UNIT- III

Java Servlets and XML

★ Difference :

Server side scripting	Client Side Scripting
i) Used to create web pages that provide some service	i) Used to create web pages as request/response to server
ii) These scripts generally run on web server	ii) These scripts generally run on web browser
iii) User's request is fulfilled by running script directly on web server to generate dynamic HTML pages	iii) Processing of these scipt takes place on end-users computer.
iv) Uses : Processing of user request, accessing database	iv) Uses : Making interactive web pages, for interacting with temporary storages like cookies, etc
v) Eg, PHP, ASP.NET, all programming languages	v) Eg, HTML, CSS & Javascript

Study material provided by: Vishwajeet Londhe

Join Community by clicking below links

Telegram Channel

https://t.me/SPPU_TE_BE_COMP

(for all engineering Resources)

SPPU Engineering & Technical UPDATES
WhatsApp channel

WhatsApp Channel

(for all Engg & tech updates)

https://whatsapp.com/channel/0029ValjFriICVfpcV9HFc3b

Insta Page

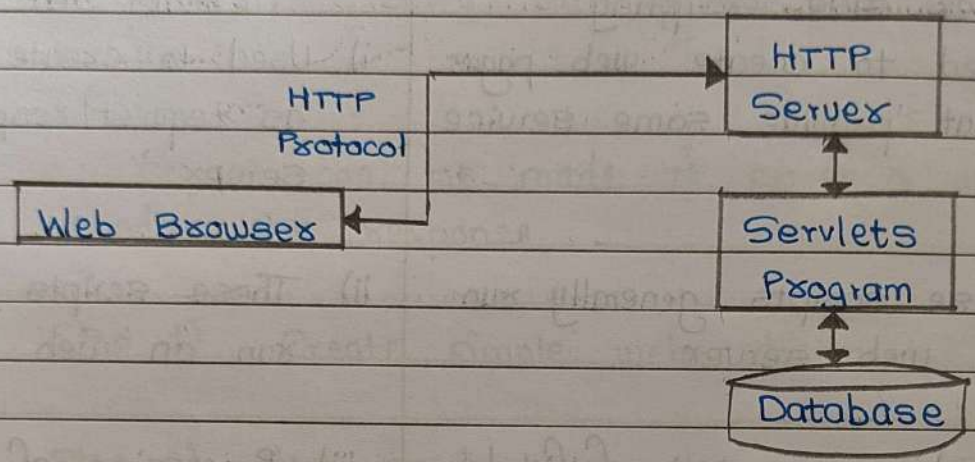
(for all Engg & tech updates)

@SPPU_ENGINEERING_UPDATE

https://www.instagram.com/sppu_engineering_update

★ Servlet Architecture :

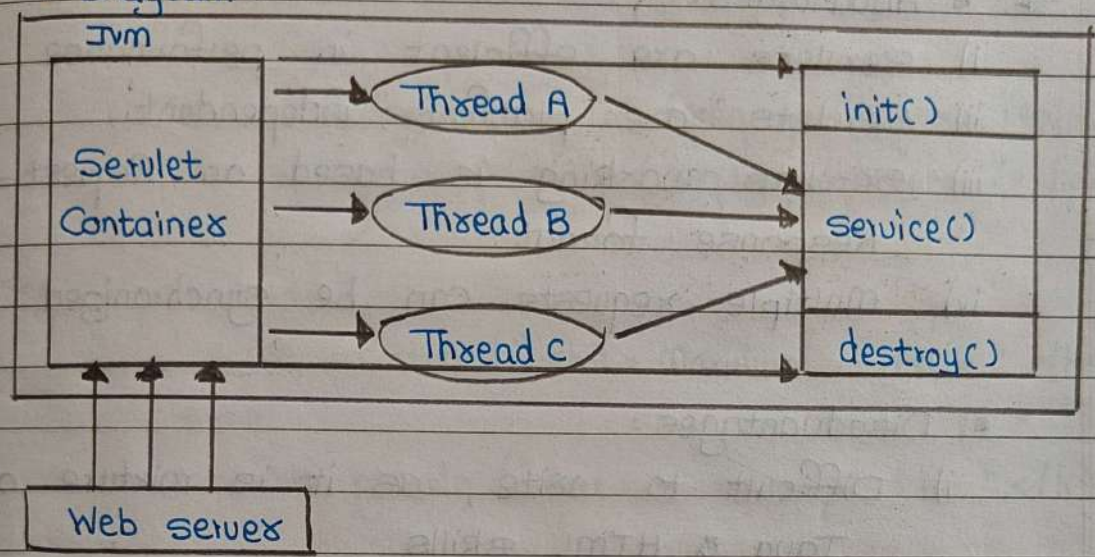
→ • Diagram :-



- Servlets are basically Java programs that run on server
- Servlet class is instantiated when web server begins execution
- Execution of servlet is managed by servlet container, such as TomCat.
- Servlets are most commonly used with HTTP, hence it is also known as "HTTP Servlet"

* Servlet Lifecycle :

→ • Diagram :



- First, the HTTP requests coming to server are delegated to servlet container

- It then loads servlet before Invoking service() method.

- Then, servlet container handles multiple requests by spawning multiple threads, each thread executing service() method of single instance of servlet.

i) Servlet is initialized by calling Init() method
ii) Servlet calls service() method to process client's request
iii) Then servlet is terminated by calling destroy()
iv) Finally, servlet is garbage collected by garbage collector of JVM

Date
Page

★ Advantages & Disadvantages of Servlet -

→ ▪ Advantages:

i) Servlets are efficient in performance
ii) Servlets are platform independent
iii) Servlets working is based on Request-Response format.
iv) Multiple requests can be synchronized

▪ Disadvantages:

i) Difficult to write, as it is mixture of Java & HTML skills
ii) Slows down application execution
iii) Java runtime environment is required to execute.

★ Servlet program which accepts two numbers using POST method & display maximum

→ MaxNumber.java -

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MaxNumber extends HttpServlet
{
    protected void doPost ( HttpServletRequest req,
                            HttpServletResponse res)
        throws ServletException, IOException
```

```
{
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();

    int a = Integer.parseInt(req.getParameter("Number 1"));
    int b = Integer.parseInt(req.getParameter("Number 2"));

    if (a>b)
        out.println("<h4> Maximum :" + a + "</h4>");
    else
        out.println("<h4> Maximum :" + b + "</h4>");
  }
}
```

★ Session :-

- When you open some application, use it for some time & then close it. This entire scenario is named as Session.

★ Cookies -

→ • Cookie is small piece of information that is persisted between multiple client requests

- Generally this information is left on your computer by some advertising agencies on internet.

- A cookie has a name, a single value & optional attributes such as a comment, maximum age, etc.

- Syntax to create cookies in Servlet:
  i) Cookie()
  ii) Cookie(String name, String Value)

- Types of cookie are:
  i) Non-persistent cookie -
     It is valid for single session only.
     It is removed each time when user closes browser

  ii) Persistent cookie -
     It is valid for multiple session.
     It is not removed after user closes the browser.

- Creation of cookies & adding to response object:

```
Cookie cookie = new Cookie("user","dk");
response.addCookie(cookie);
```

- Reading cookie:
  Value from cookie can be obtained using getName() & getValue() methods

```
int n = my.cookies.length;
for (int i=0; i<n; i++)
{
```

```
        String name = my_cookies[i].getName();
        String value = my_cookies[i].getValue();
    }
```

* URL Rewriting -

→ • If the client has disabled cookies in browser then session management using cookies won't work.

• In that case, URL Rewriting can be used as backup

• In URL Rewriting, a token (parameter) is added at end of URL

• Token consists of name/value pair seperated by equal (=) sign

• Eg, hello ? id = 123 & user = alex

• When the user clicks on URL having parameters, the request goes to web Container with extra bit of information at end of URL.

• The web container will fetch the extra part of requested URL and use it for session management.

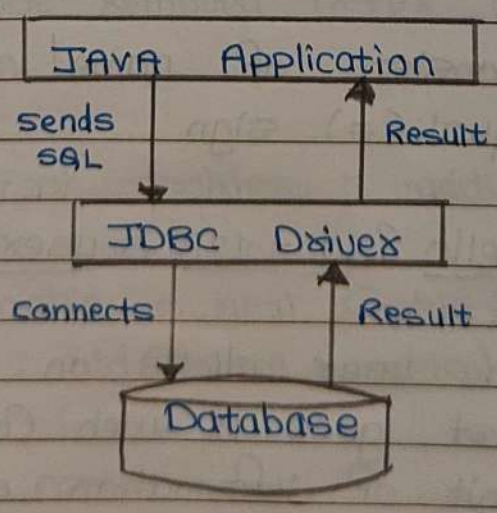
- Advantages -
  i) It will always work whether cookies is disabled or not
  ii) Extra form submission is not required on each pages

- Disadvantages:
  i) Works only with links
  ii) It can only send textual information.

★ JDBC -

→ • JDBC stands for Java DataBase Connectivity
  • It is nothing but an API.

- Working -



i) Java Application establishes connection with data source
ii) Then it invokes classes/interfaces from JDBC driver for sending queries to data

iii) JDBC driver connects to corresponding database & retrieves result

iv) These results are returned to Java application for processing.

★ XML :-

→ • XML stands for eXtensible Markup Language

• This scripting language is similar to HTML, however it is used to transport or store the data

• XML is used to display the meta contents i.e it describes content of document

▪ Need of XML :

• For a data to be transferred from one machine to another, it needs to be converted in compatible format first.

• Hence, we need XML, as it doesn't depend on platform & language thereby making it easy to transfer the data.

- Strengths of XML:

i) XML document is human readable & we can edit any XML document in simple text editor

ii) XML document is language neutral
i.e Java program can generate XML document & it can be parsed by Perl

iii) Every XML document has a tree structure. Hence, complex data can be arranged systematically

iv) XML files are independent of OS.

- Declaration Syntax of XML:

```
<? xml version = "1.0" encoding = "UTF-8" ?>
```

- Structure:

XML documents are hierarchical, containing start tag, content & end tag. Eg,

```
<book>
    <title> Harry Potter </title>
    <author> J. K. Rowling </author>
</book>
```

- XML allows us to create different elements with common name, known as Namespace

Eg,

```
<File - Description>
    <text fname = "input.txt">
        <describe> Text file </describe>
```

```
    </text>
    <text fname = "flower.jpg" >
        <describe> Image File <describe>
    </text>
</ File - Description>
```

★ DTD :-

→ • Document Type Definition (DTD) is used to define basic building block of any xml document.

• Using DTD, we can specify various elements types, attributes & their relationship with one another

▪ Elements -

• Elements are main building block of both XML & HTML documents

• Eg, of XML document could be "note" & "message"

• Elements can contain text, other elements or be empty

▪ Attributes -

• Attributes are generally used to specify values of elements

• Eg < flag type = "True" >

- Types of DTD -

i) Internal DTD:

- DTD is declared inside the XML file, and wrapped inside <!DOCTYPE> definition

- Eg,

```
<?xml version="1.0" ?>
<!DOCTYPE note [
<!ELEMENT note (to, from)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
]>

<note>
  <to> Darshan </to>
  <from> DR </from>
</note>
```

ii) External DTD:

- In this, a seperate DTD file is maintained along with XML file
- Name of DTD file must be specified in XML

- Eg,

XML file	DTD file
<?xml version="1.0" ?>	<!ELEMENT note (to, from)>
<!DOCTYPE note SYSTEM "note.dtd"> <note> <to> Darshan </to> <from> DR </from> </note>	<!ELEMENT to (#PCDATA)> <!ELEMENT from (#PCDATA)>

★ Comparison -

DTD	XML Schema
i) It doesn't support namespace	i) It supports namespace
ii) It is comparatively harder than XSD	ii) It is relatively more simpler than DTD
iii) It doesn't support datatypes	iii) It supports datatypes
iv) It is not extensible in nature	iv) It is extensible in nature

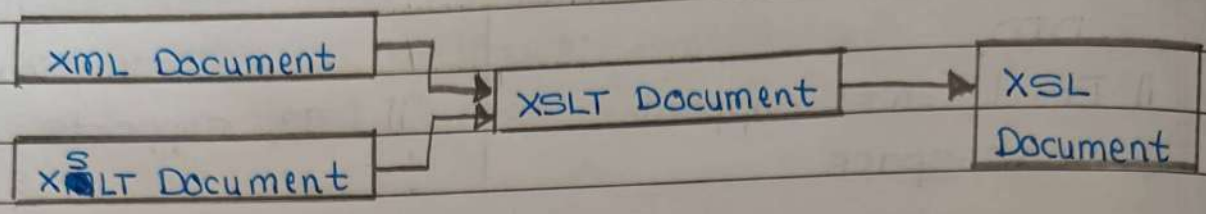
★ XML Transformation - (better than CSS)

→ • XSLT (extensible Stylesheet Langugge Transformation) is standard style sheet language for XML.

• XSLT helps to add/remove elements & attributes to/from output file

• XSLT is used to transform XML data from one format into other.

- Process :-



- XSLT processor takes two input document, one is XML document & other is XSLT document.

- XSLT document is nothing but a program and XML document is input data.

- Then, some or whole part of XML document is selected, modified & merged with XSLT program document in order to produce new document

- This new document is provided as input to XSLT processor which in turn produces another document called XSL document.

* Comparisons :-

XML	XSLT
i) XML stands for extensible markup language	i) XSLT stands for extensible stylesheet language transformation
ii) XML is used to store data in structured format	ii) XSLT is used for transforming & formatting XML file
iii) XML does not perform transformations	iii) XSLT can transform one XML format into another XML / HTML
iv) XML contains user defined tags	iv) XSLT contains special XSL tags.

* AJAX -

→ • AJAX is Asynchronous Java Script & XML

Here,

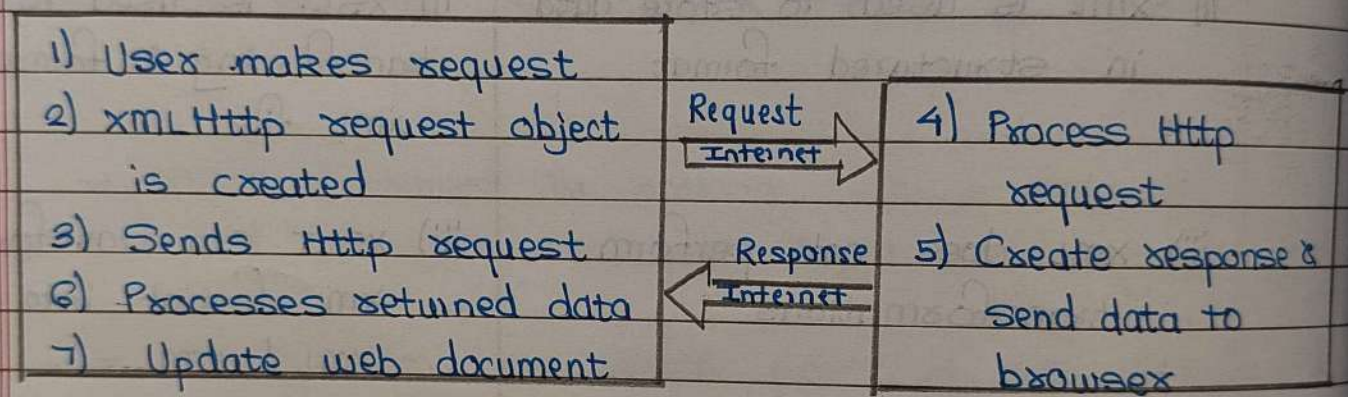
Asynchronous means execution of script does not disturb user's work

Javascript because it makes use of JS to do actual work

XML because along with JS, XML is also supported.

- It allows developer to exchange data with server & updates part of web document without reloading web page

- Working -



- When user makes request, browser creates object for HttpRequest & request is made to server over internet

- Server processes this request and sends required data to browser

- At browser side, returned data is processed using JS & web document gets updated accordingly.

- Example,
  Google Maps, Gmail, Yahoo etc make use of AJAX.

UNIT - IV

JSP and Web Services

★ JSP -

→ • Java Server Pages (JSP) technology is used to create web application just like Servlet.

• A JSP page consists of HTML tags & JSP tags.

• It provides some additional features such as Expression Language, Custom Tag, etc.

• Advantages of JSP over Servlet -

i) Extension to Servlet:
- We can use all features of servlet in JSP
- In addition, we can use
  - implicit objects
  - predefined tags

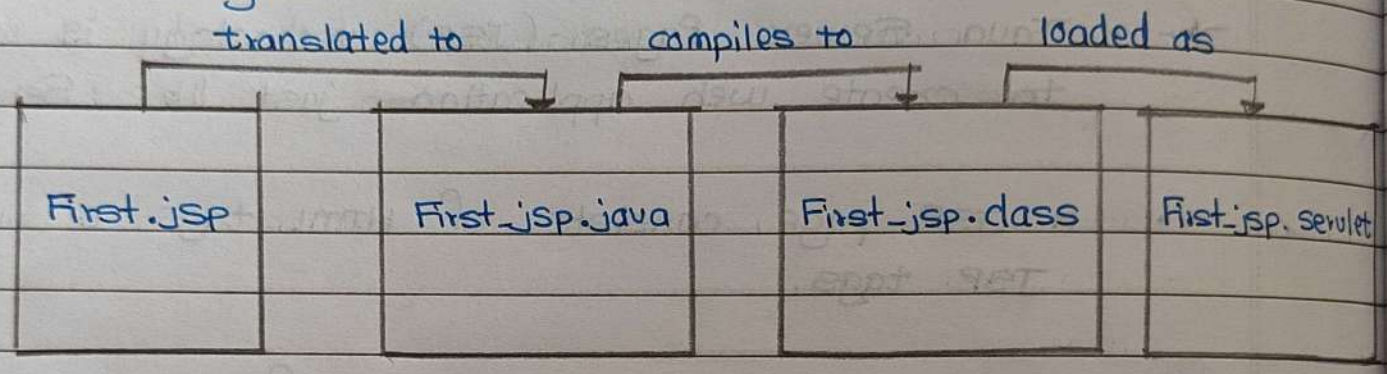
ii) Easy to maintain -
- JSP can be easily maintained because we seperate business logic with presentation logic

iii) Fast Development: No need to recompile and redeploy

iv) Less code than Servlet

★ JSP Life Cycle -

→ • Diagram -



• JSP pages follow these phases:

1) Translation of JSP page

2) Compilation of JSP page

3) Class loading (class file is loaded by class loader)

4) Instantiation (object of Generated Servlet is created)

5) Initialization (jspInit() method is invoked by container)

6) Request Processing (_jspService() method is invoked by container)

7) Destroy (jspDestroy() method is invoked by container)

- In the end, a JSP becomes Servlet

★ JSP Program to demonstrate use of page directive, scriptlet, expression & comment -

Page Directive →

```
<%@ page language = "java" contentType = "text/html" %>
<!DOCTYPE html>
<html>
<head>
    <title> JSP Example </title>
</head>
<body>
    <!-- This is JSP comment -->

    <%-- This is another comment --%>

    <%
        // This is scriptlet
        String message = " Hello ";
        out.println ("<h1>" + message + "</h1>");
    %>

    <p>
        <!-- This is JSP expression -->
        <%= " Current data : " + new java.util.Date() %>
    </p>
</body>
</html>
```

★ Comparison -

Include directive	Include Action
i) Includes file at transition time	i) Includes file at run time
ii) If included file is changed but not JSP, then changes will not be reflected	ii) If included file is changed but not JSP, changes will be reflected
iii) Syntax - <%@ include file = "filename" %>	iii) Syntax - <jsp: include page = "filename"/>
iv) We cannot pass parameter to included file	iv) We can pass parameter to included file using param action

★ Simple JSP page that demonstrates use of implicit objects -

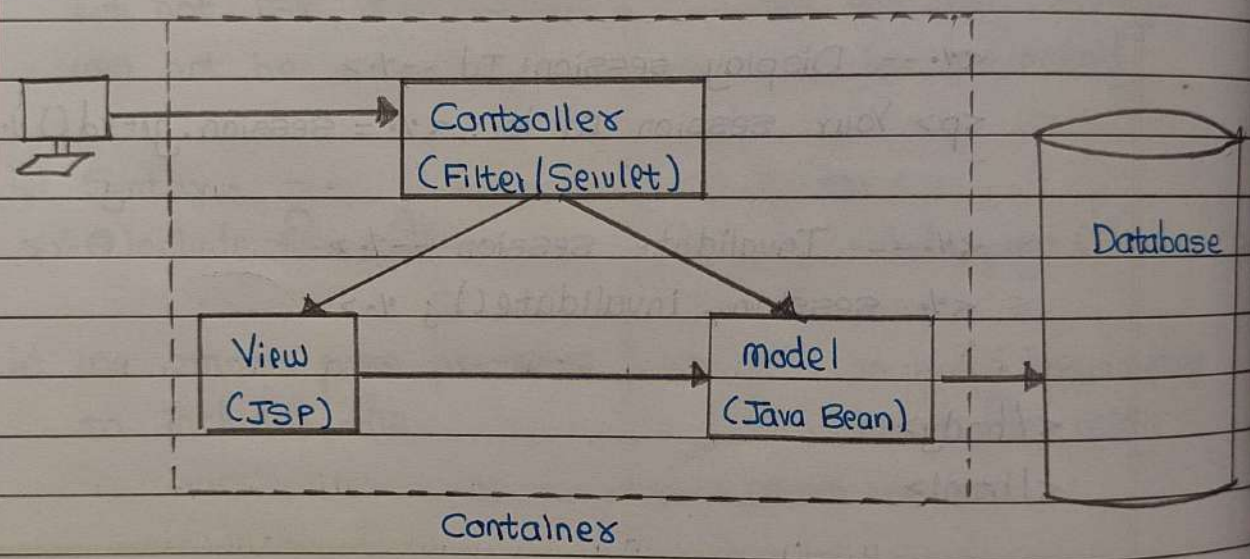
```
<%@ page language = "java" contentType = "text/html" %>
<! DOCTYPE html>
<html>
<head>
    <title> Implicit example </title>
</head>
<body>
```

```
<%-- Set session attribute --%>
<% Session.SetAttribute ("username", "Dk"); %>

<h1> Welcome to Website! </h1>

<%-- Retrieve session attribute & display --%>
<p> Hello, <% = Session.getAttribute ("username") %>! </p>

<%-- Display session Id --%>
<p> Your session Id is : <% = Session.getId() %></p>

<%-- Invalidate session --%>
<% session. invalidate(); %>

</body>
</html>
```

★ JSP - Support to Model - View - Controller :

→ • MVC stands for model, View & Controller. It is design pattern that seperates business logic, presentation logic & data.

• Diagram -



• Advantages -
i) Navigation control is centralized
ii) Easy to maintain large application

① Model Layer -
• This is the data layer which consists of business logic of system
• It consists of all data of the application
• The model connects with database and stores the data into database which is connected to it.

② View Layer -

- This is the presentation layer, consisting of HTML, JSP, etc into it
- It is used to display data which is fetched from controller which in turn fetching data from model layer classes

③ Controller Layer -

- It acts as an interface between View & Model
- It receives the request from view layer & processes request & does necessary validation for request
- This request is further sent to model layer for data processing
- After processing, it sends back to controller with required information & displayed accordingly by view.

Date
Page

★ Web Services & Benefits -

→ • The Web Services are the software systems that are displayed by web browsers using web protocol

• These software systems are used by some software applications rather than end-users directly.

▪ Benefits -

i) Scalability :
Web services are scalable because they operate over standard internet protocols

ii) Reusability :
Once a web service is developed & deployed, it can be used by multiple clients across different applications

iii) Accessibility :
Web services can be accessed over the internet from anywhere.

iv) Platform - independent :
They can be implemented & accessed using various programming languages.

v) Security :
They support various security mechanisms like SSL/TLS encryption, OAuth, etc.

★ WSDL -

→ • WSDL is Web Service Descriptor Language which is based on XML

• The purpose of this file is to describe web service & method of accessing it.

• Following elements are described in WSDL file:

i) Types -

• If we declare & use some variable in web service then their data types can be specified.

ii) Messages -

• The message defines part of the element along with data types.

• The message name specifies name of function

• Eg,

```
< message name = "My_Interface_Hello">
    <part name = "String_1" type = "xsd:string"/>
```

iii) Port Types -

• It defines various operations that can be performed by web service

• Eg, one-way, request-response

iv) Binding -

- Binding defines the message format and details of protocol
- There are 3 main elements:
  Binding, soap:binding & operation

v) Service:

- Service element specifies name for overall web service.

★ SOAP -

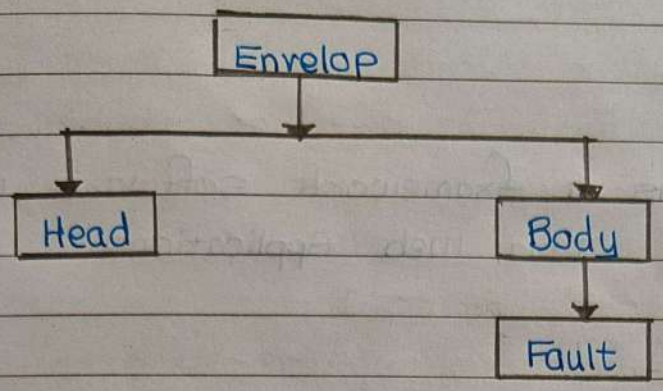
→ • Simple Object Access Protocol (SOAP) is a protocol based on XML

- It is used by web services for exchange of information

- This exchange of information is done over HTTP.

- SOAP allows to talk different applications that are running in two different OS.

- Structure of SOAP -



i) Envelop :
- It is root of any SOAP document
- It defines 2 things : Namespace & encodingStyle

ii) Header :
- It contains 3 attributes :
  mustUnderstand, actor & encodingStyle

iii) Body :
- It contains the actual SOAP message

iv) Fault :
- It is used to represent error code & is optional.

Date
Page

★ Struts -

→ • Struts is a framework software used for developing Java Web Applications.

- Features -

1) It is based on MVC architecture

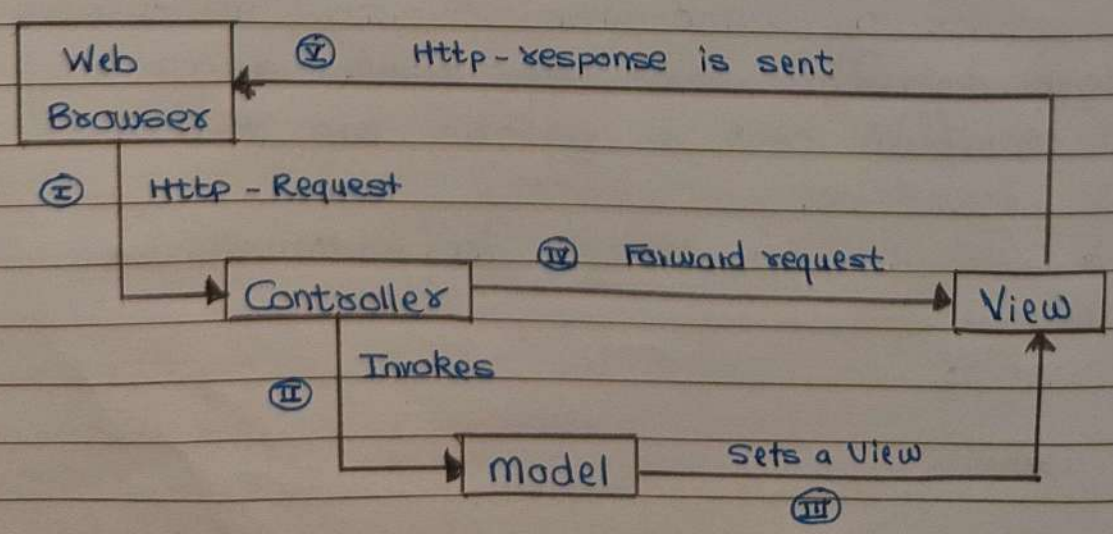
2) Action class is POJO (Plain Old Java Object)

3) Struts2 provides support to AJAX technology

4) Struts 2 framework can be integrated with other applications.

5) Struts 2 support multiple views for same application.

★ Struts Architecture -



(I) Normally, when client wants any web page, he demands it using get/post request by clicking Submit button, then controller is invoked

(II) Controller takes user input and invokes the model

(III) Model stores user's data in database & returns some result to controller

(IV) Using result, controller figures out user requirements & accordingly forwards result to View page

(V) Finally required web page can be obtained as response.

★ Struts Configuration -

→ • There are two main configuration files used in Struts :

i) struts.xml -
• This file contains configuration information that you will be modifying as actions are developed

- <struts> is root tag element, under which we declare different packages using <package> tags

- This file can be used to override default settings for an application.

ii) web.xml file -

- It is a J2EE configuration file that determines how elements of HTTP request are processed by container

- This file provides an entry point for any web application.

★ Actions -

→ • Actions are core of Struts2 framework, as they are for any MVC framework.

- Each URL is mapped to actions

- Action performs 2 tasks -

i) They play important role from request to view. This view is written in JSP

ii) Action directs framework to display particular view as response to request.

★ Interceptors -

→ • Interceptor is an object which intercepts an action dynamically

• In struts2, interceptor is used to perform operations such as validation, exception handling, etc.

• Types of interceptors -

i) Alias : It converts similar parameters that have different names

ii) checkbox : It is used to handle check boxes in form

iii) cookie : It adds a cookie to current action

iv) createSession : It creates HttpSession object if it doesn't exist.

v) debugging : It provides support for debugging

vi) fileUpload : It provides support for file upload in struts 2

vii) timer - It outputs time needed to execute an action

* Exception Handling -

→ • Struts provides easier way to handle uncaught exception & redirect users to dedicated error page.

• Struts make exception handling easy by use of "exception" interceptor

• Eg, create a file called Error.jsp

```
<html>
    <body>
        This is error page
    </body>
</html>
```

Configure struts to use this error page in case of exception

Modify struts.xml by adding following line -

```
<result name = "error" > / Error.jsp </result>
```

UNIT-V

Server Side Scripting Languages

- **Definition -**

  PHP :

  - PHP stands for Hypertext Preprocessor
  - PHP is server-side scripting language. It is mainly used for form handling & database access.

* Control Statement -

→ • Control statement in PHP are similar to control statements that are used in C.

i) If...else :

- Syntax -

```
If (condition)
    statements;
else
    statements;
```

- Eg,

```
<? php
$n = 5;
If ( $n % 2 == 0)
  Echo " Number is even";
Else
  Echo " Number is odd";
?>
```

Date
Page

ii) while loop -

- Syntax -

```
while (condition is true)
{
    code;
}
```

- Eg,

```
<?php
$x = 1;
while ($x<=5) {
    echo " Number is : $x <br>";
    $x++;
}
?>
```

iii) For loop -

- Syntax -

```
for ( init counter; test counter; increment)
{
    code;
}
```

- Eg,

```
<?php
for ($x =0; $x <= 10; $x++)
{
    echo " Number is : $x <br>";
}
?>
```

Date
Page

* Arrays in PHP -

→ • Array is a collection of similar types of element, but in PHP you can have elements of mixed type.

• In each PHP, each elements has two parts key and value

▪ Creating Array -

• For creating an array in PHP, we use array() construct

• Syntax -

```
$array_name = array (value)
```

▪ Adding Elements -

• In PHP, arrays are dynamic i.e they can grow/shrink in size

• We can add element in array using key/index that hasn't been used

Eg, `$name[5] = "DK";`

▪ Deleting Elements -

• We can delete elements from an array using unset() function

• Eg, `unset ($array[2]);`

Date
Page

★ Types of Arrays -

→ ① Indexed Array -

- Indexed array are the arrays with numeric index.

- There are 2 ways to create indexed arrays:

i) Index can be assigned automatically :

```
$cars = array("Volvo", "Bmw", "Jeep");
```

ii) Index can be assigned manually :

```
$cars [0] = "Volvo";
$cars [1] = "Bmw";
$cars [2] = "Jeep";
```

② Associated Array -

- Associated arrays are the arrays with named keys.

- It is the kind of array with name & value pair.

- There are 2 ways to create Associative array :

i) `$age = array("Peter" => "35", "Dk" => "20");`

ii)
```
$age ['Peter'] = "35";
$age ['Dk'] = "20";
```

Date
Page

③ Multidimensional Array -

- A multidimensional / two-dimensional array is an array of arrays.

- PHP understands multidimensional arrays that are two, three, four or more levels deep

- Example -

```
$cars = array (
        array ("volvo", 22, 18)
        array ("BMW", 15, 13));
```

★ Functions in PHP -

→ • Function in PHP are very much similar to functions in C

- Syntax -

```
function name_of_function (parameters)
{
    statement to be executed
    :::
}
```

- Eg,

```
<?php
function myfunc()
{
    print " <i> This is myfun() </i> ";
}
print " <b> Function Program </b> ";
```

```
myfun();
?>
```

- The return statement is used to return some value from function body

- Eg,
```
<?php
function Addition()
{
    $a = 10;
    $b = 20;
    $c = $a + $b;
    return $c;
}
```

- Parameterized function -
  - Eg,
```
<?php
function Add ($a, $b) {
    $sum = $a + $b;
    echo "Sum is " $sum";
}
Add(10,20);
?>
```

classmate

Date
Page

* Cookie & Session Management -

→ • A cookie is a small file that server embeds on user's computer.

• With PHP, you can both, create & retrieve cookie values

• Cookie is created in PHP with setcookie() function
setcookie(name, value, expire_period, path, domain)
Only the name parameter is required, rest are optional

• Eg, <?php
setcookie ("name", "Dk", time() + (86400*30), "/");
?>

• Session -
• When you open some application, use it for some time then close it. This entire scenario is named as 'session'.

• A session is started with session_start() function

• Session variables are set with PHP global variable:
$ SESSION

- Eg,
```
<?php
session_start(); ?>
<html>
<body>
<?php
$SESSION["favcolor"] = "green"; // set variable
echo " Session variable set ; ?>
</body>
</html>
```

★ Connecting MySQL with PHP -

→ • The PHP function mysql_connect connects to the MySQL server

- There are 3 parameters that can be passed to this function

- Eg, mysql_connect ("localhost", "root", "mypassword");

  where, localhost is the localhost on which mysql is running

- Code -
```
<?php
// Create connection
$conn = new mysqli ("localhost", "root", "");
// MySQLi extension (i stands for improved)
```

```
//Check connection
if (!$conn) {
    die ('Could not connect:'.mysqli_connect_error());
}
echo 'Connected successfully <br />';
?>
```

★ WAP -

→ • Wireless Access Protocol (WAP) is a set of standards designed to extend Internet services to mobile phones.

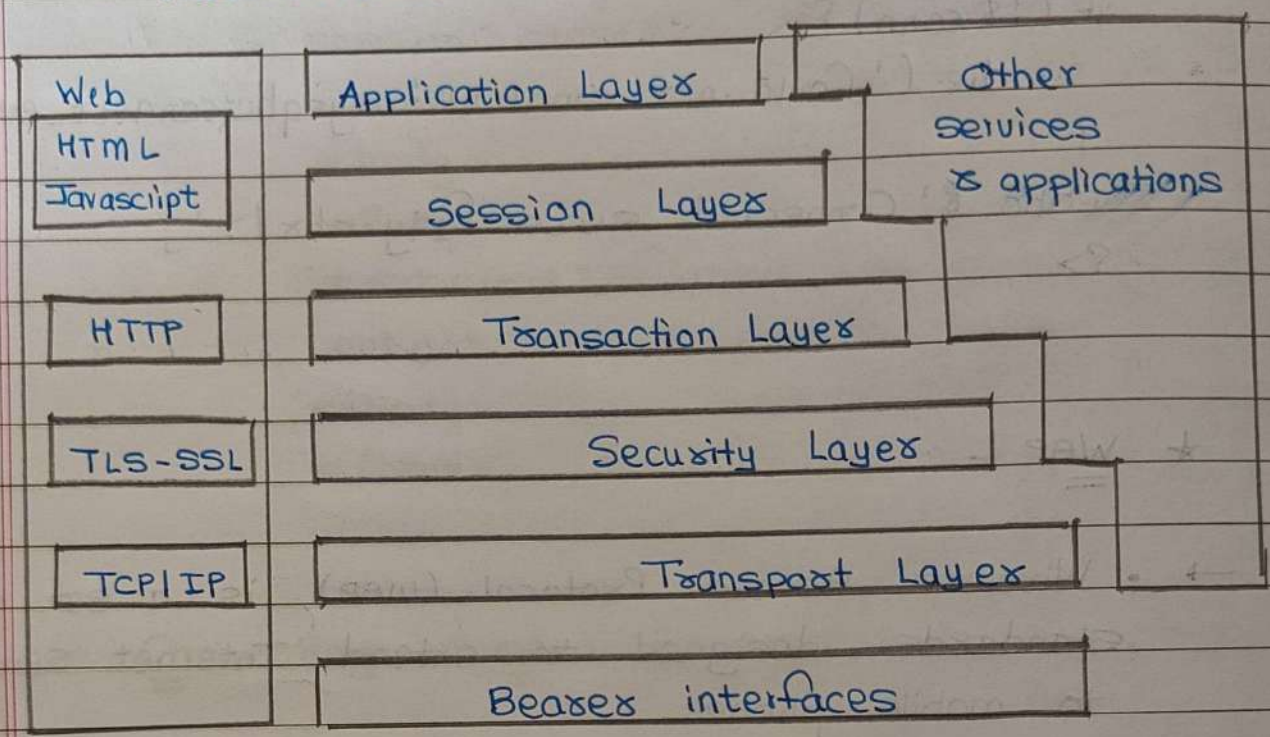
• The mobile network gets connected to Internet through WAP Gateway.

• Function of WAP gateway is to convert a WAP request to WEB request or vice-versa

• The mobile phones act as a client when connected to internet.

• The demanded web pages can be displayed on mobile phones if it is WAP enabled.

- Architecture -



i) Application Layer -
- WAP's application layer is Wireless Application Environment (WAE)
- WAE supports WAP application development using WML

ii) Session Layer -
- This layer contains WSP which provides fast connection & reconnection

iii) Transaction Layer -
- This layer contains WTP which provides transaction level services

Date
Page

iv) Security Layer -

- This layer contains Wireless Transaction Layer Security (WTLS), which offers data integrity, privacy.

v) Transport Layer -

- This layer contains Wireless Datagram Protocol (WDP) which offers consistent data format to higher levels

★ WML -

→ • WML stands for Wireless Markup Language

- Role of WML is same as that of HTML. WAP sites are written in WML, while web sites are written in HTML

- WML is basically the web language for making sites on mobile phones.

- WML follows a deck and card
  - A WML document is made up of multiple cards
  - Cards can be grouped together into a deck
  - A WML deck is similar to an HTML page

- WML elements -

i) text :

ii) hyperlink (anchor) : <a href=...> </a>

iii) action - <do> </do>

iv) input user text - <input/>

v) value of variable display - $(...)

vi) image - <img src = ... />

vii) set variable - <postfield name = ...

- Eg,

```
<?xml version = "1.0" ?>
<wml>
    <card id = "one" title = "First Card">
    <p>
        This is first card in deck
    </p>
    </card>

    <card id = "two" title = "Second Card">
    <p>
        This is second card in deck
    </p>
    </card>
</wml>
```

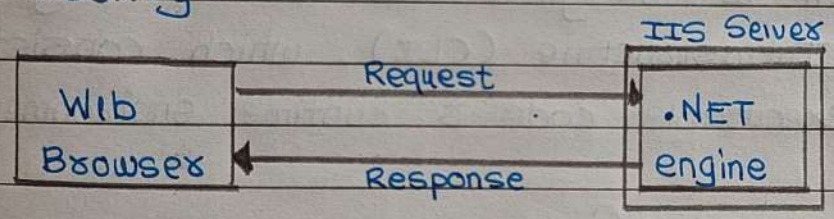
* ASP.NET -

→ • Active Server Pages (ASP) is a server side scripting language

• ASP.NET is a server side scripting technology offered by Microsoft

• It is a powerful scripting language for creating dynamic & interactive web pages.

▪ Working -



I) When user demands for ASP.NET file, web browser of client's PC sends this request to IIS server

II) IIS server passes this request to .NET engine

III) .NET engine reads the file line by line & generates script in a file

IV) The requested .NET file is returned to web browser.

- The ASP.NET application programs can be written in one of following languages:

i) C# ii) Visual Basic .NET
iii) Jscript iv) J#

★ C# -

→ • C# is a modern, object oriented programming language developed by Microsoft.

- C# is designed for Common Language Infrastructure (CLI), which consists of executable code & runtime environment

- Features of C# :

i) Object oriented
ii) Standard Library
iii) Automatic Garbage Collection
iv) Properties & Events

- C# is case sensitive & every statement is terminated by semicolon

- C# code -

```
using System;

namespace MyFirstCode
{
    class Program
    {
```

```
        static void Main (string[] args)
        {
            Console.WriteLine ("Welcome");
        }
    }
}
```

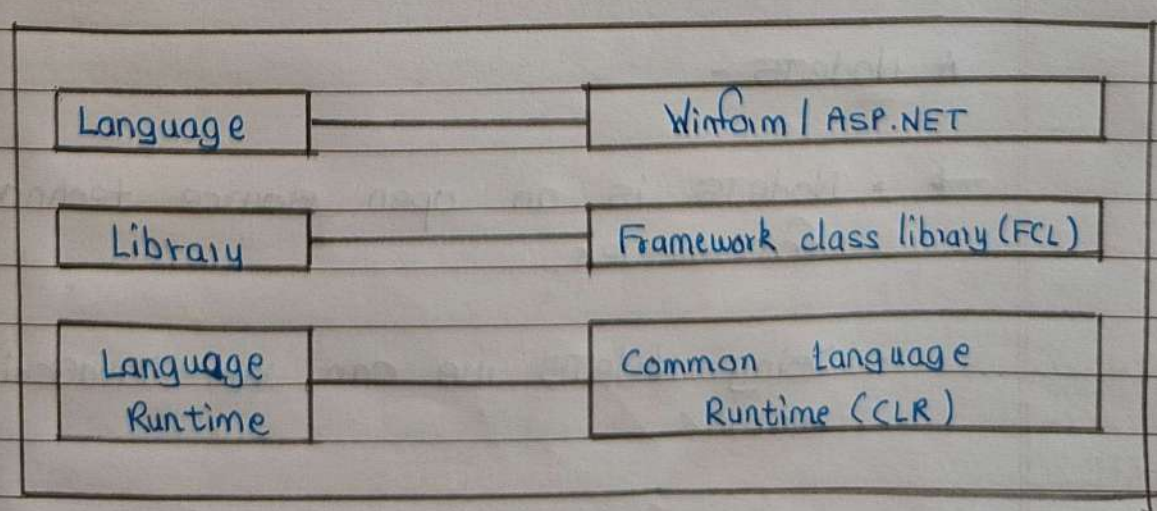
★ .NET framework -

→ • .NET framework is a software development platform developed by Microsoft

• It is designed for building & running applications

• It supports various programming languages like C#, VB.NET, etc.

■ Architecture -



Date
Page

i) CLR -

- It stands for Common Language Runtime.
- It is a platform on which .NET programs are executed.
- CLR also performs other tasks such as Exception handling, memory management, etc.

ii) Framework Class Library (FCL) -

- It is a standard library which is actually collection of various classes, interfaces, etc used to build application

iii) Language -

- .NET supports many languages
- Various types of applications that can be developed using .NET:
  i) Winforms
  ii) ASP.NET
  iii) ADO.NET

★ NodeJS -

→ • NodeJS is an open source technology for server

- Using NodeJS we can run JavaScript on server

- It runs on various platforms such as Windows, Linux, MacOS, etc.

- Features -

1) Non blocking thread execution -
We can continue executing next task despite waiting for response for something during current task

2) Efficient -
NodeJS is built on V8 Javascript engine & is very fast in code execution

3) No buffering -
Node.js applications never buffer data

4) Single threaded & Highly Scalable.

UNIT - VI

Ruby And Rails

* Ruby and Features -

→ • Ruby is a pure object oriented programming language

• Ruby is an open source & it is freely available on web

• Ruby is similar to Perl, Python

• Ruby executes on variety of platforms such as Windows, Mac, Linux, etc

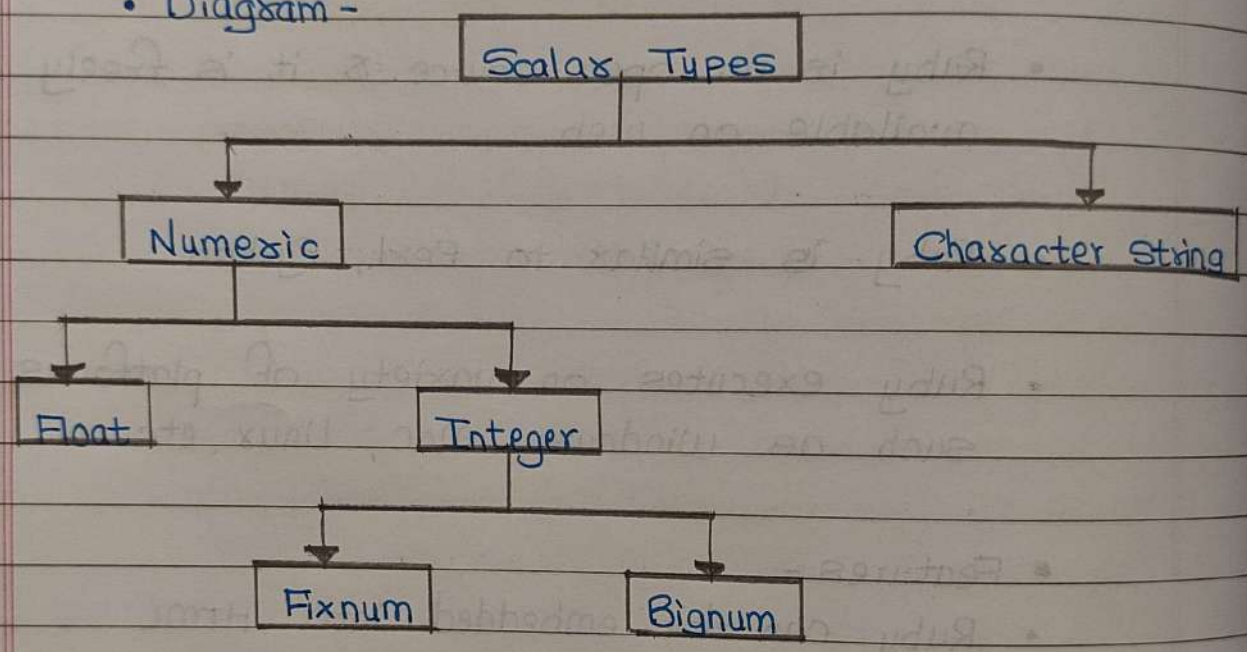
• Features -

• Ruby can be embedded into HTML
• Ruby is highly scalable
• Ruby has clean & easy syntax that allows newbie to learn quickly
• Ruby can be easily connected to MySQL, Oracle, DB2, etc.
• Ruby has rich-set of in-built functions

★ Scalar Types in Ruby -

→ • Scalar Types are classified as Numeric & character strings.

• Diagram -



▪ Strings -

- String is made up of multiple characters
- They are defined by single or double quotes
- Eg, puts "Hello, DK!"

▪ Numbers -

- Integers & Float are two main kinds of numbers
- Eg, my_int = 34
  my_flt = 3.14

Date
Page

★ Operations on Scalar Type -

→ ▪ Operations on Numbers :

i) .even? -
- To check whether or not integer is even
- Returns true/false boolean
- Eg, 15.even? #=> False

ii) .odd? -
- To check whether or not integer is odd
- Eg, 15.odd? #=> True

iii) .ceil -
- Rounds floats up to nearest integer
- Eg, 6.7.ceil #=> 7

iv) .floor -
- Rounds floats down to nearest integer
- Eg, 8.3.floor #=> 8

▪ Operations on String :

i) str.size or str.length -
- Get the length of string
- Eg, "Hello World".length

ii) str.reverse -
- Reverse a string
- Eg, str = "Dk"
  str.reverse

iii) str.upcase -

- Convert each character of string into uppercase
- Eg, "hello worLD".upcase

★ Ruby Arrays -

→ • Arrays in ruby are dynamic i.e array can grow & shrink

- Array in Ruby can store elements of different data types

- Arrays can be created using 2 ways:

i) Using keyword 'new':

eg, Marks = Array.new

Marks = Array.new(10)

ii) Assign the list of literals to variables

eg, Student = [10, 20, 30, "DR", [1]]

- Build-in Methods:

ⓐ 'shift' method removes first element of array

Eg, A = [10, 20, 30]

num = A.shift

ⓑ 'push' method is used to push elements from end in array

Eg, A = [10, 20, 30]
A. push (40, 50)

ⓒ Two arrays can be combined together using '+' operator

Eg, A = [10, 20, 30]
B = [1.1, 2.2, 3.3]
C = A + B

ⓓ Reverse - Reverse contents

ⓔ Sort - Sort contents

* Ruby Methods -

→ • Methods is a set of expression that returns a value

• Ruby methods prevent us from writing the same code in program again & again.

• Defining Method -
• Ruby method is defined with 'def' keyword followed by method name
• At the end use 'end' keyword

• Method name should always start with lowercase letter

Date
Page

- Syntax -

```
def
    methodName
    code...
end
```

- Eg, -

```
def welcome
    puts "Welcome Guys"
end
```

- Parameters -
- There are 2 types of parameters

i) Parameter values that appear in call to a method are called actual parameters

ii) Parameter names used in method, which correspond to actual parameters, are called formal parameters.

★ Classes -

→ • Ruby is purely an object oriented programming language. Hence, 'class' is an important factor

- Every class contains two things, variables & functions

Date
Page

- Class in Ruby is defined as:

```
class className
....
end
```

- Class name must begin with capital letter & class must be terminated by 'end' keyword

- In Ruby, there are two types of variables -

i) Instance variables:
  - Instance variable are defined in method
  - Instance variable change from object to object
  - Syntax - @variable_name

ii) Class variables -
  - Class variable belong to class
  - These are available across objects
  - Syntax - @@ variable_name

- Objects -
  - Objects are instances of class & these can be created using method 'new'

  - Eg, Obj1 = Myclass.new

★ Pattern matching -

→ • Pattern matching is a feature that is commonly found in functional programming language

• For pattern matching in Ruby, the operator =~ is used

• The pattern can be placed within /

• Eg, T = "Success"
=> "Success"

T =~ /es/
=> 4

• The part of a string that matched a part of pattern can be saved in implicit variable for later use.

★ Overview of Rails -

→ • Rails is a server-side web application development framework.

• It is based on MVC architecture i.e Model - View - Controller

• We can develop a web application at least ten times faster with Rails than any other framework

• Most significant ~~approach~~ characteristic of Rails is its approach to connecting object-oriented software with relational database

• Rails can be more commonly used with AJAX.

▪ Advantages -

i) Tooling -

• It provides tooling that helps us to deliver more features in less time

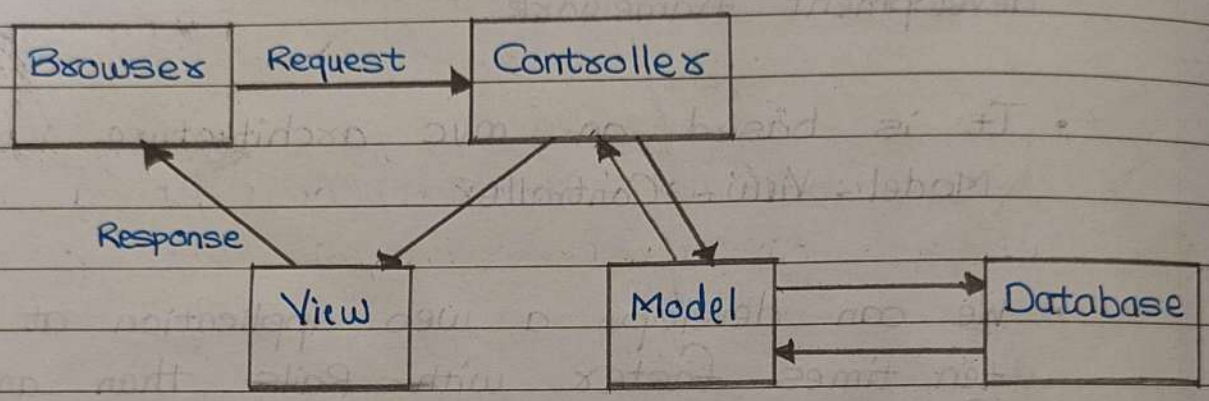
ii) Large Community

iii) Productivity -

• Ruby on Rails is incredibly fast from another language

★ Document Requests -

→ • Diagram -



• Documents are of 2 types :

i) Static Document •

ii) Dynamic Document -

• Dynamic document is a kind of web document in which contents are changing more often

• Dynamic documents can be constructed in Rails by embedding Ruby code in template file.

- Ruby code is embedded in template file by placing it between <% and %> markers.

★ Rails Layout -

→ • The app/View directory of each application has two directories, one is for controller & another is 'Layout'

- In Rails, layouts are pieces that fit together (eg, header, footer, menus, etc) to make a complete view

- Rails automatically pair up layouts with respective controllers having same name

- Layouts are enabled by default in Rails.

- First, we need to define a layout template & then define a path for controller to know that layout exists

- Rails searches for the layout with same name in appl layouts directory as controller's name

- Eg, If you have a controller called 'abController' then rails will search for layouts/ab.html.erb layout.

Date
Page

★ Rails with AJAX -

→ • AJAX stands for Asynchronous JavaScript and XML. It allows client side changes without reloading the page.

• In the basic web architecture -
- The user clicks a link or submits a form

- The form is submitted to server, which then sends back a response

- The response is then displayed for user on new page.

• Rails provide quite a bit of built-in support for building web pages with this technique

• Rails uses a technique called "Unobtrusive JavaScript" to handle attaching JavaScript to DOM.

• This is a technique by which we can mix JavaScript code into HTML.

* EJB -

→ • J2EE application container contains the components that can be used by clients for executing business logic.

• These components are known as Enterprise Java Beans (EJB)

• EJB mainly contains business logic & business data

• EJB application can also be deployed on server, hence it is called server side component also

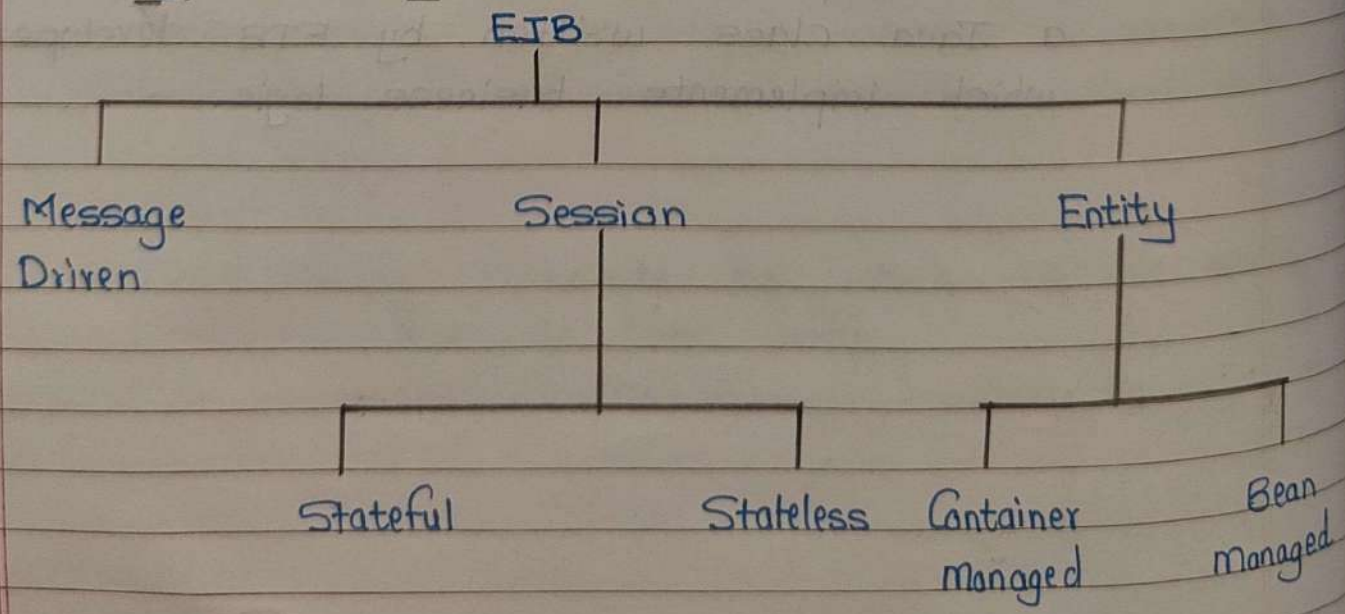
• EJB components always lie in some container which is called as EJB container

• EJB component is an EJB class. It is a Java class written by EJB developer which implements business logic.

★ Comparison -

Java Beans	EJB
i) Components can be visible or invisible to user	i) Components are always invisible to user
ii) Java beans normally run locally	ii) EJB are distributed components & always exist on servers
iii) Java beans can be used as ActiveX control	iii) EJB can't be used as ActiveX Control
iv) JB can be described with properties, BeanInfo & Customizers	iv) EJB can be described using Deployment Descriptor

★ Types of EJB -



Date
Page

i) Message Driven Bean -

- It is similar to session bean but gets activated only after passing message

ii) Session -

- Session beans are used for managing processes or tasks.
- They are client specific

- Subtypes -

ⓐ Stateful :

- Stateful bean is used to store internal state of EJB

ⓑ Stateless :

- Stateless bean is designed for simple & lightweight components
- This brings scalability in server performance

iii) Entity -

- Entity beans are used for modelling the business concepts. i.e they represent data.
- Entity beans are persistent

- Subtypes -

ⓐ Container Managed :

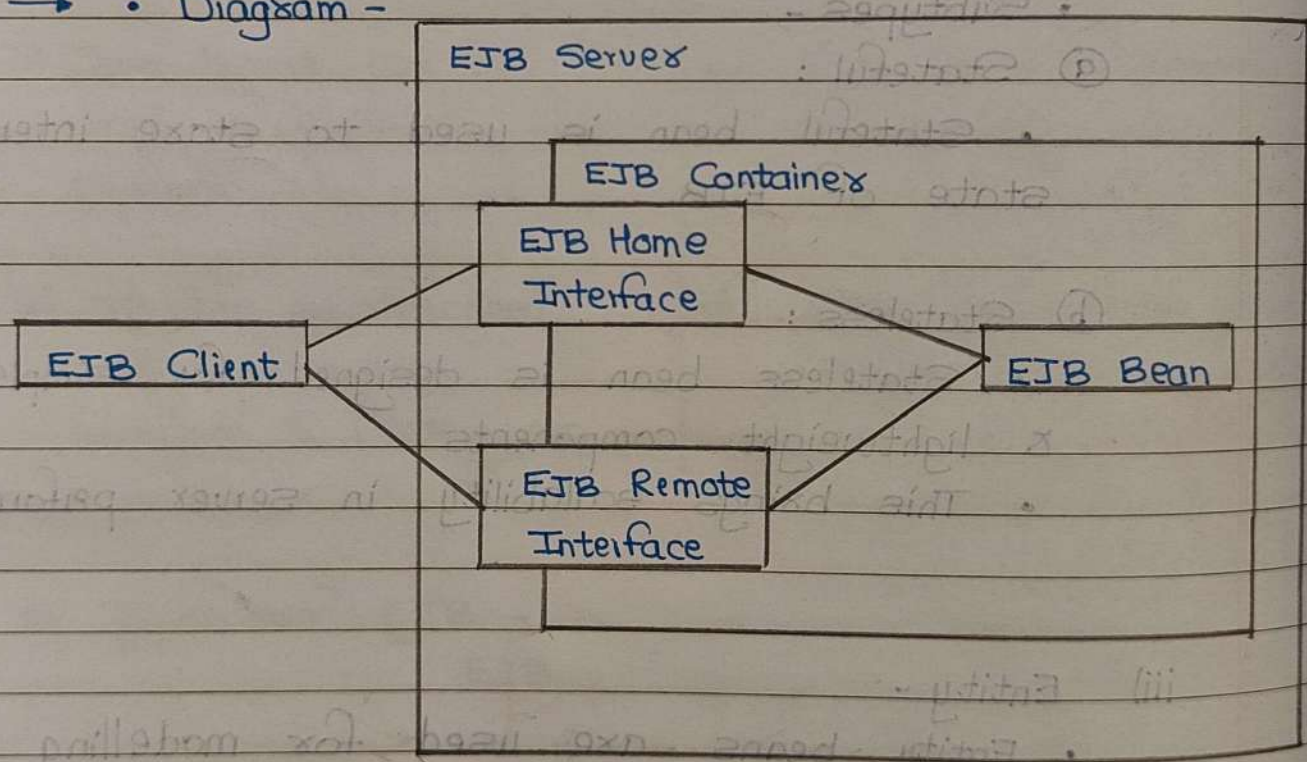
- This bean is used to manage the persistence of entity bean

ⓑ Bean Managed :

- This bean is used for saving bean's state.
- Container does not need to make any database call.

★ EJB Architecture -

→ • Diagram -



1) Application / EJB Server -

- This server is outermost layer in architecture and it holds the container.
- It provides necessary environment to execute applications developed using beans
- Eg, TomCat, JBoss, etc

Date
Page

2) Container -

- It is second outermost layer in architecture & provides following services to Beans :

ⓐ Transactional Service - like registering objects, assign remote interface

ⓑ Monitoring activities of objects & components

ⓒ Security service & pooling of resources

ⓓ Manages life cycle of beans

3) Beans -

- Beans provide business logic for developing robust, secured & large scale business applications

4) Client makes use of these EJB components & when it is no longer required, these components can be returned to pool

- EJB Clients are :

ⓐ Java Applications
ⓑ Java Applets
ⓒ Web services
ⓓ Servlets, JSP, etc.