena-
                                                    -me";
Connection dbconnection = DriverManager.getConnection (dbaseURL,
                          "dbaseuser", "dbasepassword");
Statment query = dbconnection.getStatment();
Resultset results = query.executQuery ("Select First_name,
                                last_name, user_table);
while(results.next())
{String fname = result.getstring("First_name");
String lname = result.getstring("last_name");
System.out.println("Found user" + fname + " " + lname);
}
catch (SQL Exception e)
{
System.out.println("SQL Exception:"+e);
}
Catch(class NotFound Exceptione)
{
System.out.println ("Class Not Found Exception:"+e);
}
}
```

Prepared statements :-

A prepared statement is a special kind of statement object with some useful features. Remember, you need a statement in order to execute either a query or an update. You can use a prepared statement instead of a statement & benefits from the features of prepared statement.

* The prepared statement's primary features are;
* Easy to insert parameters into sql statement
* Easy to reuse the prepared statement with new parameters.
* May increase performance of executed statements
* Enables easier batch updates.

Creating a prepared statement :-

Ex:-

```
String sql = "Select * from people where id = ?";
```

1) `Prepared statement : prepared statement = connection.prepared statement (sql);`

2) `String sql = "update people set firstname = ?, lastname = ? where id = ?";`

```
Prepared statement preparedstatement = Connection.preparedStatement(sql);
preparedstatement.setString(1, "Gorky");
preparedstatement.setString(2, "Larson");
preparedstatement.setLong(3, 123);
```

int re

Reusing

once a p
after ex
by setting
execute

Performan

It takes
create a
an analy
the quer

There are
prepared
1. Reuse of
2. Reuse by

The jdbc
internally
-ment
prepared
Sec
cou

int rowAffected = preparedStatement.executeUpdate();

Reusing prepared statement

Once a prepared statement is prepared it can reused after execution. You reuse a prepared statement by setting new values for the parameter & then execute it again.

Performance of prepared statement

It takes time to pass database an sql string & create a query plan for it. A query plan is an analysis by how the database can execute the query in the most efficient way.

There are two levels of potential reuse for a prepared statement :-

1. Reuse of prepared statement by Jdbc driver.
2. Reuse of prepared statement by the database.

The jdbc driver can cache prepared statement object internally, thus the reuse the prepared statement object. This may save a little of prepared statement creation time.

Second, cached parsing & query plan could potentially be reused across Java application for instance application server in a cluster, using the same database.

```
atement(sql);
where id=? ";
edStatement(sql);
);
```

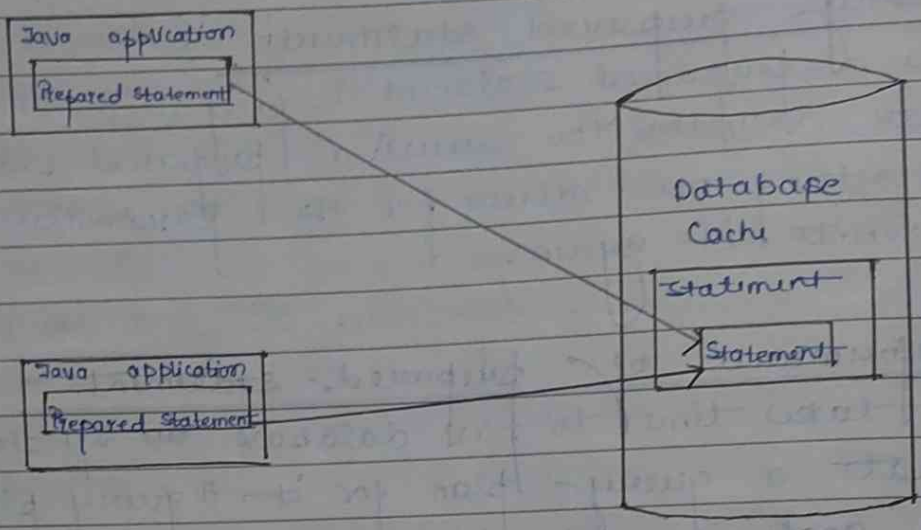


Fig: Caching of prepared statements in the database.

Transaction Processing {management}

Method	Description
1. Void SetAutoCommit(boolean Status)	Set void It is true by default means each transaction is committed by default
2. void commit()	Commits the transaction
3. Void rollback()	Cancel the transaction

```
Class fetchRecords
import java.sql.*;
class FetchRecords
{Public Static void main(string args[])
throw Exception
{
ClassforName("Oracle.jdbc.driver.oracleDriver");
Connection Con=DriverManager.getConnection("jdbc:oracle
:thin:@localhost:1521xe","System","Oracle");
Con.SetAutoCommit(false);
Statiment smt=Con.Creatstatiment();
Smt.executUpdate("Insert into user420 values(190,'abhi',20000)");
smt.executUpdate("Insert into user420 values(190,'umesh',40000)");
Con.commit();
Con.Close();
}
}
```

Servlet

13/11/18

UNIT- 5

Servlet

Servlets are java programs that can be deployed on java enabled web server to inhance & extend the functionality of the web server

Ex. You can write a servlet to add a messenger servic to the UPTU website. servlet can also be used to add dynamic content to the web-pages

Java servlets are server side components that provides a powerful mechanism for deploying server side web application.

With java servlet web developer can create fast & efficient server side application & can run it on any servlet enabled web server

Servlet provide components - based, platform independent methods for building web based applications without the performance limitation of CGI programs.

Servlet are not designed for specific protocol. It is different things that they are not most commonly used with the http protocol. Servlet use the classes in the java packages.

javax.servlet
javax.servlet.http

http servlet typically used to -

1) Provide dynamic content like getting the result of database query & returning to the client.

2) Process/store the data submitted by the html.

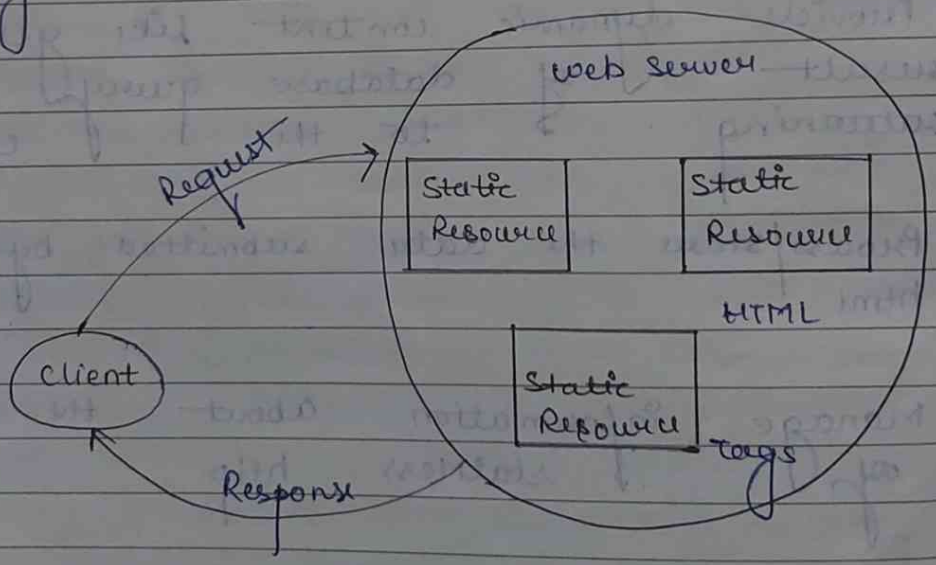
3) Manage information about the state of stateless http

Ex. Online shoping can manage request for multiple concurrent customers.

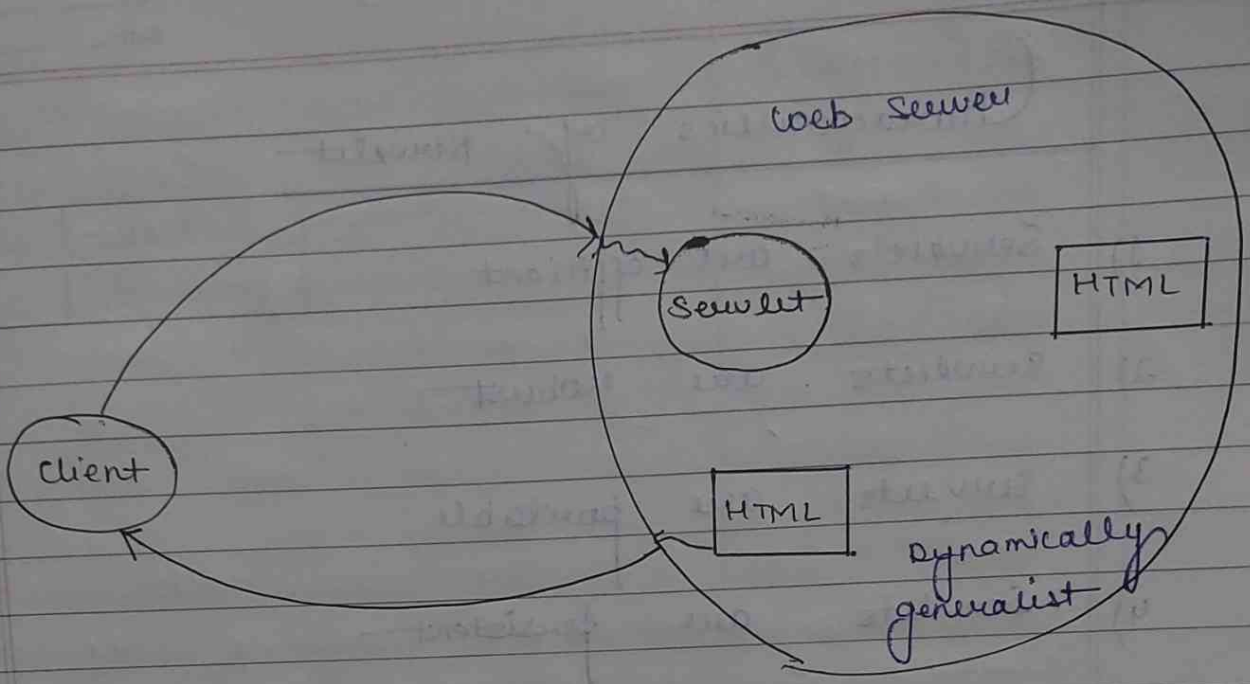
4) Servlet is a java program that programatically extend the functionality of the web server in request response programming model

5.) Servlet is executed within a web server hence it does not have visual interface.

Servlet are responsible for generating dynamic content in web applications when data or information does not differ from user to user we will use static pages otherwise dynamic page will be used.



In this case web server can only serve static contents.



web server can provide both static & dynamic content.

Note:

Servlet are not server side applet but it can be think as of servlet applets as both have similar life cycle method. Applet is a sub class of panel so applet are viewable but servlet are not viewable.

Advantages of Servlet

1) Platform independence
2) Performance
3) Extensibility
4) Safety
5) Security

Characteristics of Servlet

1) Servlets are efficient.
2) Servlets are robust
3) Servlets are portable
4) Servlets are persistent

Life cycle of Servlet

A servlet is loaded only one in the memory and is initialized in the init() method after the servlet is initialized it starts accepting request from the client & process them through the service method till it is shutdown by the destroy() method.

The service() method is executed for every incoming request. The life cycle of servlet is depicted below

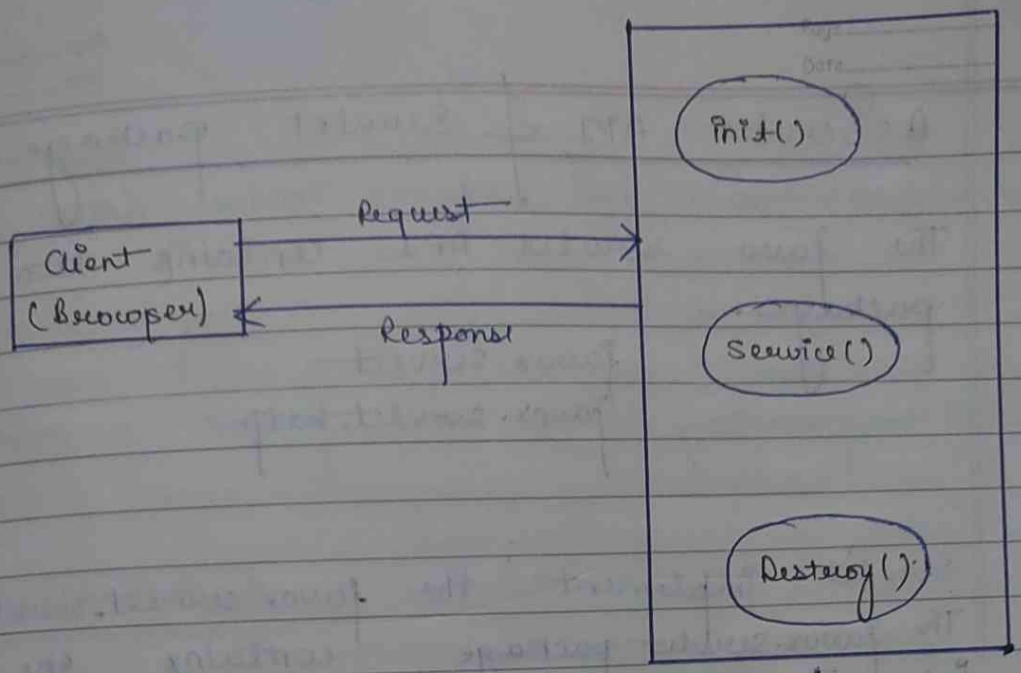


Fig: Life cycle of servlet

JSDK {Java Servlet development Kit}

J2EE {Java Second enterprise edition} is a set of specification that defines the standard ~~addition~~ form creating distributed object.

J2EE also specifies how these technologies can be integrated to provide complete Soln.

It is also standard architecture that defines a multitier programming model.

The ~~do~~ Java 2 SDK enterprise edition (J2EE) server is a product from Sun microsystem

Servlet API & Servlet package.

The java servlet API contains two core packages :-

javax.servlet
javax.servlet.http

Servlet implement the javax.servlet.servlet interface. The javax.servlet package contains the generic interface & classes that are implemented & extended by all servlet.

The javax.servlet.servlet defines five method-

1. **Service() method**
Called by servlet to handle the client request in a new thread.

2. **Init()**
Init() method is called by ones when servlet is loaded it is a good place to initialize global variables.

3. **getServletConfig()**
Return the servlet configuration object passed to Init().

4. Destroy()

Is called when servlet is unloaded from memory

5. getServletInfo()

Return a string with version, copyright information etc.

* http() method

1. get
2. Post
3. put
4. head
5. delete
6. patch
7. options

1. get().

The get() the data is send as URL parameters that are usually strings of name & value pairs separated by &. In general a URL will get data will look like this

http://www.com/action.php?name=sam&weight=55

the bold part in the url are the get parameters & the underline part are the value of those parameters.

* More than one parameter equal value can be embedded in the URL by concatinating with &.

* One can only send simple ~~text~~ data via get().

Advantages of using get()

1) Since the data send by the get() are displayed in URL, it is possible to bookmark the page with specific query string values.

2) get request can be cached & get request remain in the browser history

~~One can only send~~

3) ~~Get~~ request can be bookmarked.

Disadvantage

1. The get method is not suitable for passing sensitive information such as username & password because they are fully visual in the URL query string as well as potentially

2. Stored in the client browsers memory as a visited page.

3. Because the get() assigns data to a server environment variable, the length of the URL is limited so there is a limitation for the total data to be send.

2. Post()

In post method the data is sent to the server as a package in a separate communication with the processing script. Data sent through post method will not be visible in the URL.

Post/test/demoform.php.http/1.1 Host:gfg.com. SAM=951&MAT=62

The query string name & weight is sent in the http message for body of a post request

Advantage

* It is more secure than get because user entered information is never visible in the URL query string or server logs.
* There is more security.
* There is a much larger limit on the amount of data that can be passed & one can send text data.

as well as binary data using post method.

Disadvantages

- Since data send by post method is not visible so it is not possible to bookmark the page with specifies query
- Post request are never cached
- Post request do not remain in the browser history

Difference b/w get() & post()

Property	get()	post()
History	Parameters remain in the browser history because they are part of URL	Parameters are not saved in browser history
Bookmarked	Can be bookmarked	Cannot be bookmarked
Back button / Resubmit behaviour	Get request are re-executed but may not be resubmitted to the server if the html is stored in the browser cache	The browser usually alerts the user that data will need to be resubmitted

Hacked	Easier to hacked for script kiddies	More difficult to hack
Security	Get is less secure because compare to post, data sent is part of the URL so is saved in browsers history & server logs in plain text	Post is little safer that get because the parameters are not stored in browsers history or in web server logs
Usability	get() should not be used when sending password or other sensitive information	Post() is used when sending password or other sensitive information
Visibility	get() is visible to everyone & has limits to the amount of information	post method variables are not displayed in the URL
Cached	Can be cached	Not cached

Restriction on form datatype	Yes, only ASCII characters allow.	No, binary data is also allowed

Redirecting request to other services :-

URL redirection, also known as URL forwarding, is a technique to give a page, a form or a whole web application, more than one URL addresses. Http provides a special kind of response http redirects to perform this operation used for numerous goals! temporary or redirection while site maintenance is on going, permanent redirection to keep external links working after a change of the site architecture, process page when unloading a file & so on.

Permanent redirection
These redirections are meant to last forever. They imply that original url should not be

URL rewriting: You can append some extra data on the end of the url that identifies the session, & the server can associate that session identifier with data it has stored about that session.

Ex.

With http://google.com/file.htm;sessionid=12345;

The session identifier is attached as sessionid =12345 which can be accessed at the web server to identify the client.

URL rewriting is a better way to maintain session as it works even when browsers don't support cookies. The drawback of URL rewriting is that you would have to generate URL dynamically to assign a sessionid, even in case of a simple static html page.

Session Tracking API: Session tracking API is built on top of the first four method. This is in order to help the developer to minimize the overhead of session tracking. This type of session tracking is provided by the underlin technology lets by take the

Java servlet example then, the servlet container manages the session tracking task of the user need not do it explicitely using the java servlets. This is the best of all methods because all the management of errors related to session tracking will be taken care of by the container itself.

Every client of the server will be mapped with a javax.servlet.Http.Httpsession object.

Java servlet can use the session object to store & retrieve java objects across the session.

Session tracking is at the best when it is implimented using session.

The Httpsession object

Apart from the method mention three ways, servlet provide Http session Interface which provides a way to identify a user across more than one page request or visit to a website & to store information about that user.

The servlet container uses this interface to create a session b/w an Http client and an Http server.

The session persist for a specified time period, across more than one connection or page request from the user.

You would get Http session object by calling public method getSession() of Http servlet request

```
Http session  session = request.getSession()
```

Method.

* Public object getAttribute(String name)

Description This method returns the object bound with the specified name in this session or null if no object is bound under the name.

* Public Enumeration getAttributeNames()

Description This method returns an enumeration of string object containing the names of all the object bound to this session.

The servlet container uses this interface to create a session b/w an Http client and an Http server.

The session persist for a specified time period, across more than one connection or page request from the user.

You would get Http session object by calling public method getSession() of Http servlet request

```
Http session  session = request.getSession()
```

Method.

* **Public object getAttribute(String name)**

Description This method returns the object bound with the specified name in this session or null if no object is bound under the name.

* **Public Enumeration getAttributeNames()**

Description This method returns an enumeration of string object containing the names of all the object bound to this session.

* Public long getCreationTime()

Description This method returns the time when this session was created measured in miliseconds since mid-night

* Public String getId()

Description This method returns a string containing the unique identifier assigned to this session.

Cookies :- You can use Http cookies to store information. Cookies will be stored at browser bit

URL rewriting : With this method, the information is carried through URL as request parameters. In general added parameter will be sessionId, userId.

Http session : Using Http session we can store information at server side, Http session provides method to handle session related information.

JSP { Java Server page }

JSP technology allows you to rapidly dwelop & easily maintain information rich, dynamic web pages as part of the java family, JSP enables dwelopment of web based applications that are platform independent.

JSP technology works with a wide variety of web servers, application server browsers & dwelopment tools

* Features of JSP.

1) Platform & server independent
2) Environment.
3) Extensible JSP tags.
4) Reuseability across platform.
5) Easier maintinance

* Working of JSP technology

JSP page is simply an html web pages, which contains additional bits of code that generate dynamic content of the page. In other words java server page, lets you separate the dynamic part of your web page from the static html.

This is obtain by using XML & script written in the java.

* Comparison with Existing technologies

→ JSP vs CGI

Using the common gateway interface i.e, CGI when a request is sent from browser to the web page, an external script (often written in PERL) is run on the server so that the page can be created. Each request involves creating a new process into which the script is run, loaded & unloaded when the request has been processed. Thus CGI takes up valuable system

resources.

Since CGI requires to pass request information to the script, you have to roll your own session support to remember a users state b/w request whereas JSP can maintain state on the server b/w request

→ JSP vs Client-side scripting

Using client side scripting lang. such as java script & VB script being in the issue of varying interpretation by different browser versions, a user decide to disable scripting, all together since JSP runs on server, the browser is not an issue.

→ JSP vs ASP

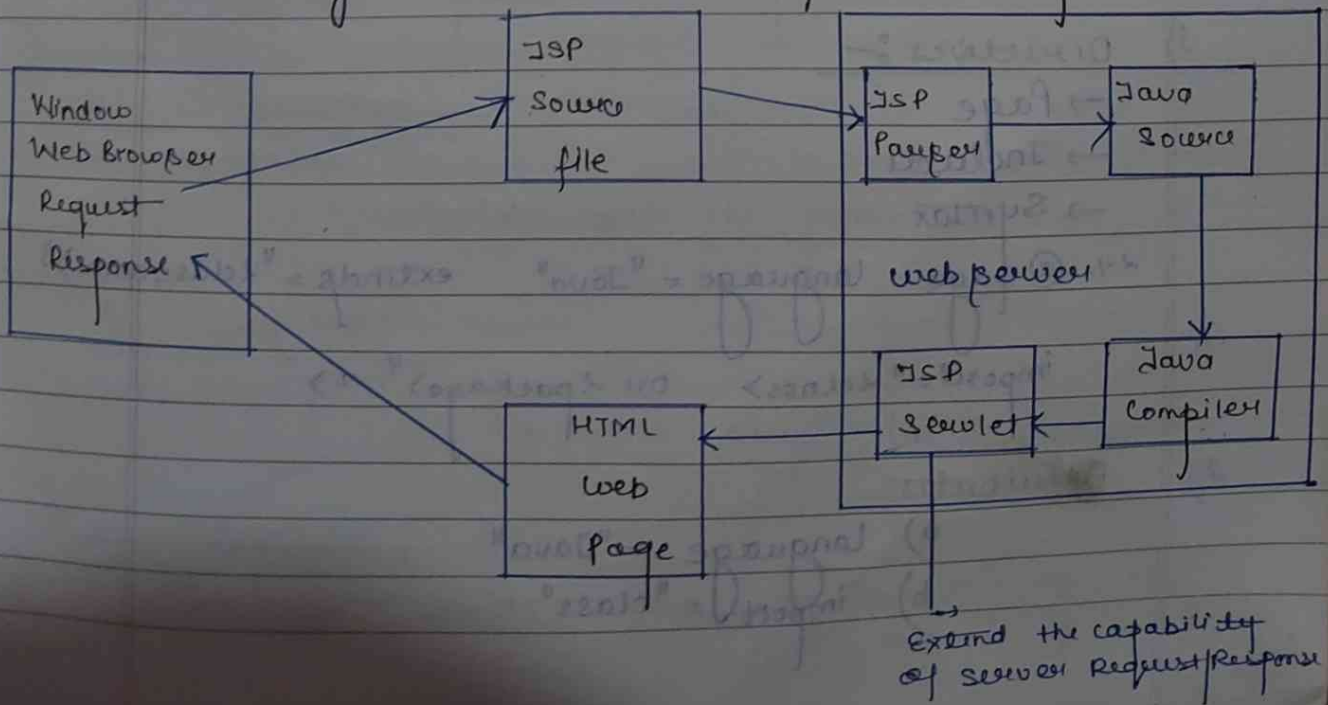
JSP & microsoft Active server page are similar in many respects both are designed to create interactive web pages for web based application to a large intrnt, both JSP and ASP can create web page that separate programming logic from the

page design

Working model of JSP.

JSP is a technology using server-side scripting i.e. actually translated into servlets & compiled before they are run.

JSP page provides tags & allows developers to perform most dynamic content operations without writing complex java code. JSP page are efficient, it loads into the web servers memory on receiving the request very first time & the subsequent call are server within a very short period of time.



JSP pages components.

JSP page is built using the following components

1) Directives
2) Declaration
3) Scriptlt
4) Expression
5) Standard Actions
6) Custom tags

1) Directives :-
   → Page
   → Include
   → Syntax
   <%. @page language = "Java" extends = "<classname>"
   import = "<class> or <package>" %>

2) Attributes
   a) language = "Java"
   b) import = "class"

c) Buffer size = " "

d) scope = " Request | Page | Session | Application

e) And etc

Page :-

Page directives is aimed to define certain attributes JSP page.

Ex- language of the page in which the page content should written, which class to be imported so that it can be used within the JSP page.

Include:-

Syntax

<%@ include file = "<filename>" %>

Attribute

file = "<filename>"

2) Declaration

Syntax

<%! Declare all the variables %>

3) Scriplets :-

Syntax

<% All your script will come here %>

4) Expression

You can write XML equivalent of the

above syntax as follows –

```
<JSP : scriplets>
    code fragments.
</JSP : script>
```

Any ~~text~~ HTML tag : or JSP elements you write must be outside the scriptlet

```
<html>
<head>
<title> Hello world </title>
</head>
<body>
<%
  out.println("Your IP address is" + request.getRemoteAddr());
```

4) Expression :-

A JSP expression element contains a scripting language expression i.e., evaluated, converted to a string & inserted when the expression apears in the JSP file

```
<JSP: expression>
    expression
</JSP : expression>
```

5) Standard actions :-

<jsp:include page="<filename>"/>

6) Custom tags :-

JSP implicit objects.

JSP implicit objects are java objects that the JSP container makes available to developers in each page & developer can call them directly without being explicitly declare.

object	description
Request	This is the HttpServletRequest object associated with the request.
Response	This is the HttpServletResponse object associated with the response to the client.
Out	This is the printwriter object used to send o/p to the client.
Session	This is the Http session object associated with the request

object	Description
Application	This is the ServletContext object associated with application context.
Config	This is the servlet config object associated with the page.
Page context	This encapsulate use of the server specific features.
Exception	The exception object allows the exception data to be accessed by designated JSP

Steps to develop first JSP application (page)

1) Install the Apache tomcat server (sunmicrosystem) or IIS (Microsoft)
Now click start >> All programs >> Apache tomcat server 6.0 >> monitor Tomcat

2) Go to the following path
C:/Program Files/Apache software foundation/Tomcat 6.0/webpages.

3) write a JSP program as follows :-

```
<html>
<body>
<H1> my first JSP program </H1>
<font color="red" size="7"> my first JSP program
    </font> <Br>
<%. for (int i=1; i<=4; i++)
{
 out.write("Hello G.C.R.G <br>");
}
%.>
</body>
</html>
```

* Save above program with name first.jsp at the location

Step 4 C:/Program files/Apache Software foundation/Tomcat 6.0/webpages/my-jsp

Now open Internet explorer or chrome & write following URL

http://localhost:8080/my-jsp/first.jsp

Life cycle of JSP

JSP architecture is based on the servlet architecture actually JSP page is the text file stored on the web-server & accessed via there path. The

following URL is used to open the file first.jsp

```
<html>
<head>
<script type="text/javascript">
function loadXMLDOC()
{
// AJAX script goes here
}
</script>
</head>
<body>
<div id="myid">let AJAX change this content</div>
<button type="button" onclick="loadXMLDOC">change
content</button>
</body>
</html>
```

Vanshita Kashyap