Unit III

3 Java Servlets and XML

3.1 : Servlet Architecture Overview

Q.1 What is difference between server side scripting language and client side scripting language. ☞ [SPPU : May-22, Marks 5]

Ans. :

Sr. No.	Server side scripting	Client side scripting
1.	The server side scripting is used to create the web pages that provide some services.	The client side scripting is used to create the web pages as a request or response to server. These pages are displayed to the user on web browser.
2.	These scripts generally run on web servers.	These scripts generally run on web browser.
3.	A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser.	The processing of these scripts takes place on the end users computer. The source code is transferred from the web server to the users computer over the internet and run directly in the browser.
4.	Uses : Processing of user request, accessing to databases.	Uses : Making interactive web pages, for interacting with temporary storages such as cookies or local storage, sending request to server and getting the response and displaying that response in web browser.
5.	**Example :** PHP, ASP.NET, nearly all the programming languages including C++, Java and C#	**Example :** HTML, CSS, JavaScript(primarily)

Q.2 Describe servlet architecture in detail. [SPPU : May-22, Marks 4]

Ans. :

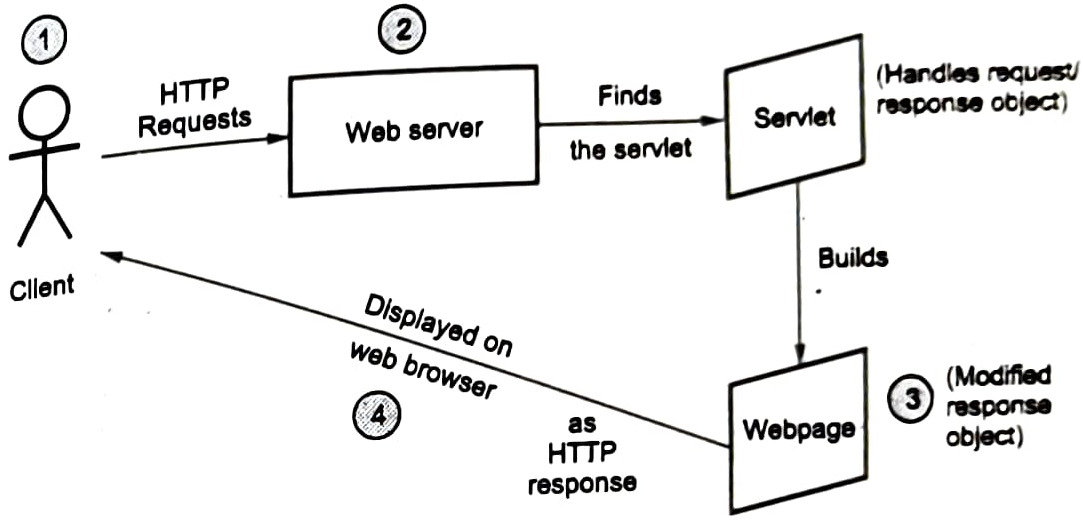


Fig. Q.2.1 How servlet works ?

1. When a client make a request for some servlet, he/she actually uses the **Web browser** in which request is written as a URL.
2. The web browser then sends this request to **Web server**. The web server first finds the requested servlet.
3. The obtained servlet gathers the relevant information in order to satisfy the client's request and builds a **web page** accordingly.
4. This web page is then **displayed** to the client. Thus the request made by the client gets satisfied by the servlets.

Q.3 Enlist any four advantages of servlets.

Ans. : • The servlets are very **efficient** in their performance and get executed in the address space of the belonging web server.

- The servlets are **platform independent** and can be executed on different web servers.
- The servlets working is based on **Request-Response**. Any HTML form can take the user input and can forward this input to the servlet. The servlets are then responsible to communicate with the back-end database and manipulate the required business logic. These servlets embedded on the web servers using **Servlets API.**

- Servlets provide a way to generate the **dynamic document**. For instance : A servlet can display the information of current user logged in, his logging time, his last access, total number of access he made so far and so on.
- **Multiple users** can keep a co-ordination for some application among themselves using servlets.

3.2 : A "Hello World" Servlet

Q.4 Write a Java servlet program which will display "welcome to servlet" message. ☞ **[SPPU : May-19, Marks 5]**

Ans. :

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class FirstServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServletResponse
response)
    throws IOException,ServletException
    {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>My First Servlet</title>");
        out.println("<body>");
        out.println("<h1>Welcome to Servlet</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

3.3 : Servlets Generating Dynamic Content

Q.5 Write a servlet to display current date and time

Ans. :

```
import java.io.*;
import javax.servlet.*;
public class DynamicServletDemo extends GenericServlet
{
    //implement service()
    public void service(ServletRequest req, ServletResponse res)
    throws IOException, ServletException
    {
        //set response content type
        res.setContentType("text/html");
        //get stream obj
        PrintWriter out = res.getWriter();
        java.util.Date date = new java.util.Date();
        out.println("<h2>"+"Current Date & Time: "
                    +date.toString()+"</h2>");
        //close stream object
        out.close();
    }
}
```

Program Explanation : In above program,

1. We have used Date() method of java.util class that is intended to display current date and time.
2. The date is displayed using the **date.toString()** method.

3.4 : Servlet Life Cycle

Q.6 Explain the lifecycle of servlet. Write a simple servlet to print "Hello, Good Day". **[SPPU : March-20, Marks 5]**

Ans. : Life Cycle : In the life cycle of servlet there are three important methods. These methods are,

1. **Init**
2. **Service**
3. **Destroy**

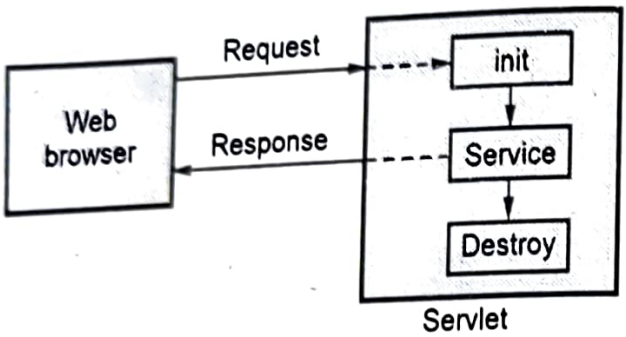


Fig. Q.6.1 Life cycle of servlet

- The client enters the URL in the web browser and makes a request. The browser then generates the HTTP request and sends it to the Web server. (Refer Fig. Q.6.1)
- Web server maps this request to the corresponding servlet.

1. **Init ( ) Method :** The server basically invokes the **init()** method of servlet. This method is called only when the servlet is loaded in the memory for the first time.

   Using this method initialization parameters can also be passed to the servlet in order to configure itself.

2. **service ( ) Method :** Server can invoke the service for particular HTTP request using **service()** method. The servlets can then read the data provided by the HTTP request with the help of **service()** method.
3. **destroy ( ) Method :** Finally server unloads the servlet from the memory using the **destroy()** method.

Example Program

Following is a simple servlet in which these three methods viz. init(), service() and destroy() are used -

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

public class LifeCycle extends GenericServlet
{
  public void init(ServletConfig config) throws ServletException
  {
    System.out.println("init");
  }

  public void service(ServletRequest request, ServletResponse response)
  throws ServletException, IOException
  {
    System.out.println("from service");
    PrintWriter out = response.getWriter();
    out.println("Twinkle Twinkle Little Stars.");
    out.print("How I wonder what you are.");
  }
  public void destroy()
  {
    System.out.println("destroy");
  }
}
```

Program Explanation : In the above program,

1. We have first of all used an **init()** method to which object of the **ServletConfig** interface is passed. This interface allows the servlet to get the initialization parameters.
2. Then we have used the **service()** method to which the ServletRequest and ServletResponse parameters are passed for making the HTTP request and getting the HTTP response from the sevlet.

3. The output stream is created using **getWriter()** and along this output stream some messages are written. These messages will then be displayed on the web browser when the servlet gets executed.
4. Finally the **destroy()** function is written in order to unload the servlet from the memory.

Servlet Program to Print Hello : Refer Q.4.

3.5 : Parameter Data

Q.7 What is GET and POST method ?

Ans. :

i) **GET method :** The GET method sends user information along with ? symbol called query string. For instance

http://localhost/hello?user = aaaa&age = 20

In servlet, this information is processed using **doGet()** method.

ii) **POST method :** This is the most reliable method of sending user information to the server from HTML form. Servlet handles this request using **doPost** method.

Q.8 What is the difference between GET and POST method ?

Ans. :

GET	POST
Using GET request limited amount of information can be sent.	Using POST large amount of information can be sent.
GET request is not secured as information is visible in URL	This is a secured request.
This request is can be bookmarked.	This request can not be bookmarked.
This request is more efficient.	This request is less efficient.

Q.9 How does servlet read form data ?

Ans. : Servlet makes use of following three methods to read the data entered by the user on the HTML form

1. **getParameter()** - You call request.getParameter() method to get the value of a form parameter.
2. **getParameterValues()** - Call this method if the parameter appears more than once and returns multiple values, for example checkbox.
3. **getParameterNames()** - Call this method if you want a complete list of all parameters in the current request.

Q.10 Write a servlet which accept two numbers using POST methods and display the maximum of them.

Ans. :

Step 1 : The HTML document for inputting two numbers is as follows -

NumbersInput.html

```
<html>
<head>
<body>
<div align="center">
  <br> <br>
  <form action="http ://localhost/examples/servlets/servlet/MaxNumber'
method="post">
    Enter First Number :
     <input type="text" value="" name="Number1" size='5'>
     <br/><br/> Enter Second Number :
     <input type="text" value="" name="Number2" size='5'>
     <br/> <br/> <br/>
     <input type="submit" value="Submit">
  </form>
</div>
</body>
</html>
```

Step 2 : The servlet code that handles the post method and finds the maximum of the two input numbers is as follows -

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();

    // get request parameters for userID and password
    int a = Integer.parseInt(req.getParameter("Number1"));
    int b = Integer.parseInt(req.getParameter("Number2"));

    if (a>b)
      out.println("<h4>The maximum number is:"+a+"</h4>");
    else
      out.println("<h4>The maximum number is :"+b+"</h4>");

  }
}
```

Q.11 Write HTML form to read user name and password. This data is sent to the servlet. If the correct user name and password is given then welcome him/her by his/her name otherwise display the message for invalid user.

Ans. :

Step 1 : Create HTML form for accepting user name and password

Input.html

```
<html>
<head>
</head>
<body>
<form      action="http://localhost/examples/servlets/servlet/Welcome"
method ="get">
```

```
User Name:<input type="text" name="uname"/>
<br/>
Password:<input type="password" name="pwd"/>
<input type="submit" value="Submit"/>
</form>
</body>
</html>
```

Step 2 : Create the servlet program to read user name and password and validate it.

Welcome.java

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

public class Welcome extends HttpServlet
{
    public void doGet(HttpServletRequest req,HttpServletResponse res)
    throws ServletException,IOException
    {
        PrintWriter out=res.getWriter();
        res.setContentType("text/html");

        String username=req.getParameter("uname");
        String password=req.getParameter("pwd");
        if ((username=="Ankita")&&(password=="1234"))
            out.print("Welcome "+username);
        else
          out.println("Invalid username");
    }
}
```

3.6 : Sessions

Q.12 Explain the session tracking technique.

Ans. : • **HTTP** is a **stateless protocol** in which each request is independent of the previous request. And HTTP is a protocol using which user can

interact with the server via web browser and it cannot remember previously held communications but sometimes there is serious need to keep track of previous communication sessions. This can be achieved by **session tracking**.

- The **session tracking technique** is a mechanism by which we can keep track of previous sessions between server and the browser.
- For sending all state information to and fro between browser and server, usually an ID is used. This ID is basically a **session-ID.**
- Thus session-ID is passed between the browser and server while processing the information. This method of keeping track of all the information between server and browser using session-ID is called **session tracking.**
- In servlets, for creating the sessions **getSession()** method can be used. This method returns the object which stores the bindings with the names that use this object. And these bindings can be managed using **getAttribute(), setAttribute(), removeAttribute()** methods. Actually in session tracking two things are playing an important role -

1. One is **HttpServletRequest** interface which supports **getSession()** method.
2. The another class is **HttpSession** class which supports the binding managing functions such as **getAttribute(), setAttribute(), removeAttribute()** and **getAttributeNames().**

Q.13 Write a servlet program to keep track of number of times user is visiting the page. Display the count appropriately.

Ans. : Servlet Program(SessionServletDemo)

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SessionServletDemo extends HttpServlet
{
```

```
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
 res.setContentType("text/html");
 HttpSession session=req.getSession();
 String heading;
 Integer cnt=(Integer)session.getAttribute("cnt");
 if(cnt==null)
 {
  cnt=new Integer(0);
  heading="Welcome You are accessing the page for the First Time";
 }
 else
 {
 heading="Welcome once again!";
 cnt=new Integer(cnt.intValue()+1);
 }

session.setAttribute("cnt",cnt);
 PrintWriter out=res.getWriter();
 out.println("<html>");
 out.println("<head>");
 out.println("</head>");
 out.println("<body>");
 out.println("<center>");
 out.println("<h1>"+heading);
 out.println("<h2> The number of previous access= "+cnt);
 out.println("</center>");
 out.println("</body>");
 out.println("</html>");
 }
 }
```

3.7 : Cookies

Q.14 What is cookies ? Write and explain various methods used in cookies ?

Ans. : Definition : Cookies are some little information that can be left on your computer by the other computer when we access an internet.

- The **cookie** class is used to create cookies in servlet.
- **Syntax :** The Syntax for constructor for cookies are,
  - Cookie()
  - Cookie(String name, String value)
- Various methods used in Cookie are described in following table

Sr. No.	Method	Purpose
1.	public string getName()	It returns the name of the cookie.
2.	public String getValue()	It returns the value of the cookie.
3.	public string setName()	It sets or changes the name of the cookie.
4.	public String setValue()	It sets or changes the value of the cookie.
5.	public void addCookie(Cookie c)	The cookie is added in the response object of HttpServletResponse interface.
6.	public Cookie[] getCookies()	All the cookies can be returned using this method with the help of HttpServletRequest interface.

Program for Cookies : Below is simple HTML form in which a servlet is invoked. This servlet creates a cookie by the name **My_Cookie** and stores the value entered by you in the textbox of HTML form. You can further get the information stored in the cookie by another servlet program **getCookieServlet**.

HTML Program

```
<html>
<head>
```

```
<title>Demo of Cookie</title>
</head>
<body>
<form name="form1" method="post"
action="http ://localhost:8080/examples/servlet/mycookieservlet">
<h3> Enter the value for my Cookie: </h3>
<input type=text name="txt_data" size =30 value=" ">
<input type=submit  value="Submit">
</form>
</body>
</html>
```

Servlet Program [mycookieservlet.java]

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class mycookieservlet extends HttpServlet
{
public void doPost(HttpServletRequest req,HttpServletResponse res)
throws ServletException, IOException
{
String txt_data = req.getParameter("txt_data");
// Create cookie.
Cookie cookie = new Cookie("My_Cookie", txt_data);
// Adding cookie to HTTP response.
res.addCookie(cookie);
// Write friendly output to browser.
res.setContentType("text/html");
PrintWriter out = res.getWriter();
out.println("<h2>MyCookie has been set to : ");
out.println(txt_data);
out.println("<br><br><br>");
out.println("This page shows that the cookie has been added");
out.close();
}
}
```

We have first created an object of Cookie class using which the cookie can be added using **addCookie()** method.

Servlet Program[getcookieservlet.java]

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class getCookieServlet extends HttpServlet
{
 public void doGet(HttpServletRequest req,HttpServletResponse res)
 throws ServletException,IOException
 {
 Cookie[ ] my_cookies=req.getCookies();
 res.setContentType("text/html");
 PrintWriter out=res.getWriter();
 out.println("<b>");
 int n=my_cookies.length;
 for(int i=0;i<n;i++)
 {
 String name=my_cookies[i].getName();
 String value=my_cookies[i].getValue();
 out.println("name= "+name);
 out.println("and value= "+value);
 }
 out.close();
 }
 }
```

In the above program using **getName()** and **getValue()** functions we can get the name of the cookie as well as the value of the cookie respectively.

Q.15 Explain session and cookies in servlets ? Write any one program to demonstrate session or cookies. ☞ **[SPPU : Dec.-19, Marks 5]**

Ans. : Session : Refer Q.12 and Q.13.

Cookies : Refer Q.14.

3.8 : URL Rewriting

Q.16 What is session ? How cookies and URL rewriting for session management in servlet. ☞ **[SPPU : May-22, Marks 9]**

Ans. : URL rewriting : This is more precise technique in which information is embedded into URL. For instance.

```
GET    / Introduction.html    HTTP P.1

HOST   www.mywebpage.com
```

Browser —— Requests ——→ Server

The server then repsonds by putting **sid=xf1234ad** in response body.

```
HTTP/1.1            200 OK
<html>
<a  href = Introduction.html; sid=xf1234ad>
    Click here to go to Next page </a>
    : : :
</html>
```

Browser ←—— Response —— Server

The browser will demand for the required web page by putting the same URL string in the request.

```
GET    /Introduction.html  sid=xf1234ad   HTTP/1.1
```

Browser —— Request ——→ Server

When browser makes requests for another page same URL is embedded. There by server will come to know that the request is related to previous page and the work in same session will get co-related.

Session : Refer Q.12.

Cookies : Refer Q.14.

3.9 : Other Servlet Capabilities

Q.17 List and explain additional HttpServletRequest and HttpServletResponse methods.

Ans. : HttpServletRequest interface adds the methods that relates to the **HTTP** protocol.

Following table lists various useful methods of **HttpServletRequest**

Method	Description
String getRemoteAddr()	This method returns IP address of the client who is making the request.
String getRemoteHost()	This method returns the name of the client who is making the request.
String getProtocol()	This method returns the name of the protocol, client is using.
String getHeader(String field)	When the field name is given the value for the header is returned.
StringBuffer getRequestURL()	This method returns the String containing the URL which is used to access the servlet. This string does not contain the query string appended to URL.

- **HttpServletResponse** interface adds the methods that relate to the **HTTP** response. It extends the ServletResponse interface.
- Following table lists various useful methods of **HttpServletResponse**

Methods	Description
void addCookie(Cookie cookie)	Adds the specified cookie to the response.
void sendRedirect(String location)	Sends a temporary redirect response to the client using the specified redirect location URL and clears the buffer.

int getStatus()	Gets the current status code of this response.
String getHeader(String name)	Gets the value of the response header with the given name.
void setHeader(String name, String value)	Sets a response header with the given name and value.
void setStatus(int sc)	Sets the status code for this response.
void sendError(int sc, String msg)	Sends an error response to the client using the specified status and clears the buffer.

3.10 : Data Storage

Q.18 Explain the serialization technique for data storage with illustrative example.

Ans. : Serialization is the process of writing the state of the object to the byte stream. This technique is useful when we want to store the current state of the object to the File. For serialising the object we need to implement **serializable** interface. This interface does not define any member.

Java Program[SerializProg.java]

```
import java.io.*;
class SerializProg
{
public static void main(String[] args)throws
ClassNotFoundException,IOException
{
    Myclass myobject=new Myclass("Parth",10);//creating the object
    System.out.println("The object is: "+myobject);//displaying the
    contents of the object  ObjectOutputStream o=new
    ObjectOutputStream(new FileOutputStream("object.txt"));
    o.writeObject(myobject);//writing the object to a file
```

```
o.flush();
o.close();
ObjectInputStream i=new ObjectInputStream(new
FileInputStream("object.txt"));
Myclass newobject=(Myclass)(i.readObject());//reading the object
from the file
i.close();
System.out.println("\nYou have stored :"+newobject);//displaying the
object's value
}
static class Myclass implements Serializable
{
String name;
int roll;
public Myclass(String n,int r)
{
this.name=n;
this.roll=r;
}
public String toString()
{
return "name= "+name+" and roll= "+roll;
}
}
}
```

Implements the serializable interface

3.11 : Servlets Concurrency

Q.19 Write a short note on – servlets and concurrency.

Ans. : • Concurrent web servers handle multiple users as follows - It begins with one servlet, executes some instruction, then suspends it for a while and start executing the other servlet. Eventually completes the second servlet and returns back to first servlet to execute the remaining instructions. Refer Fig. Q.19.1.

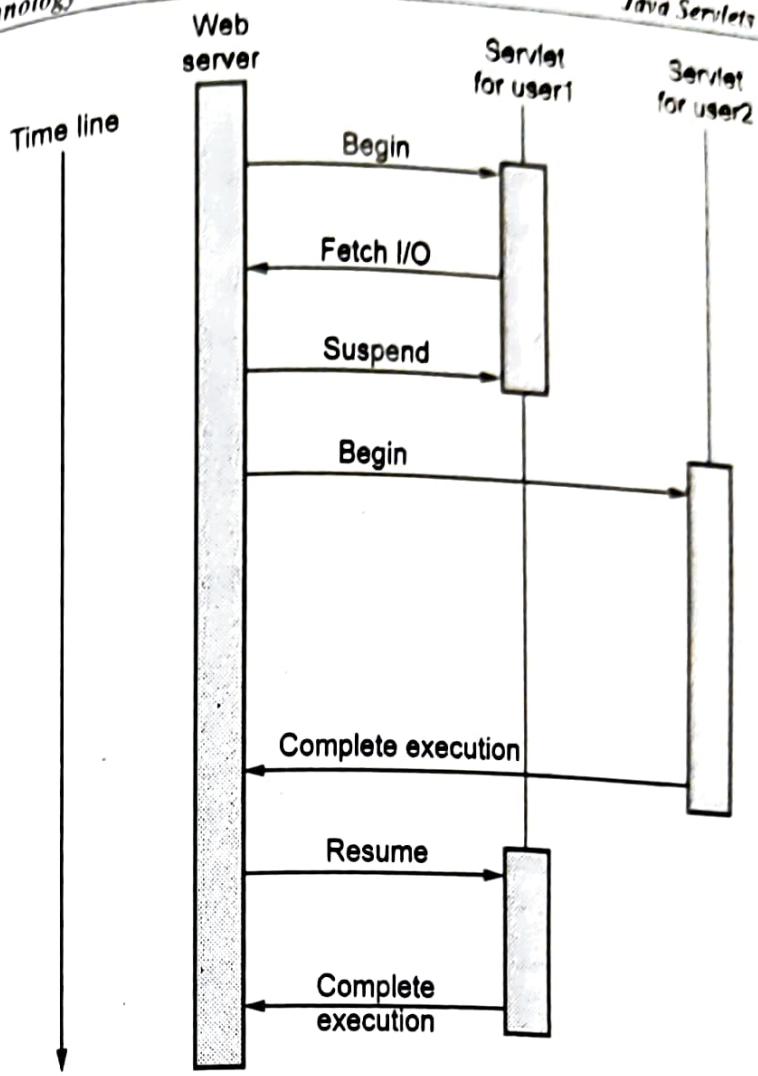


Fig. Q.19.1 Concurrency achieved by web server

- Thread is a light weight process. It is used for concurrent execution of the tasks.
- The Java Virtual Machine maintains a data structure containing the information about each thread. The server creates a thread for each HTTP request. The data structure maintains the information about the state of the thread along with the references to the **HttpServletRequest** and **HttpServletResponse** objects created by the server.
- The Java Virtual Machine also maintains the information whether the particular thread is currently running or not.

- When one thread is suspended then the state of other executing thread is loaded into Java Virtual Machine.

Example Program :

```
import java.io.IOException;
import javax.servlet.*;
import javax.servlet.http.*;
import java.math.*;
public class SimpleServlet extends HttpServlet
{

    public void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException
    {
        doPost(req, resp);
    }
    public void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException
    {
        int counter = 0;
        resp.getWriter().println(this);
        for (int i = 0; i < 10; i++)
        {
         resp.getWriter().println("Counter = " + counter);
         try
         {
             Thread.currentThread().sleep((long) Math.random() * 1000);
             counter++;
         }
         catch (InterruptedException exc) { }
        }
    }
}
```

3.12 : Databases (MySQL) and Java Servlets

Q.20 Explain the concept of JDBC.

Ans. : • JDBC stands for Java DataBase Connectivity.

- JDBC is nothing but an API (i.e. Application Programming Interface)
- It consists of various classes, interfaces, exceptions using which Java application can send SQL statements to a database. The SQL is a Structured Query Language used for accessing the database.
- JDBC is specially used for having connectivity with the RDBMS packages (such as Oracle or MYSQL) using corresponding JDBC driver.

Q.21 Explain how JDBC works.

Ans. : Following is a way by which JDBC works -

1) First of all Java application establishes connection with the data source.
2) Then Java application invokes classes and interfaces from JDBC driver for sending queries to the data source.
3) The JDBC driver connects to corresponding database and retrieves the result.
4) These results are based on SQL statements which are then returned to Java application.
5) Java application then uses the retrieved information for further processing.

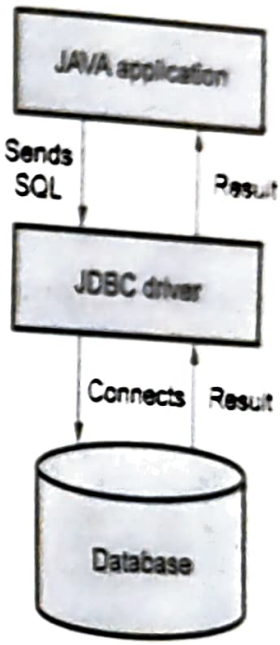


Fig. Q.21.1 Role of JDBC

Q.22 What are the uses of JDBC ?

Ans. :

1. JDBC helps the client to store and retrieve the data to the databases.
2. JDBC allows the client to update the databases.

Q.23 List and explain the steps to connect database with servlet.

Ans. : Following steps are used to connect JDBC to MYSQL

Step 1 : Import java.sql.* package in the JDBC program

Following line can be included in your JDBC program at the beginning.

```
import java.sql.*
```

Step 2 : Load JDBC Driver.

The JDBC driver for MYSQL can be loaded using following statement

```
Class.forName("com.mysql.jdbc.Driver");
```

Step 3 : Get connection using the Driver Manager

```
conn=riverManager.getConnection("jdbc:mysql://localhost:3306/my_database","root","password");
```

Step 4 : Create Statement

```
stmt = conn.createStatement();
```

Step 5 : Execute Query

```
String sql = "SELECT Roll,StudName FROM my_table";
ResultSet rs = stmt.executeQuery(sql);
```

Step 6 : Display the result

Q.24 Write a Servlet program to connect to a MySQL database. Create a student database named StudentDB.

Ans. :

1. Create a StudentDB in MySQL using following query

```
create database StudentDB;
```

2. The Servlet program can be written as follows

```
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletDemo extends HttpServlet {
  public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    try {
```

```
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/StudentDB","",
"");
        Statement stmt = con.createStatement();
        out.print("You are successfully Connected to Database...");

    }
    catch (SQLException e) {
        throw new ServletException("Could not display records.", e);

    }
    catch (ClassNotFoundException e) {
        throw new ServletException("JDBC Driver not found.", e);

    }
    out.close();
  }
}
```

3.13 : XML : XML Documents and Vocabularies

Q.25 Describe the following web technology : XML.

[SPPU : Dec.-19, Marks 2]

Ans. : • XML stands for eXtensible Markup Language.

- This scripting language is similar to HTML. That means, this scripting language contains various tags. But these tags are not predefined tags, in-fact user can define his own tags.
- Thus HTML is designed for **representation of data** on the web page whereas the XML is designed for **transport or to store data.**

Q.26 What are the strengths of XML technology ? Also list the limitations of using XML. **[SPPU : Dec.-18, Marks 5]**

Ans. : Strengths :

1. XML document is human readable and we can edit any XML document in simple text editors.
2. The XML document is language neutral. That means a Java program can generate an XML document and this document can be parsed by Perl.

3. Every XML document has a tree structure. Hence complex data can be arranged systematically and can be understood in simple manner.
4. XML files are independent of an operating system.

Limitations :

1. XML syntax is verbose and redundant as compared to other text based data transmission formats.
2. Redundancy in syntax of XML causes higher storage and transportation cost when there is large volume of data.
3. XML does not support array.
4. XML document is less readable.
5. Due to verbose nature of XML syntax the XML file size is usually very large.

Q.27 What are the strengths of XML technology ? Explain the need for XML. **[SPPU : May-19, Marks 5]**

Ans. : Strengths : Refer Q.26.

Need :

1. XML is used to display the meta contents i.e. XML describes the content of the document.
2. XML is useful in exchanging data between the applications.
3. The data can be extracted from database and can be used in more than one application. Different applications can perform different tasks on this data.

3.14 : XML Declaration

Q.28 What are the basic building blocks of XML ?

Ans. : • Various building blocks of XML are -

1. Elements

- *The basic entity is* **element.** *The elements are used for defining the tags. The elements typically consist of opening and closing tag. Mostly only one element is used to define a single tag.*

2. Attribute

- The attributes are generally used to specify the values of the element. These are specified within the double quotes.

For example -

```
<flag type="True">
```

- The type attribute of the element flag is having the value True.

3. CDATA

CDATA stands for Character Data. This character data will be parsed by the parser.

4. PCDATA

It stands for Parsed Character Data (i.e. text).

Q.29 What are the rules that must be followed while writing XML ?

Ans. : • Rules that must be followed while writing XML -

Here are some rules that must be followed while writing the XML programs -

1. XML is a **case sensitive**. For example -

```
<hobby>I like drawing</Hobby>or
<Hobby>I like drawing</hobby>
```

is not allowed because the words **hobby** and **Hobby** are treated differently.

2. In XML each **start tag** must have **matching end tag**. For example -

```
<staring> XML is funny to write
<simple> It is simple to implement </simple>
```

That means a closing tag is a must.

3. The elements in XML must be **properly nested**. For example-

```
<one><two>Hello how are you?</one></two>   is wrong but
<one><two>Hello how are you?</two></one>   is correct
```

3.15 : XML Namespaces

Q.30 Explain the concept of XML namespaces along with example.

Ans. : • Sometimes we need to create two different elements by the same name. The xml document allows us to create **different elements** which are having the **common name**. This technique is known as **namespace**.

- In some web documents it becomes necessary to have the same name for two different elements. Here different elements mean the elements which are intended for different purposes. In such a case support for namespace technique is very much helpful.
- **For example :** Consider the following xml document -

namespacedemo.xml

```
<File-Description>
    <text fname="input.txt">
        <describe>It is a text file</describe>
    </text>
    <text fname="flower.jpg">
        <describe>It is an image file</describe>
    </text>
</File-Description>
```

The above document does not produce any error although the element **text** is used for two different attribute values.

3.16 : DOM based XML Processing

Q.31 What is DOM ? How DOM based XML processing is done ?

Ans. : Document Object Model (DOM) is a set of platform independent and language neutral Application Programming Interface (API) which describes how to access and manipulate the information stored in XML or in HTML documents.

Thus XML DOM is for

- **Loading** the XML document.
- **Accessing** the elements of XML document.
- **Deleting** the elements of XML document.
- **Changing** the elements of XML document.

Q.32 List and explain various properties and methods used for DOM based XML processing.

Ans. : Various properties are,

Property	Meaning
nodeName	Finding the name of the node.
nodeValue	Obtaining the value of a node.
parentNode	Getting the parent element.
childNodes	Obtaining the child nodes.
attributes	For getting the attributes values.

Various methods are,

Method	Meaning
getElementsByTagName(name)	This method is used to access all the elements that are specified with the tag name.
appendChild(node)	This method is used for inserting a child node.
removeChild(node)	This method is used for removing the child nodes.

3.17 : Transforming XML Documents

Q.33 Write short note on - XML transformation.

[SPPU : May-22, Marks 5]

OR Discuss the XSLT technology with an example.

[SPPU : May-19, Marks 5]

Ans. : • The XSLT stands for **XSL** Transformations and XSL stands for eXtensible Stylesheet Language.

- XSL consists of three parts :
  1. XSLT is a language for **transforming XML documents.**
  2. XPath is a language for **navigating in XML documents.** In other words we can reach to any node of XML document using XPath.

3. XSL-FO is a language for formatting XML documents for **displaying it in desired manner.**

- Using XSLT we can decide the way by which we want the element to get displayed.
- XSLT uses **XPath** to find information in an XML document. XPath is used to navigate through elements and attributes in XML documents. We can sort these elements, hide or display particular element and can apply some decision making logic on these elements.
- XSLT processors **take two input documents** one is **XML document** and another is **XSLT document.** The XSLT document is nothing but a program and XML document is nothing but the input data, thus this program works on the XML input data. Then some or whole part of the XML document is selected, modified and merged with XSLT program document in order to produce another document.
- This newly produced document is provided as input to the XSLT processor which in turn produce another document called the **XSL document.**
- The XSL document is used along with the application so that particular application can be displayed on the web browser in some desired manner.

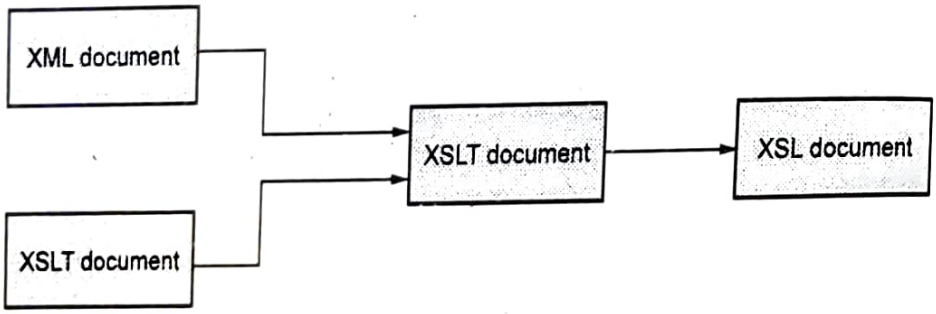


Fig. Q.33.1 Processing of XSLT document

- The XSLT document makes use of **templates** using which particular code can be described in XML document.
- XML document and then particular code in XSLT is executed when the match is found.

- XSLT processor processes the XML document sequentially by reading each line one by one.
- The XSLT model can be described as **template driven** or **data driven** model.

Q.34 Write at least any five differences between XML and XSLT.

☞ **[SPPU : March-20, Marks 5]**

Ans. :

Sr.No.	XML	XSLT
1.	XML stands for extensible markup language.	XSLT stands for extensible stylesheet language transformations.
2.	XML is used to store data in structured format.	XSLT is used for transforming and formatting the XML file.
3.	XML does not perform transformations.	XSLT can transform one XML format into another XML format or into HTML or into plain text.
4.	XML contains the user defined tags.	XSLT contains special XSL tags and programming constructs.
5.	XML uses the user defined raw data for storage.	XSLT uses XPath for transformation and formatting of XML file.

3.18 : DTD Schema, Elements, Attributes

Q.35 What is XML schema ?

Ans. : The XML schemas are used to represent the **structure of XML document.**

- The goal or purpose of XML schema is to define the building blocks of an XML document. These can be used as an alternative to XML DTD.
- The XML schema language is called as **XML Schema Definition (XSD) language.**

- XML schema defines elements, attributes, elements having child elements, order of child elements. It also defines fixed and default values of elements and attributes.
- XML schema also allows the developer to use **data types.**

Q.36 Give an example of simple schema.

Ans. : XML Schema [StudentSchema.xsd]

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
                    <xs:element name="Student">
        <xs:complexType>
         <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="std" type="xs:string"/>
            <xs:element name="marks" type="xs:string"/>
         </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Q.37 Explain the differences between external and internal DTD.

☞ **[SPPU : May-18, Marks 5]**

Ans. :

Sr. No.	Internal DTD	External DTD
1.	When a DTD is declared **in the same** file it is called internal DTD	When DTD is declared in a **separate file** it is called as external DTD
2.	The syntax for internal DTD is : <!DOCTYPE root-element [element-declarations]> , where root-element is the name of root element and element-declarations is where you declare the elements.	The syntax for external DTD is : <!DOCTYPE root-element SYSTEM "file-name.dtd"> where file-name is the file with **.dtd extension.**

3.	When the internal DTD is used, it is used inside the XML document.	An external DTD, on the other hand, is a separate file that specifies the style of an XML document.
4.	The Internal DTD has declarations	The external DTD has no declarations
5.	To reference it as internal DTD, standalone attribute in XML declaration must be set to yes.	To reference it as external DTD, standalone attribute in the XML declaration must be set as no. This means, declaration includes information from the external source.
6.	**Example :** In an XML document we declare following internal DTD <!DOCTYPE student [<!ELEMENT student (RollNo,Name,address)> <!ELEMENT RollNo (#PCDATA)> <!ELEMENT Name (#PCDATA)> <!ELEMENT address (#PCDATA)>]>	**Example :** In student.dtd file we declare following DTD <!ELEMENT student (RollNo,Name,address)> <!ELEMENT RollNo (#PCDATA)> <!ELEMENT Name (#PCDATA)> <!ELEMENT address (#PCDATA)>

Q.38 What are XML schemas ? How are they better than DTDs ?

☞ **[SPPU : May-18, Marks 5]**

Ans. : XML schema : Refer Q.36.

Advantages of schema over DTD

1. Both the schemas and DTDs are useful for defining structural components of XML. But the DTDs are basic and cannot be much specific for complex operations. On the other hand schemas are more specific.
2. The schemas provide support for defining the type of data. The DTDs do not have this ability. Hence content definition is possible using schema.

3. The schemas are namespace aware and DTDs are not.
4. The XML schema is written in XML itself and has a large number of built in and derived types.
5. The schema is the W3C recommendation. Hence it is supported by various XML validator and XML processors but there are some XML processors which do not support DTD.
6. Large number of web applications can be built using XML schema. On the other hand relatively simple and compact operations can be built using DTD.

Q.39 Explain DTD in XML with schemes, elements and attributes.

☞ **[SPPU : May-22, Marks 9]**

Ans. : • The document type definition is used to define the basic building block of any xml document.

- Using DTD we can specify the various elements types, attributes and their relationship with one another.
- Basically DTD is used to specify the set of rules for structuring data in any XML file.
- **For example :** If we want to put some information about some students in XML file then, generally we use tag **student** followed by his/her **name, address, standard and marks.** That means we are actually specifying the manner by which the information should be arranged in the XML file. And for this purpose the document type definition is used.
- Various building blocks of XML are -

1. Elements

The basic entity is **element.** The elements are used for defining the tags. The elements typically consist of opening and closing tag. Mostly only one element is used to define a single tag.

2. Attribute

The attributes are generally used to specify the values of the element. These are specified within the double quotes.

For example -

```
<flag type="True">
```

The type attribute of the element flag is having the value *True*.

3. **CDATA**

CDATA stands for character data. This character data will be parsed by the parser.

4. **PCDATA**

It stands for Parsed Character **Data** (i.e. text). Any parsable character data should not contain the markup characters. The markup characters are < or > or &. If we want to use less than, greater than or ampersand characters then make use of < , > or &

Q.40 Consider a hospital management system. Write a DTD program to consolidate and show the bill to be paid by the inpatients (Assume your own data).

Ans. : Patients.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Patients [
<!ELEMENT Patients (Patient)*>
<!ELEMENT Patient (Name, Address,Test,Date,Doctor,Bill)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT Test (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Doctor (#PCDATA)>
<!ELEMENT Bill (#PCDATA)>
]>
<Patients>
    <Patient>
        <Name>Anuradha</Name>
        <Address>Pune</Address>
        <Test>ECG</Test>
        <Date>1-1-2010</Date>
        <Doctor>Mr.XYZ</Doctor>
        <Bill>Rs.500</Bill>
    </Patient>
    <Patient>
```

```
        <Name>Archana</Name>
        <Address>Mumbai</Address>
        <Test>Blood Test</Test>
        <Date>10-3-2010</Date>
        <Doctor>Mr.PQR</Doctor>
        <Bill>Rs.100</Bill>
    </Patient>
</Patients>
```

Merits of DTD

1. DTDs are used to define the structural components of XML document
2. These are relatively simple and compact.
3. DTDs can be defined inline and hence can be embedded directly in the XML document.

Demerits of DTD

1. The DTDs are very basic and hence cannot be much specific for complex documents.
2. The language that DTD uses is not an XML document. Hence various frameworks used by XML cannot be supported by the DTDs.
3. The DTD cannot define the type of data contained within the XML document. Hence using DTD we cannot specify whether the element is numeric, or string or of date type.
4. There are some XML processor which do not understand DTDs.
5. The DTDs are not aware of namespace concept.

3.19 : AJAX : Introduction

Q.41 What is the purpose of AJAX ? Explain with relevant example.

☞ **[SPPU : Dec.-19, Marks 8]**

OR Write short note on : AJAX. ☞ **[SPPU : May-22, Marks 5]**

Ans. : Purpose of AJAX :

- AJAX is a **Asynchronous Java Script** and XML

- Here
  - **Asynchronous** means, the execution of script does not disturb the user's work.
  - **JavaScript** because, it makes use of Javascripting to do the actual work.
  - **XML** because along with JavaScript the XML is also supported to perform the given task in AJAX.
- It is not a new programming language but it is a kind of web document which adopts certain standards.
- AJAX allows the developer to exchange the data with the server and updates the part of web document without reloading the web page.

Example : Refer Q.42.

Q.42 Write different applications without AJAX and with AJAX.

[SPPU : May-18, Marks 4]

Ans. : Application Without AJAX :

```
<!DOCTYPE html>
<html>
<body>
<button type="button" onclick="myfunction('stationary.xml')">Display
Stationary Items</button>
<br><br>
<table id="demo" border="1"></table>
<script>
function myFunction(xml)
{
    var i;
    var xmlDoc = xml.responseXML;
    var table="<tr><th>Item</th><th>Price(Rs.)</th></tr>";
    var x = xmlDoc.getElementsByTagName("STATIONARY");
    for (i = 0; i <x.length; i++)
    {
        table += "<tr><td>" +
        x[i].getElementsByTagName("ITEM")[0].childNodes[0].nodeValue +
         "</td><td>" +
       x[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue +
```

```
        "</td></tr>";
    }
    document.getElementById("demo").innerHTML = table;
}
</script>
</body>
</html>
```

Application using AJAX : Refer Q.43.

Q.43 Create an XMLHttpRequest to retrieve data from an XML file and display the data in an HTML table. The data to be retrieved is a collection of stationary items stored in an XML file.

Ans. : Step 1 : Create an xml file storing the stationary items. The code is as follows -

```
/ELEMENTS>
<STATIONARY>
<ITEM>Notebook</ITEM>
<PRICE>40</PRICE>
</STATIONARY>
<STATIONARY>
<ITEM>Punching Machine</ITEM>
<PRICE>60</PRICE>
</STATIONARY>
<STATIONARY>
<ITEM>Pencil Box</ITEM>
<PRICE>40</PRICE>
</STATIONARY>
<STATIONARY>
<ITEM>Eraser</ITEM>
<PRICE>5</PRICE>
</STATIONARY>
</ELEMENTS>
```

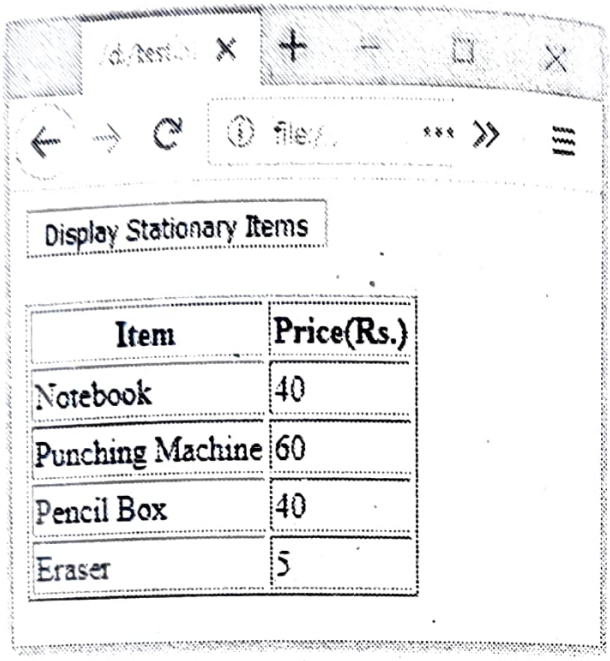
Step 2 : The XML file displaying the stationary items in a tabular form is as given below -

```
test.html
<!DOCTYPE html>
<html>
<body>
```

```
<button type="button" onclick="loadXMLDoc()">Display Stationary
Items</button>
<br><br>
<table id="demo" border="1"></table>
<script>
function loadXMLDoc()
{
 if(window.XMLHttpRequest)
 {
   xmlhttp = new XMLHttpRequest();

 }
    else
{
   xmlhttp = new ActiveXObject("Microsoft.req"):://to execute on Inter
                 Explorer

 }
xmlhttp.onreadystatechange = function()
 {
    if (this.readyState == 4 && this.status == 200) {
      myFunction(this);

}
};
       xmlhttp.open("GET","stationary.xml",true);
   xmlhttp.send();

}
function myFunction(xml)
{
 var i;
    var xmlDoc = xml.responseXML;
    var table="<tr><th>Item</th><th>Price(Rs.)</th></tr>";
    var x = xmlDoc.getElementsByTagName("STATIONARY");
    for (i = 0; i <x.length; i++)
    {
   table += "<tr><td>" +
   x[i].getElementsByTagName("ITEM")[0].childNodes[0].nodeValue +
   "</td><td>" +
   x[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue +
   "</td></tr>";
 }
 document.getElementById("demo").innerHTML = table;
}
```

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

Output



Q.44 What are the merits and demerits of AJAX ?

Ans. : Merits of AJAX

1. Response time is faster, hence increases performance and speed.
2. XMLHttpRequest object call as a asynchronous HTTP request to the Server for transferring data both side. It's used for making requests to the non-Ajax pages.
3. It is used in form validation.
4. It performs fetching of data from database and storing of data into database without reloading page.

Demerits of AJAX

1. It has browsing compatibility issues.
2. It is impossible to bookmark AJAX updated page contents.
3. Search engine would not crawl AJAX generated content. Hence, Search Engines like Google cannot index AJAX pages.

Q.45 What is use of XMLHttpRequest object ? Explain its use with the help of simple JavaScript document. ☞ **[SPPU : May-18, Marks 6]**

Ans. : • XMLHttpRequest object is an important element in AJAX.

- XMLHttpRequest is an API which is used by JavaScript, VBScript and some other scripting languages.
- The methods of XMLHttpRequest are used in transferring data between a Web browser and a web server.
- It is because of XMLHttpRequest object, to update parts of web page without reloading the whole page.

Syntax

```
Variable = new XMLHttpRequest();
```

Example : Refer Q.47.

Q.46 Explain with simple example the XMLHttp request object. List and explain different values of readystate and status property of the HTTP request object. ☞ **[SPPU : May-19, Marks 5, Dec.-18, Marks 6]**

Ans. : Example - Refer Q.47.

Various Properties of XMLHttpRequest object

Property	Description
readyState	Specifies the ready state of XMLHttpRequest 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
onreadystatechange	When readystate property changes the onreadystatechange event listener will be automatically invoked
responseText	Returns the response data as a string
status	Returns the status number. Following is the meaning of various numbers returned by the status property

	200: "OK" 403: "Forbidden" 404: "Not Found"
statusText	Returns the status text as "OK", or "Forbidden"

Q.47 Write AJAX script to obtain the student information stored in XML document. The information should be displayed on clicking the button. It should be displayed in tabular form.

Ans. :

Step 1 : Create an XML file for storing the student information. The XML file is as follows

Student.xml

```
<Student>
    <student_data>
        <Name>AAA</Name>
        <Marks>45</Marks>
    </student_data>
    <student_data>
        <Name>BBB</Name>
        <Marks>55</Marks>
    </student_data>
    <student_data>
        <Name>CCC</Name>
        <Marks>67</Marks>
    </student_data>
    <student_data>
        <Name>DDD</Name>
        <Marks>84</Marks>
    </student_data>
</Student>
```

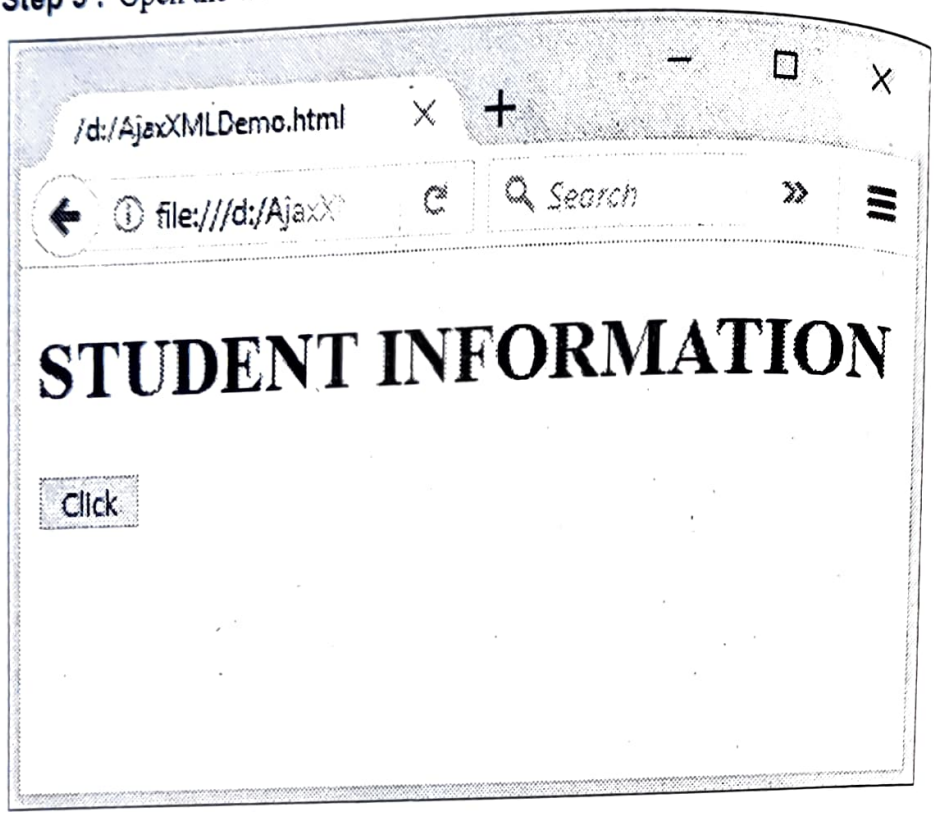
Step 2 : Create a AJAX script as follows -

AjaxXMLDemo.html

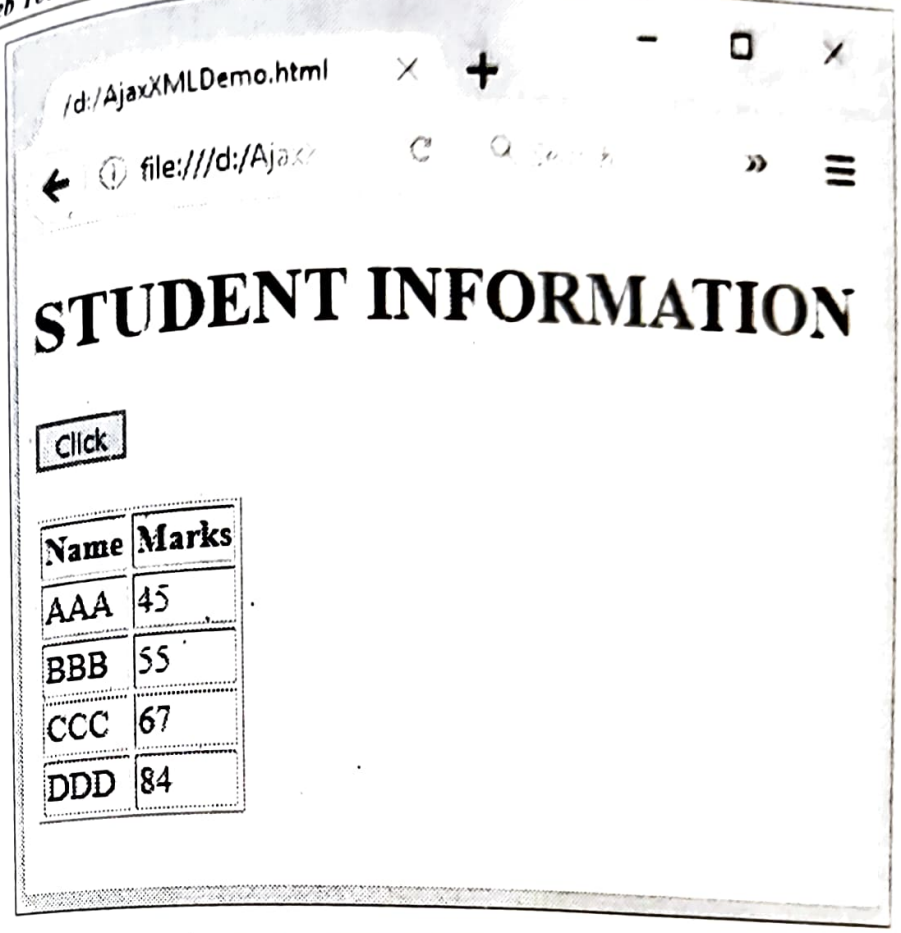
```
<!DOCTYPE html>
<html>
<body>
<h1>STUDENT INFORMATION</h1>
<button type="button" onclick="MyFun()">Click</button>
<br><br>
<table border="1" id="demo"></table>
<script>
function MyFun()
{
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function()
    {
        if (this.readyState == 4 && this.status == 200)
        {
         Load_XML_File(this);
        }
    };
    xhttp.open("GET", "Student.xml", true);
    xhttp.send();
}
function Load_XML_File(xml)
{
    var i;
    var xmlDoc = xml.responseXML;
    var table="<tr><th>Name</th><th>Marks</th></tr>";
    var x = xmlDoc.getElementsByTagName("student_data");
    for (i = 0; i <x.length; i++)
    {
        table += "<tr><td>" +
        x[i].getElementsByTagName("Name")[0].childNodes[0].nodeValue +
        "</td><td>" +
```

```
        x[i].getElementsByTagName("Marks")[0].childNodes[0].nodeValue +
        "</td></tr>";
    }
    document.getElementById("demo").innerHTML = table;
}
</script>
</body>
</html>
```

Step 3 : Open the web browser and the output will be as follows -



On clicking the button we will get



3.20 : Working of AJAX

Q.48 Draw and explain how AJAX works with the help of suitable example. ☞ **[SPPU : May-18, Marks 6]**

Ans. : • When user makes a request, the browser creates a object for the **HttpRequest** and a request is made to the server over an internet.

- The server processes this request and sends the required data to the browser.
- At the browser side the returned data is processed using JavaScript and the web document gets updated accordingly by sending the appropriate response.

Following Fig. Q.48.1 illustrates this working.

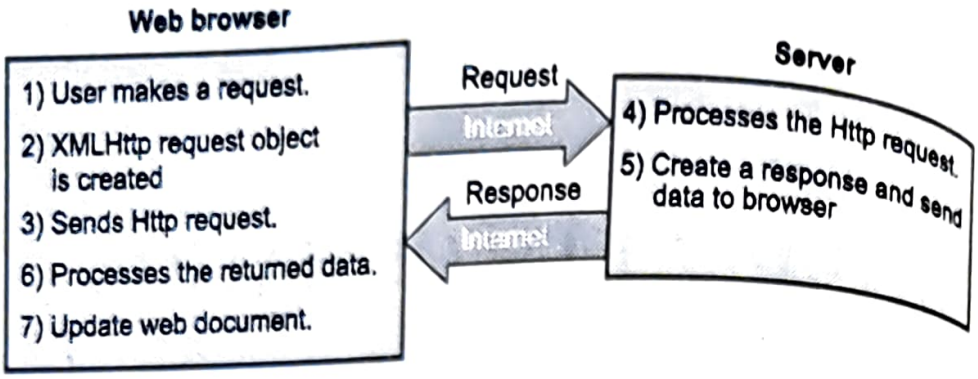


Fig. Q.48.1 Working of AJAX

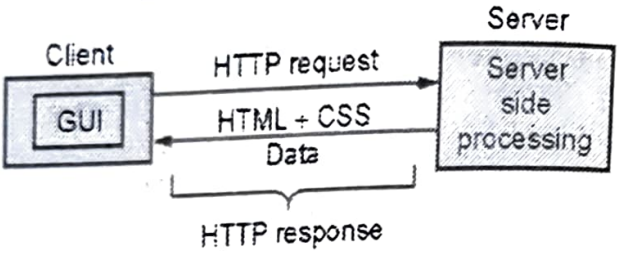


Fig. Q.48.2 Traditional web application architecture

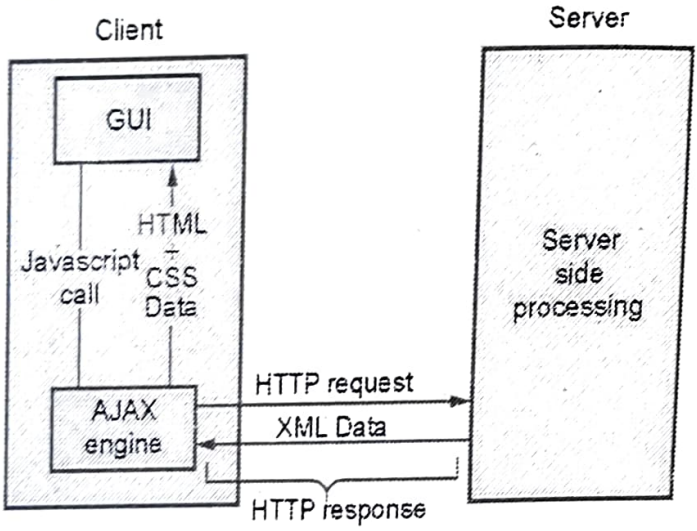


Fig. Q.48.3 AJAX web application architecture

END...

4 JSP and Web Services

4.1 : Introduction to Java Server Pages

Q.1 What is JSP ?

Ans. : • JSP stands for Java Server Pages.

- It is an alternative way than the servlet to build the dynamic web pages. It is built on top of the servlet.

4.2 : JSP and Servlets

Q.2 Explain life cycle of JSP. Write advantages of JSP over servlet.

☞ **[SPPU : April-18, Marks 5, May-22, Marks 9]**

Ans. : JSP life cycle :

- JSP pages can be processed using **JSP container** only. Following are the steps that need to be followed while processing the request for JSP page -

1. Client makes a request for required JSP page to the server. The server must have JSP container so that JSP request can be processed. For instance : Let the client makes a request for *xyz.jsp* page.
2. On receiving this request the JSP container searches and then reads the desired JSP page. Then this JSP page is straightaway converted to corresponding **servlet**.

   Every **JSP element** is converted into corresponding **Java code**. This phase is called **translation phase**. The output of translation phase is a servlet. For instance : our *xyz.jsp* gets converted to *xyzServlet.java*.
3. This servlet is then compiled to generate the servlet class file. Using this class the response can be generated. This phase is called **request processing phase**.
4. The JSP container thus **executes** the servlet class file.
5. A requested page is then returned to the client as a **response**. Refer Fig. Q.2.1.

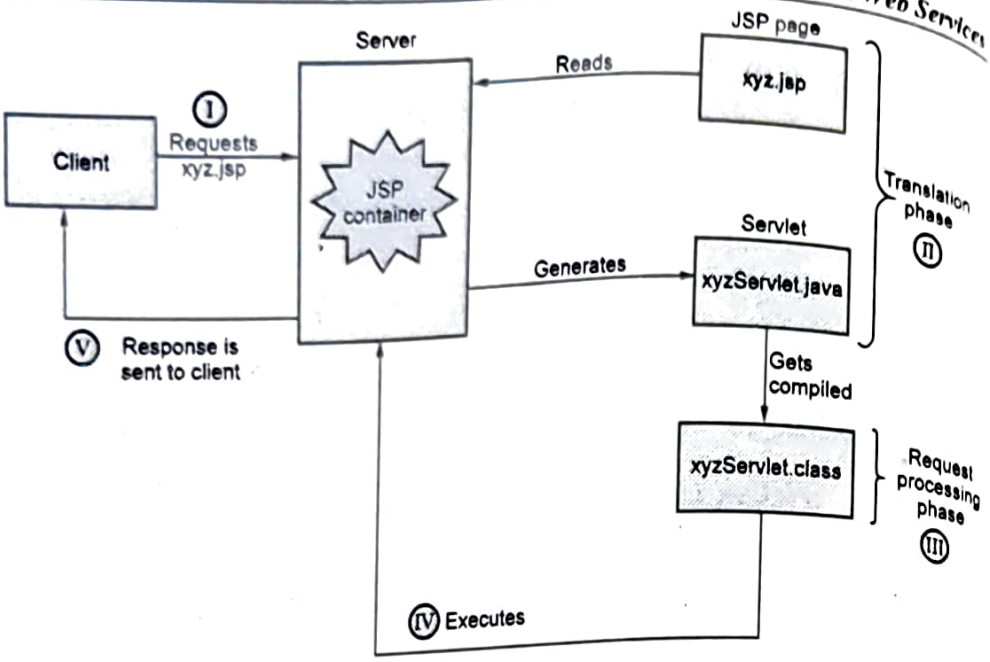


Fig. Q.2.1 JSP processing

Advantages of JSP over servlet :

- Following are some advantages of Java Server pages -
  1. JSP is useful for **server side programming.**
  2. JSP can be used **along with servlets.** Hence **business logic** for any application can be developed using JSP.
  3. **Dynamic contents** can be handled using JSP because JSP allows scripting and element based programming.
  4. JSP allows creating and using our own **custom tag libraries.** Hence any application specific requirements can be satisfied using custom tag libraries. This helps the developer to develop any kind of application.
  5. JSP is a **specification and not a product.** Hence developers can develop **variety of applications** and add up to performance and quality of software products. Due to this many companies are ready to invest in JSP technology.
  6. JSP is an essential component of J2EE. Hence using JSP it is possible to develop simple as well as complex applications.

4.3 : Running JSP Applications

Q.3 Write a simple JSP page for displaying the message "This is my first JSP page!!!" ☞ **[SPPU : May-18, Marks 5]**

Solution : We follow these steps to get the JSP page displayed -

First of all open some text editor like Notepad and type the following code.

hello.jsp

```
<%@ page language="java" contentType="text/html" %>
<%@ page import ="java.util.*" %>
<html>
<!-- This is basic JSP page -->
<title>JSP Demo </title>
<body>
<%-- Displaying the message on the browser --%>
      <% out.println("This my first JSP page!!!"); %>
   </body>
</html>
```

4.4 : Basic JSP

Q.4 What are the usage of JSP directives and JSP actions ? ☞ **[SPPU : April-18, Marks 5]**

Ans. : JSP Directives : JSP directives control the processing of entire JSP Page. It gives direction to the server regarding the processing of a page.

Syntax

```
<%@ directive name [attribute name="value" attribute name="value"
.....]%>
```

1. Page directive :

The page directive is used to provide the information about the page.

For example -

```
<%@page import="java.io.*" %>
<%@page language="java" %>
<%@page contentType="text/html" %>
```

2. Include directive :

Include directive is used to copy the content of one JSP page to another. It like including the code of one file into another.

For example

```
<% @include file=calculate.jsp %>
```

3. Taglib directive :

This directive basically allows user to use custom tags in JSP. The custom tags are those which are created by user.

For example -

```
<%@ taglib uri="mytaglib" prefix="sampletag" %>
```

JSP Actions :

- The standard actions are those actions that can be defined by the JSP specification itself.
- Following is a list of standard actions.

Action Element	Description
<jsp:useBean>	This tag is used to instantiate the object of java bean.
<jsp:setProperty>	This tag is used to set the property value for the java bean.
<jsp:getProperty>	This tag is used to get the property value from the java bean.
<jsp:include>	This tag works as a subroutine. It includes the response from servlet or JSP when request is being processed.
<jsp:param>	For adding the specific parameter to the request this tag is used. Normally this tag is used inside <jsp:include> or <jsp:forward>
<jsp:forward>	This tag helps in forwarding the current request to servlet or to JSP page.

<jsp:plugin>	This tag is used to generate HTML code and to embed the applet into it.
<jsp:attribute>	This tag sets the value of the action attribute
<jsp:element>	This tag generated the XML elements dynamically.
<jsp:text>	This tag is used to handle template text. When JSP pages are written as XML documents then this tag is used.
<jsp:body>	This tag is used to set the body element.

Q.5 Write the differences between include action and include directive in JSP. **[SPPU : Dec.-18, May-19, End Sem, Marks 5]**

Ans. : The difference is as follows -

1) Include directive includes the file at translation time - that is when the JSP gets converted into the equivalent servlet whereas the include action includes the file at runtime.
2) If the included file is changed but not the JSP which is including it then the changes will reflect only when we use include action tag. The changes will not reflect if you are using include directive as the JSP is not changed so it will not be translated for request processing and hence the changes will not reflect.
3) Syntax difference : Include directive : <%@ include file="file_name" %> whereas include action has like this <jsp:include page="file_name" />
4) When using include action tag we can also pass the parameters to the included page by using param action tag but in case of include directive it's not possible.

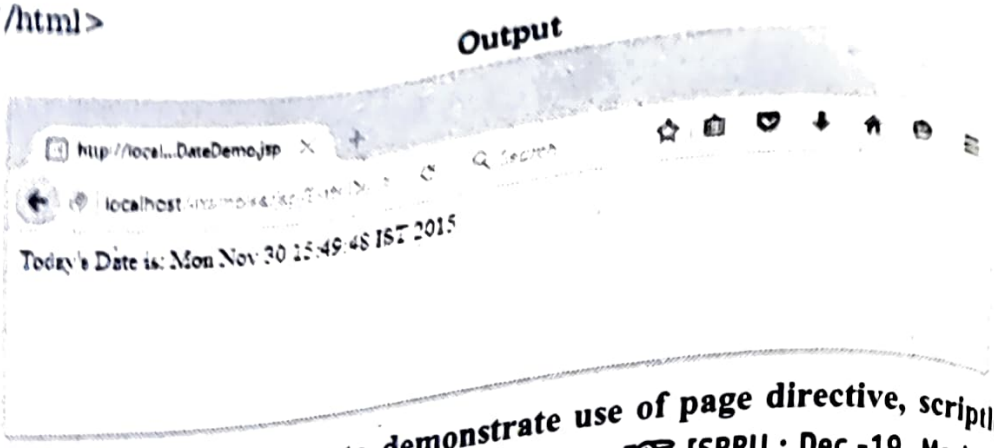
Q.6 Write a JSP program to display current date and time.

Ans. : DateDemo.jsp

```
<%@ page language="java" contentType="text/html" %>
<%@ page import ="java.util.*" %>
<html>
```

```
<body>Today's Date is:
        <%= new Date().toString() %>
</body>
</html>
```

Output



Q.7 Write JSP program to demonstrate use of page directive, scriptlet and expression tag. **[SPPU : Dec.-19, Marks 5]**

Ans. : Page Directive : Refer Q.4 (1).

Scriptlet :

- The code that appears between the <% and %> delimiters is called a *scriptlet*. Scriptlets are nothing but java code **enclosed within** <% and %> tags.
- The everything other than a JSP statement in the JSP is called template text. For example.

Templatetext.jsp

```
<%@ page language="java" contentType="text/html" %>
<html>
        <head>
         <title>Demo for Tempplate Text</title>
        </head>
        <body bgcolor="gray">
         <h1>Twinkle Twinkle little stars</h1>
         <h2>How I wonder what you are!!!</h2>
         <li>like a diamond in the sky</li>
         <p>
            <% out.print("JSP is eqaul to HTML and JAVA "); %>
        </p>
    </body>
</html>
```

Scriptlet

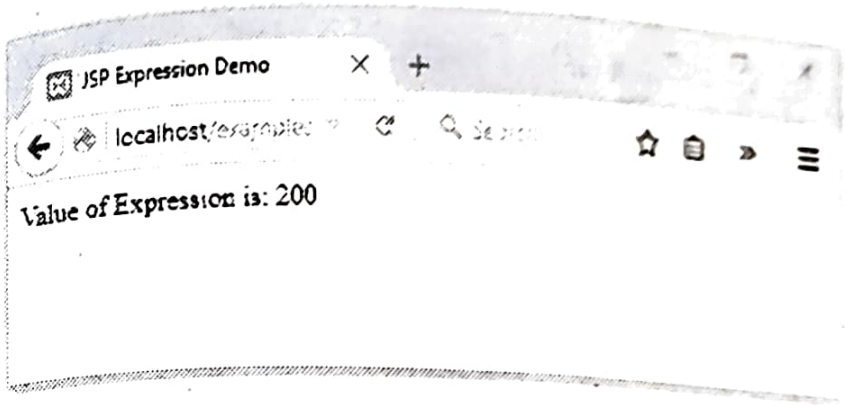
Expression : The expression tag is used to represent the expression in JSP page. The syntax of writing expression is -

```
<%= Java Expression %>
```

For example -

```
<html>
  <head>
    <title>JSP Expression Demo</title>
  </head>

  <body>
    Value of Expression is :
    <%= (10*20) %>
  </body>
</html>
```

Output



4.5 : JavaBeans Classes and JSP

Q.8 What is JavaBean class ? Also explain the characteristics and properties of JavaBean class

Ans. : • Java beans are reusable components. We can use simple Java bean in the JSP.

- Beans are used in the JSP pages as the instance of class. We must specify the scope of the bean in the JSP page. Here scope of the bean means the range time span of the bean for its existence in JSP.
- Bean is basically Java class with **getXX** and **setXX** methods.

Characteristics for JavaBean Class

1) Java Bean class contains a default **no argument constructor**. Th constructor must have the access specifier as **public**.
2) It should be serializable and should **implement serializable interface**
3) It can have any number of properties which can be read or written. For these properties we write **getter and setter methods.**

JavaBean Properties

- A JavaBean Property is a named attribute that can be accessed using the object.
- The JavaBean properties can be accessed through two methods namely
  1) **getPropertyName()** : This method is called **accessor** method. It returns the property value. This method is written in some special way. For example - If the property is **rollNo** then the method name for **rollNo** is **getRollNo()**
  2) **setPropertyName()** : This is called **mutator** method. This method is generally used to assign the value to the properties. This method is written in some special way. For example - If the property is **rollNo** then method name will be **setRollNo()**

Q.9 What is the JSP action tags ? Discuss in details jsp:useBean action tag. ☞ **[SPPU : May-18, Marks 5]**

Ans. : JSP Action Tags : Refer Q.4.

JSP::useBean : The **jsp:useBean** action tag is used to locate or instantiate a bean class. If bean object of the Bean class is already created, it doesn't create the bean depending on the scope. But if object of bean is not created, it instantiates the bean.

Syntax

```
<jsp:useBean id = "beanName" class = "className"
        scope = "page | request | session | application">
```

where

id attribute specifies the name of the bean.

Scope attribute specify where the bean is stored.

The **class** attribute specify the fully qualified **classname**.

Example :

```
<html>
   <head>
      <title> JSP BEAN DEMO </title>
   </head>
   <body>
      <jsp:useBean id="empID" class="BeanProg.Employee" >
      <jsp:setProperty name = "empID" property = "empName" value =
      "AAA"/>
   <jsp:setProperty name = "empID" property = "empNo" value = 10/>
      </jsp:useBean>
      <p>Employee Name:
      <jsp:getProperty name = "empID" property = "empName"/>
      </p>
      <p>Employee Number:
      <jsp:getProperty name = "empID" property = "empNo"/>
      </p>
   </body>
</html>
```

4.6 : Support for the Model-View-Controller Paradigm

Q.10 What is MVC ? Draw and explain MVC architecture for developing web applications. ☞ **[SPPU : Dec.-18, Marks 6]**

OR Explain JSP support for MVC i.e. model, view controller for developing web application. ☞ **[SPPU : May-22, Marks 8]**

Ans. : • The design model of JSP application is called MVC model. The MVC stands for Model-View-Controller. The basic idea in MVC design model is to separate out design logic into three parts - modelling, viewing and controlling.

- Any server application is classified in three parts such as business logic, presentation and request processing.

- The **business logic** means the coding logic applied for manipulation of application data. The **presentation** refers to the code written for look and feel of the web page. For example: background color, font style, font size, placing of controls such as text boxes, command buttons and so on. The request processing is nothing but a combination of business logic and presentation. The request processing is always done in order to generate the response.

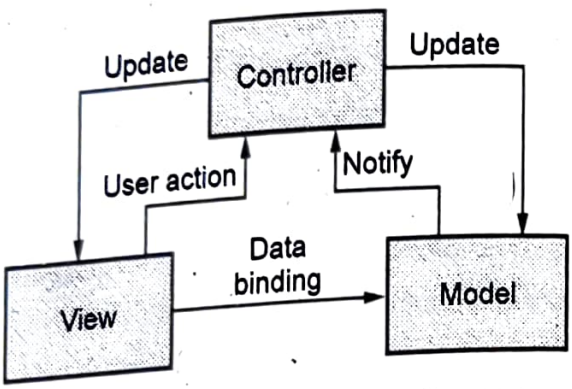


Fig. Q.10.1 Model View Controller (MVC)

- According to the MVC design model, the model corresponds to business logic, view corresponds to presentation and controller corresponds to request processing.

Advantages of using MVC design model

- The use of MVC architecture allows the developer to keep the separation between business logic, presentation and request processing.
- Due to this separation it becomes easy to make **changes** in presentation without disturbing the business logic. The changes in presentation are often required for accommodating the new presentation interfaces.

4.7 : Web Service Concepts

Q.11 Write short note on – Web service concepts.

Ans. : Definition : The Web Services are the software systems that are displayed by the web browser using the web protocol. These software

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

- Web service is a software system designed which is **independent** of specific hardware or software on which it is running.
- For example -

Credit card validation system - For using this service the client simply enters the ID and password for the credit card and web service reports the validity of it.

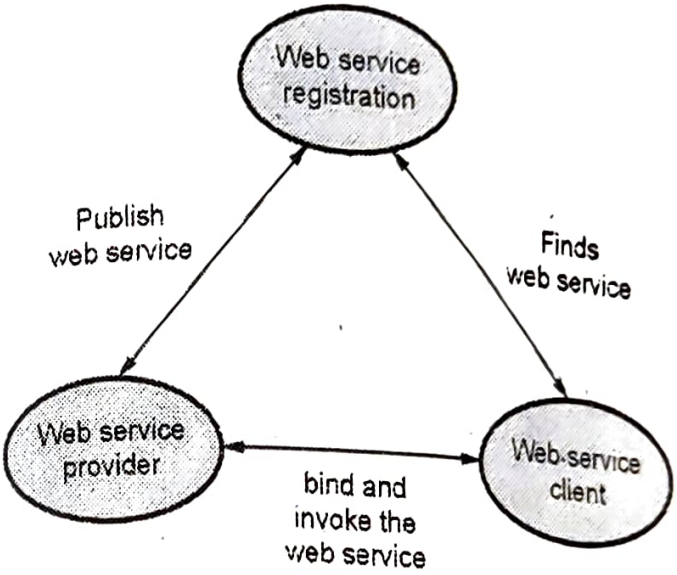


Fig. Q.11.1 Basic web service model

Following are the steps that a web service model follows -

Step 1 : In the first step, a service provider **publishes** a web service in the web **service registry**.

Step 2 : A web client who demands for some web service **searches** in the registry. After finding a match for the desired web service in the registry. After finding a match for the desired web service in the registry, the client chooses it.

Step 3 : The client **binds** to the corresponding web service provider and invokes for the service.

Q.12 What are web services ? List and discuss components of web services.

[SPPU : Dec.-18, Marks 6]

Ans. : Web services : Refer Q.11.

Components of Web Services : The role of various components such as XML, SOAP, WSDL, UDDI in web service building is as shown below -

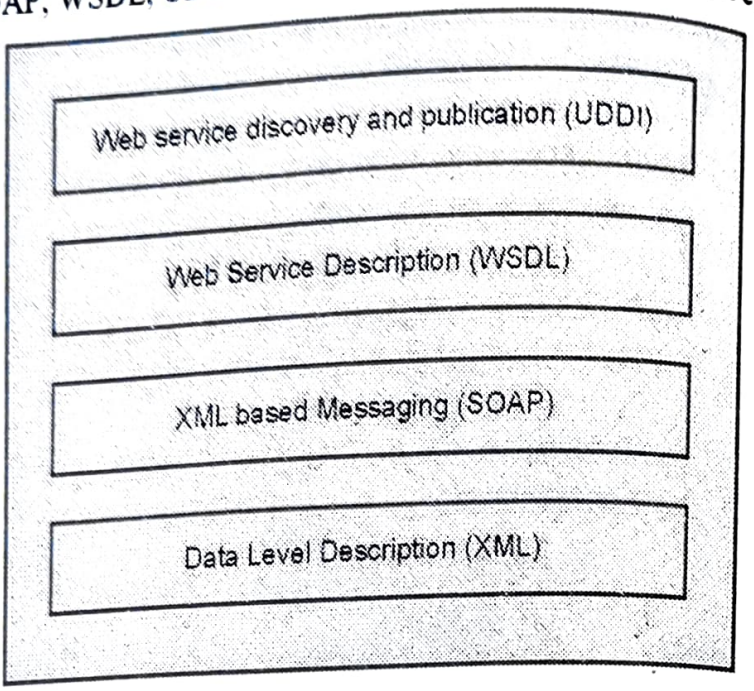


Fig. Q.12.1 Web service elements

- At the lowest level of web services there are simply XML files that are for data level description.
- SOAP - **S**imple **O**bject **A**ccess **P**rotocol is a simple XML based protocol which allows exchange of messaging over HTTP.
- Typically web services make use of this protocol for exchange of information .
  - The SOAP makes use of JAX-RPC component. These are **J**AVA **A**PI for **X**ML that makes use of **R**emote **P**rocedure **C**alls.
- At the higher level very important element of web service lies i.e. WSDL. The **W**eb **S**ervice **D**escription **L**anguage is used for abstract description of web services. This description is provided in XML document. When the client wants to make use of web service, the users of the client make read this description file and understand what input data is needed to perform

the web service. Such a special file is called WSDL i.e. Web Services Description Language.

- UDDI i.e. **Universal Description and Discovery Integration**, provides a directory of web services, so that client can easily discover the services of their choices. The UDDI maintains a registry in which the web services are registered.

Q.13 Identify and justify benefits of using web services.

☞ **[SPPU : May-18, Marks 6]**

Ans. :

1) **Web services are XML based :** As information representation and record transportation is carried out using XML, there is no need for networking, operating system or platform binding. Hence web services are highly interoperable.

2) **Loosely coupled architecture :** The user interface for web service provider can change over time without disturbing the user's interaction with the web service provider. This feature of web service architecture makes the software system more manageable.

3) **Ability of being synchronous and asynchronous :** A web service is said to be synchronous when web services are invoked through web protocol and a client waiting for a response. The synchronous web services are handled using RPC(Remote Procedure Call) messages. A web service is said to be asynchronous when web services are invoked through web protocol and a client does not wait for a response. Servlets are used to implement the asynchronous web services.

4) **RPC based :** The web service support RPC i.e. Remote Procedure Call by offering its own service just similar to traditional looking functionality.

Q.14 What are the web services ? List and explain layers in protocol stack of web service architecture. ☞ **[SPPU : May-19, Marks 8]**

Ans. : Web services : Refer Q.11.

Layers in protocol stack of web service architecture : Refer Q.12.

4.8 : Writing a Java Web Service and Web Service Client

Q.15 Create a Web service in Java environment to return the sum of two integers with necessary deployment procedures.

Ans. : Web Service Server Program

```
package com.mypackage;

public class MyServer {
    public int add(int a,int b){
        return a+b;
    }
}
```

Web Service Client Program

```
package com.mypackage;
import com.mypackage.MyServerStub.Add;
import com.mypackage.MyServerStub.AddResponse;

public class MyClient {
    /**
     * @param args
     */
    public static void main(String[] args) throws Exception{
        // TODO Auto-generated method stub
        MyServerStub stub=new MyServerStub();
        Add p=new Add();
        p.setA(11);
        p.setB(22);
        AddResponse res=stub.add(p);
        int c=res.get_return();
        System.out.println("Addition: "+c);
    }
}
```

4.9 : WSDL

Q.16 Write a short notes on : WSDL. ☞ [SPPU : May-22, Marks 4]

Ans. : • The WSDL is a Web Service Descriptor Language which is based on XML.

- The purpose of this file is to describe the web service and the method of accessing it.
- WSDL is a W3C recommendation.
- It contain various definitions. Following elements are described in WSDL file.

Elements	Description
types	It specifies the data types of the symbols used by the web service.
messages	It specifies the messages used by the web service.
portType	It specifies the name of the operation (function) defined by the web service
binding	It specifies the name of the protocol of the web service, typically it is SOAP.

4.11 : SOAP

Q.17 Write a short notes on : SOAP. ☞ **[SPPU : May-22, Marks 4]**

Ans. : • **Simple Object Access Protocol (SOAP)** is a protocol based on XML. It is used by the web services for exchange of information. It is a W3C recommendation.

- This exchange of information is done over HTTP. This is a platform and language independent protocol.
- The structure of the SOAP is defined by four building blocks.

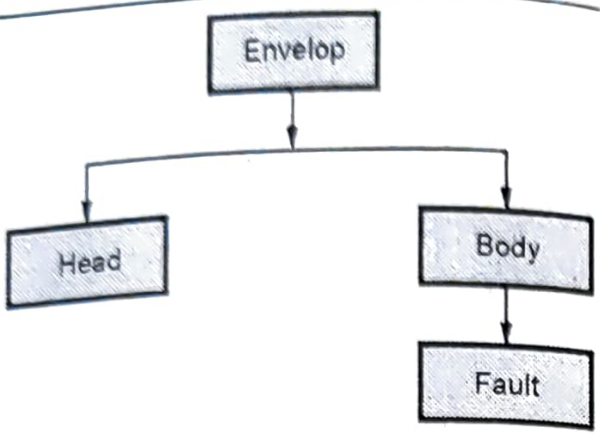


Fig. Q.17.1 SOAP building blocks

1) Envelop

The Envelop is the root of any SOAP document. It identifies that the XML document as a SOAP message. It defines two things - **Namespace** and **encodingStyle**. The Namespace value can be given as -

xmlns:soap="http://www.w3.org/2001/12/soap-envelope".

The encoding style is useful to define the data types used in the document. For example the value of encoding can be given as -

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"

2) Header

The SOAP header field is optional. It contains the header information. The header contains three attributes such as **mustUnderstand, actor** and **encodingStyle**

The **mustUnderstand** attribute specifies 0 or 1. If the value is 1 then it means that the receiver should process the header.

Normally the SOAP message travels over a message path having the endpoints at the two ends. The SOAP **actor** is used to specify the URI for this endpoint.

The **encodingStyle** is useful to define the data types used in the document

The SOAP header defines how the recipient should process the SOAP message.

3) Body

The required SOAP Body element contains the actual SOAP message intended for the ultimate endpoint of the message.

4) Fault

This is optional element of the SOAP message. It is used to represent the error code. It consists of elements such as **faultcode** which is a code for fault. The **faultstring** message gives the details about the fault.

4.12 : Struts

Q.18 What is struts ?

Ans. : • Struts is a framework software used for developing the Java Web Applications.

- The struts framework was initially created by Craig McClanahan and donated to Apache Software Foundation.
- It is an open source product.
- The latest commonly used release is **struts 2**.

Q.19 Write various features of struts framework.

☞ **[SPPU : Dec.-18, Marks 4]**

Ans. :

1. It is based on Model view Controller (MVC) architecture.
2. The action class is POJO (Plain Old Java Object) that means it is a simple Java class.
3. Struts2 provides the support to AJAX technology. That means only required field can be sent to the server and no need to send entire page. Due to this feature the execution is fast.
4. The struts2 framework can be integrated with other applications like Hibernate, Spring and so on.
5. Struts2 provides tag support such as UI tags, Data tags, control tags etc.
6. It provides support for views and templates.
7. Struts2 support multiple views for the same application.

Q.20 Draw and explain neat diagram which depicts MVC to the struts architecture. [SPPU : May-18,19, Marks 6]

Ans. : • The struts framework is based on Model, View and Controller architecture (MVC).

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

II. A Controller is nothing but a Java class in which **business logic** is written. The job of Controller is to take user input and pass it to the **Model**. The controller in turn invokes the **Model.**

III. Model stores the user's data may be in databases or so, and returns some result to the Controller.

IV. Using this result, the Controller figures out user requirements and accordingly forwards the result to the View page.

V. Finally required view can be obtained as a response (i.e. a display of required web page is obtained) on the web browser. This can be diagrammatically shown as below -

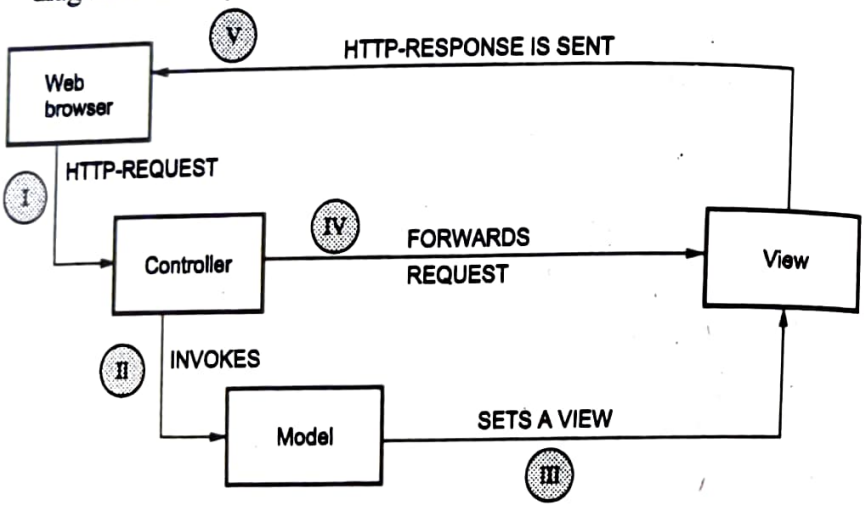


Fig. Q.20.1

In struts,

- **Model** components provide a model for "business logic" and data behind struts program.
- **View** components in struts are those components that present the information to the user or accept the user input. Typically these

components are JSP or HTML pages. View components are responsible for providing the display of required information. Struts provide large number of JSP custom tags or **struts tags** to simplify the development of view components.

- **Controllers** are the Java classes which allow developers to remove error handling activities from JSP pages. These classes are typically referred as **action classes**.

Q.21 Explain struts framework with its components. Also explain interceptors. **[SPPU : May-22, Marks 9]**

Ans. : Struts framework components : Refer Q.20.

Interceptors :

- Interceptor is an object which intercepts an action dynamically. It executed before and after the action execution.
- In Struts 2, interceptor is used to perform operations such as validation, exception handling, internationalization, displaying intermediate result etc.
- Following is a list of some Inceptors that are preconfigured and can be ready to use.
  1) **alias :** It converts similar parameters that have different names between requests.
  2) **chain :** If it is used with chain result type, it makes the properties of previous action available in the current action.
  3) **checkbox :** It is used to handle the check boxes in the form. By this, we can detect the unchecked checkboxes.
  4) **cookie :** It adds a cookie to the current action.
  5) **conversionError :** It adds conversion errors to the action's field errors.
- We can create our own Inceptor object. This custom based inceptor can be plugged in struts.
- Framework creates an object of **ActionInvocation** that encapsulates the action and all the interceptors configured for that action.

- Each interceptors are called before the action gets called.
- Once the action is called and result is generated, each interceptors are again called in reverse order to perform post processing work.
- Interceptors can alter the workflow of action. It may prevent the execution of action.

Q.22 How to use interceptors in Struts 2 ? List and describe important interceptors provided by Struts 2 framework.

[SPPU : May-18, Marks 6]

Ans. : Refer Q.21.

Q.23 Give advantages of using inceptors in Struts2.

Ans. :

1. If we want pre-processing services and post-processing services for an **Action** then these services can be provided by interceptor.
2. It provides pluggable functionalities for validation, exception handling and logging. We do not need to redeploy the application.
3. Interceptors can be coupled with request and response cycle.

Q.24 What are the different configuration files are require to develop any struts application ? Explain each configuration file.

[SPPU : Dec.-18, 19, Marks 6]

Ans. : There are two main configuration files used in struts and those are **struts.xml** and **web.xml**.

There is **struts.properties** file is used to override the default values of default.xml file provided by struts framework. So it is not mandatory.

struts.xml

The struts.xml file is used to initialize the resources such as inceptors, Actions classes. This file is present inside the **WEB-INF/classes** folder.

The structure of this file is

```
<?xml version = "1.0" Encoding = "UTF-8"?>
<!DOCTYPE struts PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
  "http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
  <constant name = "struts.devMode" value = "true" />
```

```
<package name = "FirstStrutDemo" extends = 'struts-default' >
  <action name = "demo"
  class = "myPkg.TestAction"
  method = "execute">
  <result name = "success">/MyHello.jsp</result>
  </action>

  <-- more actions can be listed here -->

</package>
<-- more packages can be listed here -->

</struts>
```

1) Package Element

The package element specifies module or your project name.

It has following attributes -

i) **name** : The name must define the package

ii) **namespace** : This is optional attribute. If namespace is not present, is assumed as the default namespace.

iii) **extends** : The package element mostly extends the struts-default package where interceptors and result types are defined.

2) Action Element :

- This element represents action. Various attributes of this element are -

i) **name** : This is any name with which the action is associated.

ii) **class** : The name of the action class is specified. This name must be accompanied with the package name inside which the action class is defined.

iii) **method** : This is optional field. By default the **execute** method is associated with this attribute.

iv) **result name** : The result name can be success, error and so on. In struts, on result element the page to be displayed on success or error is specified.

web.xml

The web.xml is a configuration file which determines how elements of the HTTP request are processed by the servlet container. It is not strictly a

Struts2 configuration file, but it is a file that needs to be configured for Struts2 to work.

The web.xml file needs to be created under the folder **WebContent/WEB-INF**.

This file contains the <welcome-file-list> tag using which the first file to be loaded on execution of struts project is specified. For example

```
<welcome-file-list>
   <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

Q.25 Identify and explain data tags from struts2 which are used to manipulate data displayed on any web application page.

[SPPU : Dec.-18, May-19, Marks 6]

Ans. : In Struts2 the data tags are used to manipulate the data displayed on the page. Following is a list of data tags used in struts2

(1) **action tag :** This tag allows helps the developer to call some action directly from JSP page. While using this tag, the action name is specified. The body contents of the tag are used to show the results of action. For example –

```
<div>Tag to say Big Hello</div>
<br />
<s:action name="hello" executeResult="true" />
```

(2) **include tag :** The include tag is a generic tag that is used to include result of servlet or a JSP page to the current page. For example –

```
<s:include value="sumoftwo.jsp">
 <s:param name="param1" value="value2" />
 <s:param name="param2" value="value2" />
</s:include>
```

(3) **param tag :** This tag is used to parameterize other tags. This tag has two parameters -

name(string): name of the parameter

value(object): value of the parameter

For example -

```
<ui:component>
```

```
<ui:param name="studname">Supriya</ui:param><br>
<ui:param name="studname">Kumar</ui:param>
</ui:component>
```

(4) **bean tag :** The bean tag instantiates a class that represents the JavaBean specification. This tag has a body which can contain a number of Param elements to set any mutator methods on that class.

For example -

```
<s:bean name="org.apache.struts2.util.Counter' var='counter'>
  <s:param name="first" value="10"/>
  <s:param name="last" value="20" />
</s:bean>
```

(5) **property tag :** The property tag is used to get the property of a value.

For example -

```
<s:property value="sampleBeanProperty"/>
```

(6) **date tag :** The date tag will allow you to format a Date easily. with the help of this tag we can specify a custom format as dd/MM/yyyy hh:mm. For example -

```
<s:date name = "my.birthday" format = "dd/MM/yyyy" />
```

Q.26 Explain the use of result type in struts

Ans. : • The result type is used to change the behaviour of view page to render the contents of action class.

- After executing the Action class, the target view page will be executed based on configuration in struts.xml.
- The Struts2 framework provides set of Result types like; chaining, dispatcher, redirect action, velocity and freemarket etc., that has to be configure in struts.xml for different kind of result.
- The <result> tag is used in **struts.xml** to render the view page. This result tag is provided with predefined result types such as dispatcher, redirect and so on. For example

```
<result name = "success" type="redirect">
```

- The **dispatcher** result type is the default type, and is used if no other result type is specified. It's used to forward to a servlet, JSP, HTML page, and so on, on the server. It can be written as

```
<result name="success" type="dispatcher">
<param name="location">/Success.jsp</param>
</result>
```

Q.27 Explain with an example how validation is performed in struts2.

Ans. : • In struts, validation framework help to perform validation before the action method is executed.

- You can validate data in action class also through **validate()** method.
- In the **validate()** method we can validate the user name, password,age,email, phone number and so on. If the validation fails an error is added using the **addFieldError()** method.

For example -

we will write the Java File for validating the user name. Two validations are used - i) the user name must not be blank and ii) the user name must be **root**. For validation purpose we will write one function named **validate()** in this Java program. This method is written in action class, hence this program will be created within package name **myPkg** under the **src** directory.

LoginProg.java

```
package myPkg;
import com.opensymphony.xwork2.ActionSupport;
public class LoginProg extends ActionSupport {
    private String userName;

    public String execute() {
      return SUCCESS;
    }
    public String getUserName() {
      return userName;
    }

    public void setUserName(String userName) {
      this.userName = userName;
```

```
    }
    public void validate() {
      if (getUserName().length() == 0) {
        addFieldError("userName", "UserName.required");
      } else if (!getUserName().equals("root")) {
        addFieldError("userName", "Invalid User");

      }
    }
  }
```

Q.28 Explain the concept of Localization. Why resource bundle is used in struts2 ?

Ans. : Localization :

- Internationalization is a process of translation or localization.
- Internationalization is abbreviated i18n because the word starts with the letter "i" and ends with "n", and there are 18 characters between the first i and the last n.
- Internationalization (i18n) is the process of planning and implementing products and services so that they can easily be adapted to specific local languages and cultures, a process called localization.
- The i18n interceptor provides multi-lingual support for your application.
- It handles setting locale for the action. It can be used if user wants to set his/her locale and get data according to the locale provided.
- A locale is an identifier and represents the local details of the user to the application. The locale can be created by **java.util.Locale** class. When a user session begins, the Locale object is sent in the HTTP request as parameter.

Concept of Resource Bundle

Struts2 makes use of resource bundle to provide multiple languages and locale options to the users. The resource bundle is a resource file in which key-value pairs are stored.

The name to these resource files is given in following naming format

Bundlename_language_country.properties

The **language_country** represents the country locale. For example English United States is represented as en_US, For Spanish it will be _es and so on. These properties file can be named as

global.properties : This is a default file for the English language.

global_fr.properites : This file specifies the French locale.

global_cn.properties : It specifies the code in Chinese.

global_hi.properties : It specifies the code in Hindi

Q.29 What is the use of Java annotation ? Explain its use in struts.

Ans. : • Java annotation is a tag used to represent the metadata. The metadata is a data about classes, methods or fields.

- Annotations in Java are used to provide some addition information. For instance - If user is entering his age in the text field, then that age must not be negative or it should not be blank - Such type of information can be provided to the user using annotations.
- A Java annotation in its shortest form looks like this :

  @Entity

- The @ character signals to the compiler that this is an annotation. The name following the @ character is the name of the annotation.
- Struts provides two forms of configuration.
  - The traditional way is to use the **struts.xml** file for all the configurations.
  - The other way of configuring Struts is by using the Java 5 Annotations feature.
- Using annotation we do not need **struts.xml** for configuration. Thus we can achieve zero configuration using annotation.
- Commonly, we can use 3 annotations :
  1) **@Action** annotation is used to mark the action class.
  2) **@Results** annotation is used to define multiple results for one action.
  3) **@Result** annotation is used to display single result.

END...

Unit V

5 Server Side Scripting Languages

5.1 : PHP : Introduction to PHP and its Uses

Q.1 What is PHP ?

Ans. : • PHP stands for PHP : **H**ypertext **P**reprocessor.

- PHP is a server-side scripting language. It is mainly used for form handling and database access.
- It is free to download and use.
- It is an alternative to CGI, ASP, ASP.NET and JSP.
- The extensions to the PHP files are .php, .php3 or .phtml.

5.2 : General Syntactic Characteristics

Q.2 Give any two general syntactic characteristics of PHP.

Ans. :

1) PHP code can be embedded in the XHTML document. The code must be enclosed within **<?php** and >
2) If the PHP script is stored in some another file and if it needs to be referred then **include** construct is used. For instance :
   Include("myfile.inc")
3) The variable names in PHP begin with the $ sign.

5.3 : Primitives, Operations and Expressions

Q.3 Classify data types of PHP and describe various data types in each type.

[SPPU : May-18,19, Marks 4]

Ans. : Integer Type

- For displaying the integer value the Integer type is used.
- It is similar to the **long** data type in C.
- The size is 32 bit.

Double Type

- For displaying the real values the double data type is used.
- It includes the numbers with decimal point, exponentiation or both. The exponent can be represented by E or e followed by integer literal.
- It is not compulsory to have digits before and after the decimal point. For instance .123 or 123. is allowed in PHP.

String Type

- The string literals can be defined using either single or double quotes.

For example -

```
'The total marks are= $marks'
```

will be typed as it is but

```
"The total marks are= $marks"
```

will display the value of $marks variable.

Boolean Type

- There are only two types of values that can be defined by the Boolean type and those are TRUE and FALSE.
- If Boolean values are used in context of integer type variable then TRUE will be interpreted as 1 and FALSE will be interpreted as 0.

5.4 : Output

Q.4 Explain the use of output statement in PHP.

Ans. : • The **print** function is used to create simple unformatted output. For example : The string can be displayed as follows

```
print "I am proud of my <b>country</b>"
```

The numeric value can also be displayed using the **print**. For example -

```
print(100);
```

It will display the output as 100.

- PHP also makes use of the **printf** function used in C. For example

```
printf("The student %d has %f marks",$roll_no,$marks);
```

5.5 : Control Statements

Q.5 Explain the use of selection statement in PHP along with example.

Ans. : The **if** statement, the **if ... else** statement or **if...elseif** statements are used in selection statements. Following is an example of it.

PHPDocument[selection.php]

```
<html>
<head>
   <title>Selection Demo</title>
</head>
<body>
<?php
   print "<h2>Selection Statement </h2>";
   $a=10;
   $b=20;
   $c=30;
   if($a>$b)
   if($a>$c)
   print "<b><I>a is the largest number </I></b>";
else
      print "<b><I> c is the largest number</I></b>";
   else
   if($b>$c)
      print "<b><I>b is the largest number</I></b>";
else
   print "<b><I>c is the largest number</I></b>";
?>
</body>
</html>
```

Q.6 Explain any one looping statement in PHP with example.

Ans. : • The **while**, **for** and **do-while** statements of PHP are similar to JavaScript.

• Following is a simple PHP script which displays the first 10 number.

PHPDocument[LoopDemo1.php]

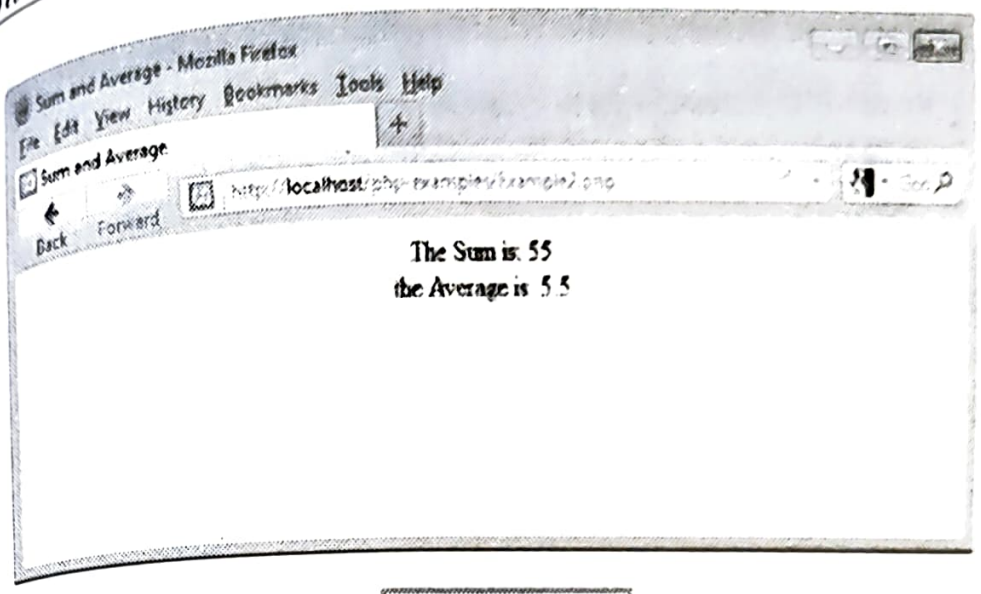
```
<?php
$i=1;
print "The numbers are ...";
print "<br/>";
while($i<=10)
{
    print $i;
    print "<br/>";
    $i++;
}
?>
```

Q.7 Write a PHP script to compute the sum and average of N numbers.

Ans. :

```
<html>
<head>
<title> Sum and Average </title>
</head>
<body>
<center>
<?php
$sum=0;
for($i=1;$i<=10;$i++)
{
    $sum+=$i;
}
$avg=$sum/10;
print "The Sum is:  $sum";
print "<br/>";
print "the Average is: $avg";
?>
</center>
</body>
</html>
```

Output



5.6 : Arrays

Q.8 How to create an array in PHP ?

Ans. : For creating an array in PHP, we use **array()** construct.

Syntax :

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

Example : Following PHP script shows how to create an array in PHP.

ArrayCreateDemo.php

```
<!DOCTYPE html>
<html>
<body>

<?php
$fruits = array("Mango", "Apple", "Banana","Grapes");
echo $fruits[0] . "," . $fruits[1] . "," . $fruits[2] . ",". $fruits[3];
?>
</body>
</html>
```

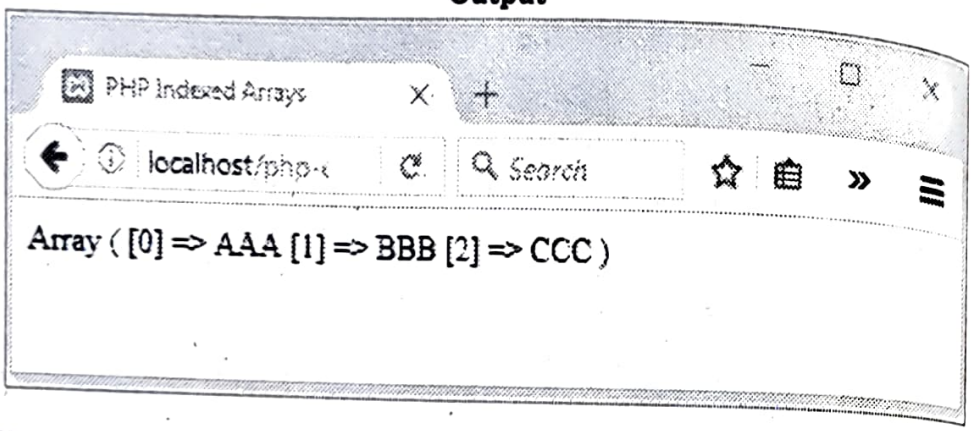
Q.9 Explain the types of arrays in PHP.

Ans. : There are three types of arrays.

1. Indexed array : Indexed array are the arrays with numeric index. The array values can be stored from index 0. For example -

```
<html>
<head>
    <title>PHP Indexed Arrays</title>
    </head>
   <body>
<?php
$names = array("AAA", "BBB", "CCC");
// Printing array structure
print_r($names);
?>
</body>
</html>
```

Output



Here values get stored at corresponding index as follows -

```
$mylist[0]   =   10;
$mylist[1]   =   20;
$mylist[2]   =   30;
$mylist[3]   =   40;
$mylist[4]   =   50;
```

We can directly assign some value at specific index.

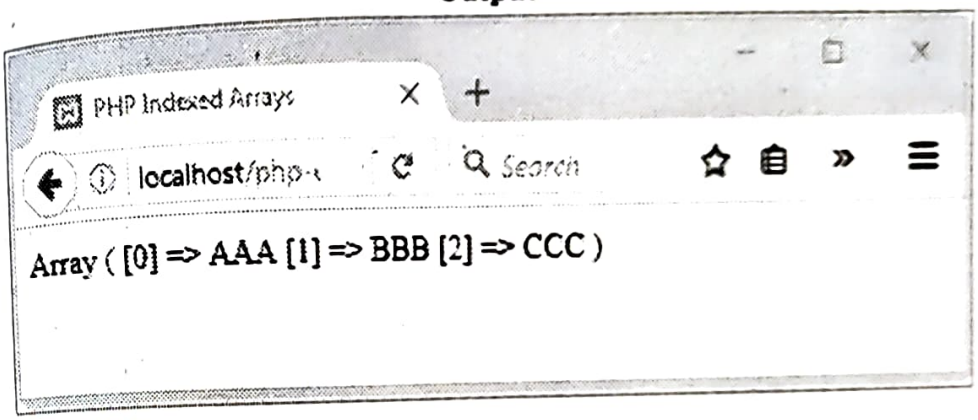
```
$mylist[5]   =   100;
```

2. Associated array : Associated arrays are the arrays with named keys. It is a kind of array with **name** and **value** pair. For example -

```
<html>
<head>
  <title>PHP Associative Array</title>
```

```
</head>
<body>
<?php
$city["AAA"] = "Pune";
$city["BBB"] = "Mumbai";
$city["CCC"] = "Chennai";

// Printing array structure
print_r($city);
?>
</body>
</html>
```

Output



3. Multidimensional arrays

- PHP support for multidimensional arrays.
- We can store the elements in two dimensional array as

```
$Student = array
(
array(10,"AAA"),
array(20,"BBB"),
array(30,"CCC"),
);
```

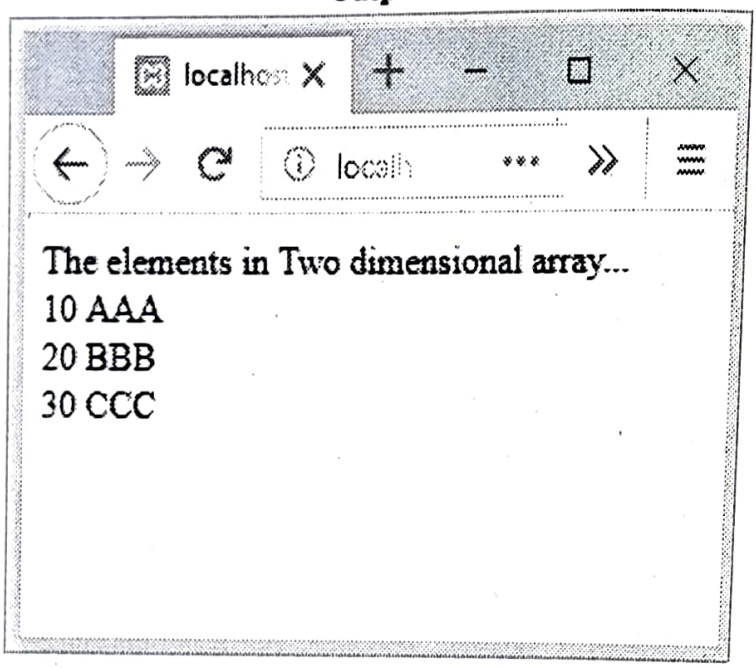
- The complete PHP program in which the multidimensional array is created and accessed is as follows -

PHP Document

```
<!DOCTYPE html>
<html>
```

```
<body>
<?php
$Student = array
(
 array(10,"AAA"),
 array(20,"BBB"),
 array(30,"CCC"),
 );
echo "The elements in Two dimensional array...<br/>";
for ($row = 0; $row < 3; $row++) {
 for ($col = 0; $col < 2; $col++) {
  echo " ".$Student[$row][$col];
 }
 echo "<br/>";
}
?>
</body>
</html>
```

Output



Q.10 What is associative arrays in PHP ? Explain it with the help of simple PHP code. ☞ **[SPPU : May-18,19, Marks 6]**

Ans. : Refer Q.9(2).

Q.11 What is multi-dimensional arrays in PHP ? Explain it with simple PHP code. ☞ **[SPPU : Dec.-18, Marks 6]**

Ans. : Refer Q.9(3).

Q.12 Explain implode and explode functions in PHP.

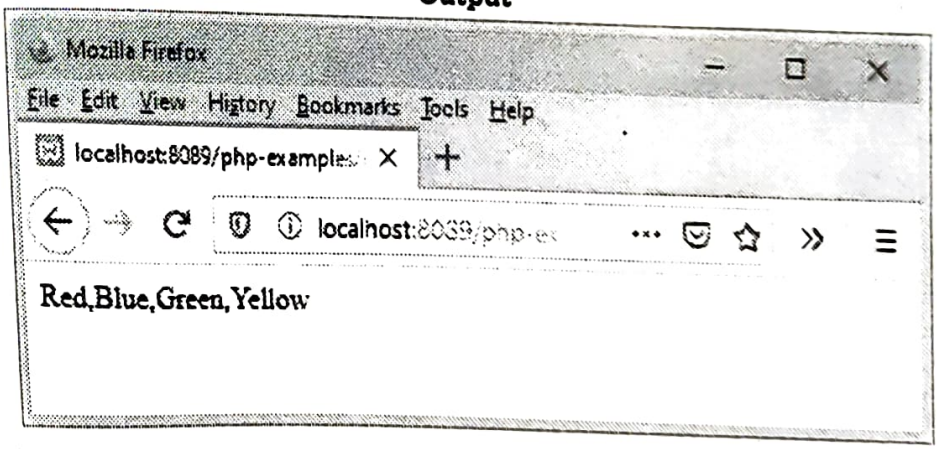
Ans. : The implode function :

The implode function converts array into the string. For example -

implodeDemo.php

```
<!DOCTYPE html>
<html>
<body>
<?php
$arr[0] = "Red";
$arr[1] = "Blue";
$arr[2] = "Green";
$arr[3] = "Yellow";
$text = implode(",",$arr);
echo $text;
?>
</body>
</html>
```

Output



The explode function

The explode() function is to split a string.

Syntax :

```
explode(delimiter, string_name, limit)
```

Where

Delimiter : It sets the boundary string within the input string

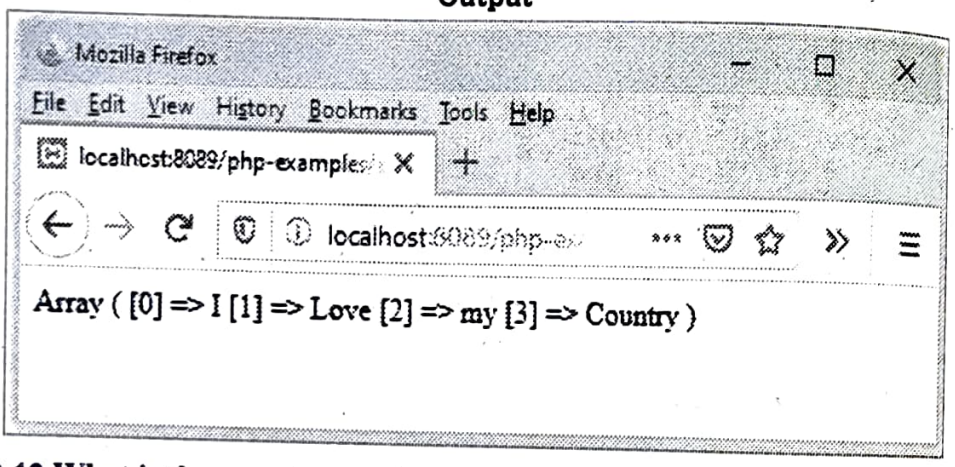
String_name : The name of the string to be split

Limit : It indicates the maximum number of elements in the output array if set to +ve value. If set to -ve value, all but the last element will be present in the output array.

Example Program

```
<!DOCTYPE html>
<html>
<body>

<?php
$str="I Love my Country";
$arr=explode(" ",$str);
print_r($arr);
?>
</body>
</html>
```

Output



Q.13 What is the use of foreach statement in PHP ?

Ans. : • The **foreach** function is used to iterate through all the elements of the loop. The syntax of foreach statement is as follows -

```
foreach($array as $value)
{
  statements to be executed
}
```

For example -

PHP Document[ArrayFunDemo6.php]

```
<?php
$mylist = array("Hello", "PHP","You","Are","Wonderful!!!");

foreach($mylist as $value)
{
    print("The current value of the array is <b>$value</b>");
    print "<br/>";
}
?>
```

5.7 : Functions

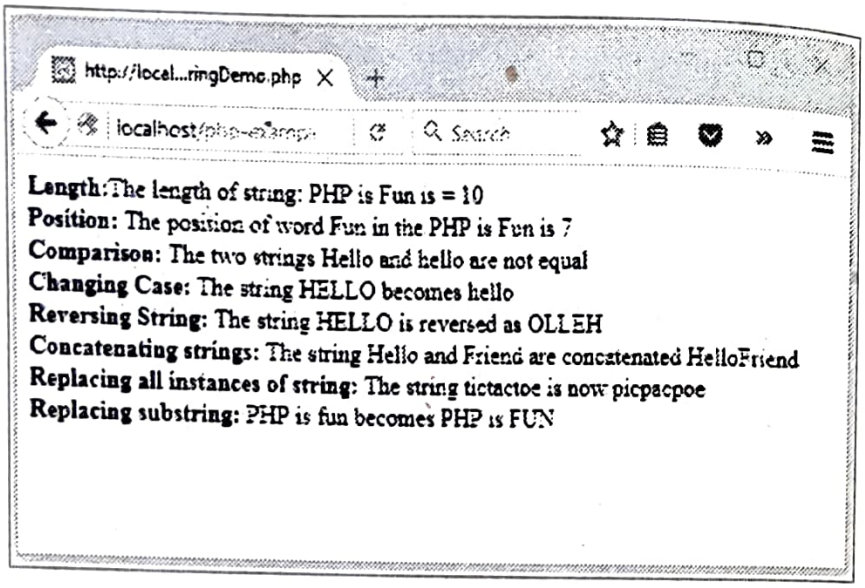
Q.14 Write a PHP program to do string manipulations.

Ans. : For this program we will apply various built in string manipulating functions to the string. The PHP code is as follows -

```
<?php
    $Str1="PHP is Fun";
    $length = strlen($Str1);
    echo "<b> Length:</b>The length of string: $Str1 is = $length";
    echo "<br/><b>Position: </b>The position of word Fun in the $Str1
is ".strpos($Str1,'Fun');
    $Str1="Hello";
    $Str2="hello";
    if(strcmp($Str1,$Str2))
        echo "<br/><b>Comparison: </b> The two strings $Str1 and
$Str2 are not equal";
    else
        echo "<br/><b>Comparision: </b> The two strings $Str1 and
$Str2 are equal";
    $Str1="HELLO";
    echo "<br/><b>Changing Case: </b> The string $Str1 becomes
".strtolower($Str1);
```

```
echo "<br/><b>Reversing String: </b> The string $Str1 is reversed as ".strrev($Str1);
$Str1="Hello";
$Str2="Friend";
echo "<br/><b>Concatenating strings: </b> The string $Str1 and $Str2 are concatenated ".$Str1.$Str2;
echo "<br/><b>Replacing all instances of string: </b> The string tictactoe is now ";
echo str_replace("t","p","tictactoe");
$Str1="PHP is fun";
$newstring=substr_replace($Str1,"FUN",7,9);
echo "<br><b>Replacing substring: </b> $Str1 becomes $newstring";
?>
```

Output



5.8 : Pattern Matching

Q.15 Explain the use of preg_match and preg_split functions used in PHP.

Ans. : • There is a **preg_match** function which is used to check the matching of the pattern from the text. It takes two parameters the first parameter is the pattern to be matched and the second parameter is the text against which the pattern is to be matched.

- The **preg_split** is the function that splits the string into the chunks.
- Following is a PHP script in which both of these functions are discussed.

PHP Document[PatternMatchDemo.php]

```
<?php
print "<h3>Use of preg_match function</h3>";
if (preg_match("/gram/", "I like Programming in PHP!"))
    print "The pattern is matching <br />";
else
    print "The pattern does not match <br />";
print "<h3>Use of preg_split function</h3>";
$text ="Hello:friends:how:are:you?";
print "The original string is<b> $text</b>";
print "<br/>";
print "The split is as follows ...<br/>";
$chunks=preg_split("/:/",$text);
while($myval=each($chunks))
{
    $val=$myval["value"];
    print("<b>$val</b>");
    print "<br/>";
}
?>
```

5.9 : Form Handling

Q.16 Explain form handling technique in PHP.

Ans. : • PHP is used for form handling. For that purpose the simple form can be designed in HTML and the values of the fields defined on the form can be transmitted to the PHP script using GET and POST methods.

- For forms that are submitted via "GET" method, we can obtain the form via the **$_GET** array variable.
- For forms that are submitted via "POST" method, we can obtain the form via the **$_POST** array variable.

Q.17 Create HTML form with one textbox to get user's name. Also write PHP code to show length of entered name when, the HTML form is submitted.

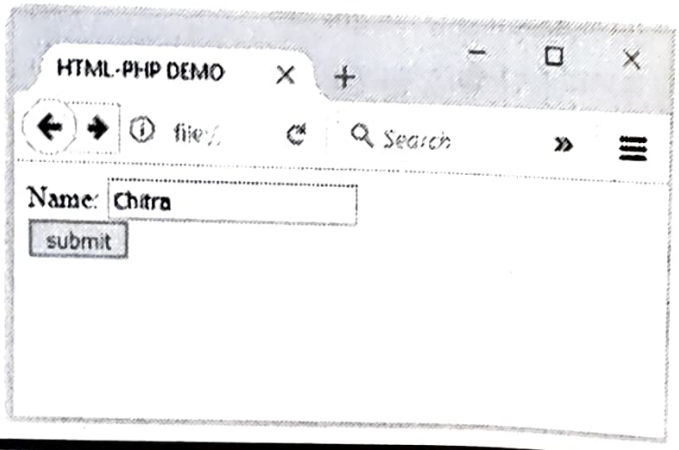
Ans. : Step 1 : The HTML form can be created as follows.

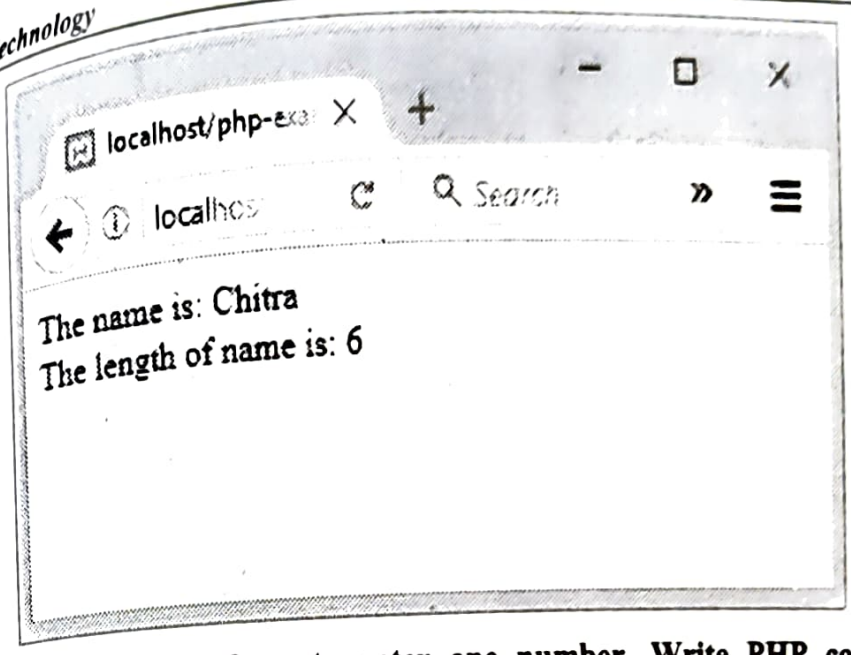
```
<!DOCTYPE html>
<html>
<head><title> HTML-PHP DEMO</title>
</head>
<body>
   <form method="post"
   action="http://localhost/getdata.php">
     Name: <input type="text" name="myname" size="20"/>
     <br/>
     <input type="submit" name="submit"
     value="submit"/>
   </form>
</body>
</html>
```

Step 2 : The PHP script to display the length of submitted name is as written below.

```
<?php
print "The name is: ";
print $_POST["myname"];
$len= strlen($_POST["myname"]);
print "<br/> The length of name is: ";
print $len;
?>
```

Output





Q.18 Create HTML form to enter one number. Write PHP code to display the message about number is odd or even.

Ans. : Step 1 : The HTML form for accepting number is created as below -

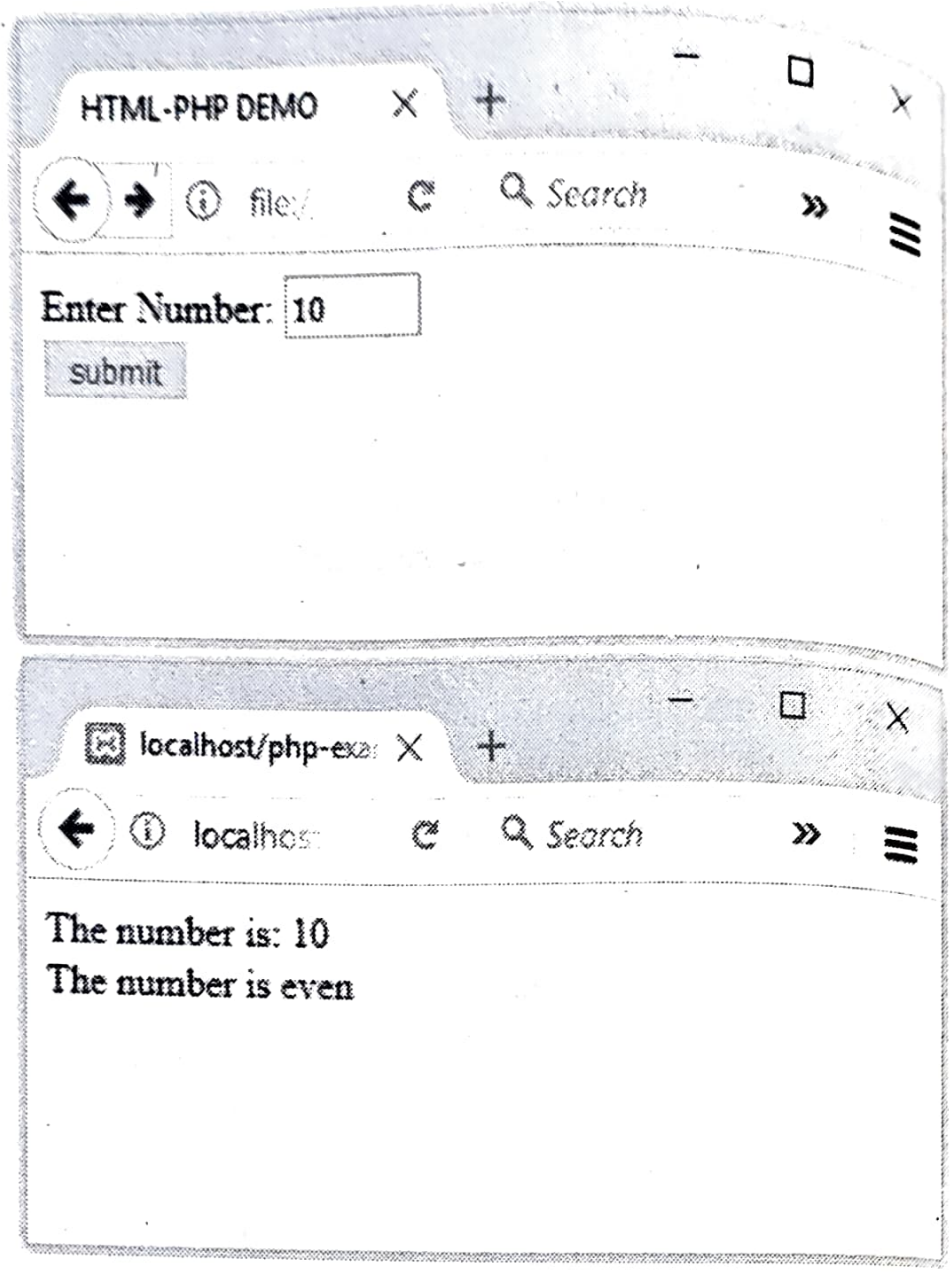
```
<!DOCTYPE html>
<html >
 <head><title> HTML-PHP DEMO</title>
</head>
<body>
  <form method="post" action="http://localhost/getdata.php">
   Enter Number: <input type="text" name="mynum" size="5"/>
    <br/>
  <input type="submit" name="submit" value="submit"/>
  </form>

</body>
</html>
```

Step 2 : The PHP script deciding whether the number is even or odd is as given below -

```
<?php
print "The number is: ";
print $_POST["mynum"];
$a= $_POST["mynum"];
if($a%2==1)
 print "<br/> The number is odd ";
else
```

```
print "<br/> The number is even ";
?>
```

Output



5.10 : Files

Q.19 Explain server side include in PHP with sample code.

[SPPU : Dec.-18, Marks 6]

Ans. : The server side 'include' is used to include a file on the basis of given path.

Syntax

```
include 'filename'
or
    include('filename')
```

For example

Step 1 : Create an HTML file as follows –

Test.html

```
<a href ="www.google.com">Search here</a>
```

Step 2 : Now create a PHP file that includes above **Test.html** file

Sample.php

```
<?php include("Test.html"); ?>
<h1> Welcome to this site </h1>
```

Q.20 Explain the open operation of file in PHP.

Ans. : • The first step in file handling is opening of the file.

- It takes two parameters - The first parameter of this function contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened.

Modes	Description
r	Read only. Starts reading from the beginning of the file.
r+	Read/Write. Starts reading from the beginning of the file.
w	Write only. Opens and clears the contents of file; or creates a new file if it is not created
w+	Read/Write. Opens and clears the contents of file; or creates a new file if it is not created.
a	Append. Opens and writes to the end of the file or creates a new file if it is not created.
a+	Read/Append. Preserves file content by writing to the end of the file.

For example :

```
$my_file = 'file.txt';
$file_handle = fopen($my_file, 'a') or die('Cannot open file:
'.$my_file);
```

The **fopen** function returns TRUE if the required file is opened.

Q.21 Explain the read and write operation on files in PHP.

Ans. : Reading from file

- The **fread** is the function which is used to read the file.
- It takes two parameters. The first parameter is the handle to the file and the second parameter is the number of bytes to be read. The **filesize** is the function which takes the filename as the parameter. For example -

```
$mystring = fread($file_handle, filesize("file.txt"));
```

- There is another function named **file_get_contents** using which the contents of the file can be obtained.
- The **fgets()** function is used to read a single line from the file. For example, following code displays the contents of the file line by line.

```
while(!feof($file_handle))
{
  echo fgets($file_handle). "<br />";
}
```

Writing to a file

- The **fwrite** is the function which is used to write the contents to the file.
- It takes two parameters - the first parameter is the handle to the file and the second parameter is the number of bytes to be written. For example -

```
$Written_string = fread($file_handle, $my_data);
```

5.11 : Cookies

Q.22 What cookies in PHP ? Explain.

Ans. : • Cookie is a small file that server embeds in the user's machine.

- Cookies are used to identify the users.

- A cookie consists of **a name** and **a textual value**. A cookie is created by some software system on the server.
- PHP can be used to create and retrieve the cookies.
- The cookie can be set in PHP using the function called **setcookie()**
- The syntax for the cookie is -

  setcookie(name,value,expire_period,path,domain)

Q.23 How to create and read cookies in PHP ?

Ans. : 1. Creation of Cookies

```
<?php
$Cookie_period=time()+60*60*24*30;
setcookie("Myname", "Monika", $Cookie_period);
?>
```

2. Reading Cookies

We can retrieve the cookie and read the value to ensure whether or the cookie is set.

PHP Document[CookieReadDemo.php]

```
<html>
 <head><title>Reading Cookies</title>
<body>
 <?php
 if (isset($_COOKIE["Myname"]))
  echo "<h3>Welcome " . $_COOKIE["Myname"]."!!!</h3>";
 else
  echo "<h3>Welcome guest!</h3>";
 ?>
 </body>
 </html>
```

5.12 : Session Tracking

Q.24 Explain functions in PHP with example and session management.

☞ [SPPU : May-22, Marks 9]

Ans. : • Sometimes the information about the session is required by the server. This information can be collected during the session. This process is called **session tracking.**

- In PHP, session state is available to the developer using the $_SESSION variable. The unique ID for the sessions can be stored in superglobal array $_SESSION.
- PHP keeps track of session by using a function called **session_start()**. Due to the call to **session_start()** function the session ID is created and recorded.
- Following is a simple PHP script in which the information about session is tracked.

PHP Document[SessionDemo.php]

```
<?php
session_start();
if(isset($_SESSION['pgvisit']))
{
    $_SESSION['pgvisit']=$_SESSION['pgvisit']+1;
echo "<h3>You are visiting this page for <i>". $_SESSION['pgvisit']."</i>
times</h3>";
}
else
{
    $_SESSION['pgvisit']=1;
echo "<h3>You are visiting this page for the first time</h3>";
}
?>
```

Q.25 Explain how session and cookies are used for session management in PHP. ☞ **[SPPU : Dec.-19, Marks 8]**

Ans. : Cookies : Refer Q.23.

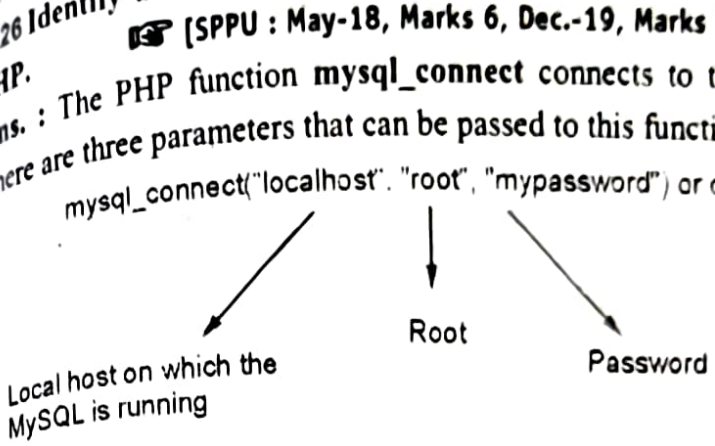
Session : Refer Q.24.

5.13 : Using MySQL with PHP

Q.26 Identify and explain steps involved in connecting to MySQL with PHP. **[SPPU : May-18, Marks 6, Dec.-19, Marks 8, May-22, Marks 9]**

Ans. : The PHP function **mysql_connect** connects to the MySQL server. There are three parameters that can be passed to this function. For example -

```
mysql_connect("localhost", "root", "mypassword") or die(mysql_error());
```



For example

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
else
{
print"connected";
}
mysql_close($conn); //closing the database
?>
```

The database can be closed using the function **mysql_close**

Q.27 Write a PHP code to insert data into the database created in MySQL.

Ans. : Using PHP we can insert the data in the database created in the MySQL.

For example -

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","mypassword");
```

```
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn); //select the database
$query=" INSERT INTO my_table (id,name) VALUES(1,'SHILPA')";
mysql_query($query,$conn);//Execution of Query
$query=" INSERT INTO my_table (id,name) VALUES(2,'MONIKA')";
mysql_query($query,$conn);//Execution of Query

mysql_close($conn); //closing the database
?>
```

Q.28 Create a HTML form "result.html" with a text box and a submit button to accept registration number of the student. Write a "result.php" code to check the status of the result from the table to display whether the student has "PASS" or "FAIL" status. Assume that the MYSQL database "my_db" has the table "result_table" with two columns REG_NO and STATUS.

Ans. : Step 1 : Create a database named my_db. Create a table result_table for this database and insert the values in this table. The table is created as follows -

Sr. No.	REG_NO	STATUS
1	101	PASS
2	102	FAIL
3	103	PASS
4	104	FAIL
5	105	PASS

Table Q.28.1

Step 2 : Create an HTML form to accept the registration number; the HTML document is as follows -

```
result.html
<!DOCTYPE html>
<html>
```

```
<head>
   <title> STUDENT RESULT </title>
</head>
<body>
   <form name="myform" method="post"
action="http://localhost/php-examples/result.php">
    <input type="text" name="reg_no"/>
    <input type="submit" value="Submit"/>
   </form>
</body>
</html>
```

Step 3 : Create a PHP script to accept the registration number. This php script will connect to MYSQL database and the status (PASS or FAIL) of the corresponding registration number will be displayed.

result.php

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("my_db",$conn); //select the database
//Execution of Query for displaying the data
$reg_no = intval($_POST['reg_no']);

$result=mysql_query("SELECT REG_NO,STATUS FROM result_table
where REG_NO=$reg_no ");
 while($row = mysql_fetch_array($result))
  {
   echo $row['REG_NO'] . " is " . $row['STATUS'];
   echo "<br />";
  }
 mysql_close($conn); //closing the database
 ?>
```

5.14 : WAP and WML

Q.29 Explain in detail WAP architecture and WML.

[SPPU : May-22, Marks 9]

Ans. : WAP architecture :

- Wireless Application Protocol (WAP) is a set of standards designed to extend Internet services to mobile phones and Personal Digital Assistants (PDA).
- The mobile network gets connected to the Internet through **WAP Gateway**. The function of WAP gateway is to convert the **WAP request** to a **WEB request** or a WEB request to a WAP request.
- The mobile phones act as a client when gets connected to the internet. The demanded web pages can be displayed on the mobile phones if it is WAP enabled.
- The Web pages are normally written in HTML language but the web pages written in HTML gets loaded very slowly on the mobile phones. Hence the web pages for mobile phones are written in the language called **Wireless Markup Language (WML)**
- The WAP model follows the OSI model and is represented by Fig. Q.29.1.

Application layer
Session layer
Transaction layer
Security layer
Transport layer
Bearer interfaces

Fig. Q.29.1 WAP model

Application Layer

- The WAP's application layer is the Wireless Application Environment (WAE).
- The WAE supports WAP application development using the Wireless Markup Language (WML).
- The WAE also includes the Wireless Telephony Application Interface (WTA) which provides the programming interface to telephones. This allows WAP technology to support for sending messages, initiating calls and other networking capability.

Session Layer

- The session layer makes use of **Wireless Session Protocol** (WSP).
- This protocol is similar to HTTP. But HTTP is not efficient for displaying the web pages on mobile devices, WAP makes uses of WSP.
- The WSP conserves the bandwidth on the wireless links.
- The HTTP works with textual data whereas the WSP work with binary data.
- It supports for both **connection-oriented** and **connectionless** sessions.

Transaction, Security and Transport Layer

There are three different protocols used in these layers.

- **Wireless Transaction Protocol (WTP)** provides transaction level services. These services are used for both reliable and unreliable transports. If packets gets dropped then WTP performs the retransmission of packets. Similarly it avoids transmission of duplicate copies of packets.
- **Wireless Transaction Layer Security (WTLS)** is just similar to Secured Socket Layer (SSL). It performs the authentication and encryption functionality.

- **Wireless Datagram Protocol (WDP)** is similar to User Datagram Protocol (UDP). WDP offers to the upper layers an invisible interface independent of the underlying network technology used.
- At the lowest layer of the stack, there are **bearer** services. WAP supports dial-up networking using IP and Point to Point Protocol (PPP) as a bearer service. It also supports for Short Message Service (SMS) and General Packet Radio Service (GPRS).

WML :

- The topmost layer in the WAP architecture is made up of WAE (Wireless Application Environment), which consists of WML and WML scripting language.
- WML stands for **W**ireless **M**arkup **L**anguage.
- The role of WML is the same as that of HTML in web applications. 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 is an application of XML, which is defined in a document-type definition.

For example -

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="one" title="Welcome Card">
<p>
Hello, This is first card.
</p>
</card>
<card id="two" title="GoodBye Card">
<p>
Good bye, this is second card.
</p>
</card>
</wml>
```

5.15 : Introduction to ASP.NET

Q.30 Write short notes on : Overview of ASP. NET.

[SPPU : May-22, Marks 5]

Ans. : Active Server Pages (ASP) is a server side scripting language. It is a part of **Internet Information Server** (IIS). This is basically a product of Microsoft.

The ASP.NET is a server side scripting technology offered by Microsoft. It is a powerful scripting language for creating dynamic and interactive web pages. The ASP.NET is next generation ASP but it is entirely new technology. The ASP.NET runs under IIS server.

Working

Step 1 : When user demands for the ASP.NET file, the web browser of the client's PC sends this request to IIS server.

Step 2 : The IIS server passes this request to .NET Engine.

Step 3 : The .NET engine reads the file line by line and generates the script in a file.

Step 4 : The requested .NET file is returned to the web browser.

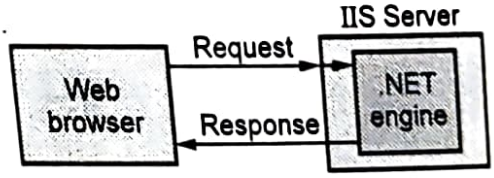


Fig. Q.30.1

Features of ASP.NET

- ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.
- ASP.NET pages are called **web forms.** These web forms have the extensions **.aspx.**
- These files contain static XHTML markup.
- The static and dynamic content can be inserted in the ASP.NET code.

5.16 : Overview of the .NET Framework

Q.31 Write short note on : overview of .NET framework.

Ans. : • The .NET framework is a software development platform developed by Microsoft. It is designed for building and running windows applications.

- The first version of .NET framework was released in the year 2002. This version was known as .NET framework 1.0.
- The .NET framework is used to create both form based and web based applications.
- The .NET framework is also useful in building the web services.
- It supports various programming languages like C#, VB.NET, Visual Basic and many more. Developers can choose and select the language to develop the required application.

Q.32 Draw and explain the architecture of .NET framework.

Ans. : The .NET framework architecture consists of three major components -

1. CLR
2. Library
3. Language

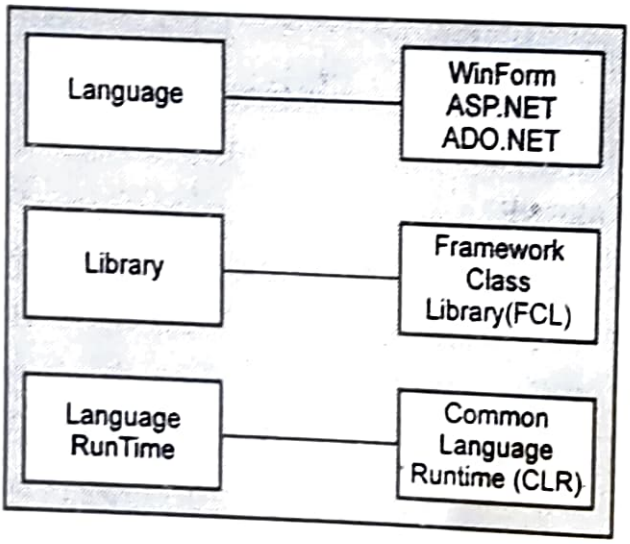


Fig. Q.32.1 The .NET framework

1. CLR

- The CLR stands for Common Language Runtime. It is a platform on which the .NET programs are executed. Any .NET programming language can be executed by this platform.
- Just in Time (JIT) is a part of CLR. It converts the source code into native intermediate code and then into machine instructions.
- The CLR is also responsible for performing some other tasks such as exception handling, memory management and garbage collection.
- The important features of CLR are interoperability, security and portability.

2. Framework Class Library (FCL)

- It is a standard library which is actually a collection of various classes, interfaces, and namespaces that can be used to build the application.
- It is the key component of .NET framework that provides the core functionalities of .NET.
- Most of the methods of classes are split either into System.* or Microsoft.* namespaces. A namespace is a logical separation of methods.

3. Language

- The .NET supports many languages. Developer can choose the desired language for development of required application.
- Following are the categories of various types of applications that can be developed using .NET
  - **WinForms :** This is useful for developing form based applications.
  - **ASP.NET :** The web based applications can be developed using ASP.NET. These applications will run on the web browsers such as Google Chrome, Mozilla FireFox, Internet Explorer and so on. The IIS server is responsible for executing the ASP.NET application which is a part of Microsoft component.

- **ADO.NET** : This technology is used to interact with the databases like Microsoft SQL servers, Oracle and so on. The most commonly used languages for this category is C# and VB.NET.

5.17 : Overview of C#

Q.33 Write short notes on : Overview of C#. ☞ **[SPPU : May-22, Mark 4]**

Ans. : • The C# is a modern, **object oriented programming** language developed by Microsoft.

- C# was developed by Anders Hejlsberg during the development of .NET framework.
- C# is designed for **Common Language Infrastructure (CLI)**, which consists of the executable code and runtime environment that allows use of various high-level languages to be used on different computer platforms and architectures.
- Various **features** of C# are
  1. Object Oriented
  2. Standard Library
  3. Automatic Garbage Collection
  4. Boolean Condition
  5. Properties and Events
  6. Simple Multithreading
  7. Integration with Windows.
- **Example of C# Program :** Refer Q.34.

Q.34 Write a C# program to display welcome message to user.

Ans. :

```
using System;

namespace MyFirstApplication
{
  class Program
  {
```

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

5.18 : Introduction to ASP.NET

Q.35 Write short note on - ASP.NET.

Ans. : • ASP.NET is a platform for developing web applications. It works on **HTTP protocol** and makes use of HTTP commands.

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

  1. C#
  2. Visual Basic .NET
  3. Jscript
  4. J#

- ASP.NET consists of large number of controls such as textboxes, buttons, labels and checkboxes and so on for developing the web pages. Basically the HTML tags are used to manipulate the code.

ASP.NET Component Model

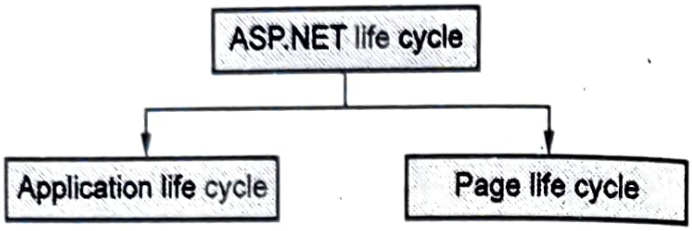
- The ASP.NET component model provides various building blocks of the ASP.NET pages. It describes the -

  1. Server side **pages** that uses the HTML tags
  2. Server **controls** that are used for building complex web applications.

- In ASP.NET programming the **.aspx page** uses the HTML tags. During execution, the .aspx page is transformed into an **instance of a class** which inherits from the base class Page of the .Net framework.
- In this technology **each ASP.Net page is an object** and all its components i.e., the server-side controls are also objects.

Q.36 Explain in detail the life cycle of ASP.NET application.

Ans. : The life cycle of ASP.NET application is divided into two categories :



Application life cycle

There are mainly two phases in the application life cycle -

1. Request phase

a) User makes a request for accessing particular resource.

b) Browser sends this request to the web server.

c) An object of **ApplicationManager** class is created.

d) An object of the **HostingEnvironment** class is created to provide information regarding the resources.

e) Required items in the application are compiled.

2. Response Phase

a) Response objects are created.

b) There are some application objects of the class **HttpContext, HttpRequest** and **HttpResponse.** These objects are initialized.

c) The request is processed by the **HttpApplication** class. Hence an instance of the **HttpApplication** object is created and assigned to the request.

d) Different events are raised by this **HttpApplication** class for processing the request.

3. Page life cycle

Various stages of page life cycle are -

1. **Page request :** The page request occurs before the page life cycle begins. When the page is requested by a user, ASP.NET determines whether the

page needs to be parsed and compiled or whether a cached version of the page can be sent in response without running the page.

2. **Start :** In stage the page properties such as **Request** and **Response** are set. At this stage, the page determines whether the request is postback or it is a new request. It then sets the **IsPostBack** property. At this stage the **UICulture** property is also set.
3. **Initialization :** In this stage controls on the page are available. Each control's **UniqueID** property is set. A master page and themes are set. For a new request postback data is loaded and the control properties are restored to the view-state values.
4. **Load :** At this stage, the control properties are loaded with information obtained from view state and control state.
5. **Postback event handling :** If the request is a postback, then control event handlers are called. Then to validate the controls the **Validate** method is called.
6. **Rendering :** At this stage, the page calls the **Render** method for each control. The output of rendering is written to the **OutputStream** class of the Page's Response property.
7. **Unloading :** The Unload event is raised after the page is fully rendered. At this point, page properties such as Response and Request are unloaded and cleanup is performed.

5.19 : ASP.NET Controls

Q.37 Enlist the ASP.NET controls.

Ans. : Various controls that can be used by the web forms are -

1. Button controls
2. Text boxes and labels
3. Check boxes and Radio buttons
4. List controls
5. List items

Q.38 Write ASP.NET code to demonstrate the CheckBox and ListBox control.

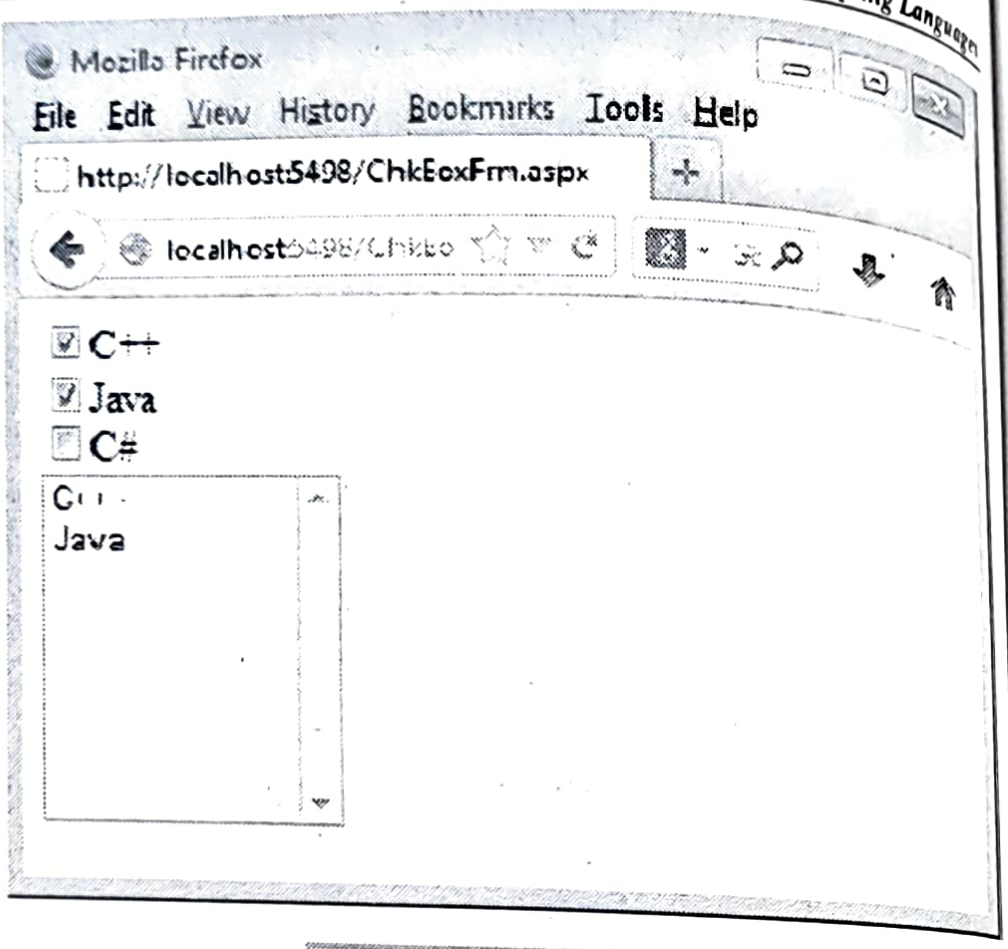
Ans. :

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="ChkBoxFrm.aspx.cs" Inherits="CheckBoxDemo.ChkBoxFrm"
%>

<script runat="server">
void CheckBox1_CheckedChanged(object sender, EventArgs e)
{
  if (CheckBox1 .Checked==true)
  {
    ListBox1.Items.Add(CheckBox1.Text);
  }
  else
  {
    ListBox1.Items.Remove(ListBox1.Items.FindByText(CheckBox1.Text));
  }
}
void CheckBox2_CheckedChanged(object sender, EventArgs e)
{
  if (CheckBox2.Checked == true)
  {
    ListBox1.Items.Add(CheckBox2.Text);
  }
  else
  {
    ListBox1.Items.Remove(ListBox1.Items.FindByText(CheckBox2.Text));
  }
}
void CheckBox3_CheckedChanged(object sender, EventArgs e)
{
  if (CheckBox3.Checked == true)
  {
    ListBox1.Items.Add(CheckBox3.Text);
  }
  else
  {
    ListBox1.Items.Remove(ListBox1.Items.FindByText(CheckBox3.Text));
  }
```

```
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <asp:CheckBox ID="CheckBox1" runat="server"
AutoPostBack="True"
      oncheckedchanged="CheckBox1_CheckedChanged" Text="C++"
/>

  </div>
  <div>
    <asp:CheckBox ID="CheckBox2" runat="server" AutoPostBack="True"
      oncheckedchanged="CheckBox2_CheckedChanged" Text="Java" />
  </div>
  <div>
    <asp:CheckBox ID="CheckBox3" runat="server" AutoPostBack="True"
      oncheckedchanged="CheckBox3_CheckedChanged" Text="C#" />
  </div>
  <div>
  <asp:ListBox ID="ListBox1" runat="server" Height="140px"
Width="120px">
  </asp:ListBox>
  </div>
  </form>
</body>
</html>
```

Output

Just check or uncheck the checkboxes to experience the effect of this control.



5.20 : Web Services

Q.39 What is web services ? Enlist its features.

Ans. : • Web service is a software program using which the information can be exchanged. That means using web service one application can invoke the method another application. These applications can be on the same computer or different computers.

- The web services make use of the standard protocols such as HTTP, XML and SOAP. As these are open and standard protocols the applications can exchange the information on any platform with the help of web services. For instance - Web service built using .NET application can be consumed using JSON or Java application.

Features

1) Web service is a language independent.

2) It is platform independent.

3) It is based on XML.

4) It is discoverable. That means, applications and developers can search for and locate the desired web services through registries.

5) It is based on a stateless service architecture.

5.21 : Overview of Node JS

Q.40 Write short note on NodeJS. ☞ **[SPPU : May-18, Marks 4]**

Ans. : • NodeJS is an open source technology for server.

- Using Node.js we can run JavaScript on server.
- It runs on various platform such as Windows, Linux, Unix, and MacOS.

Uses of Node.js

- It can perform various tasks such as -
  1. It can create, open, read, delete, write and close files on the server.
  2. It can collect form data.
  3. It can also add, delete, modify data in databases.
  4. It generate dynamic web pages.

END...

Unit VI

6 Ruby and Rails

6.1 : Introduction to Ruby : Origins and uses of Ruby

Q.1 Explain origin and uses of Ruby.

Ans. : • Ruby is a scripting language designed by Yukihiro Matsumoto also known as Matz.

- Ruby executes on variety of platforms such as Windows, Linux, Mac and so on.
- Ruby is a pure object oriented programming language. Ruby is an open source and it is freely available on the web.
- Ruby is similar to smalltalk, python, Perl.
- Ruby has the pattern matching feature which is similar to the Perl.
- Ruby can be easily connected to the databases like MYSQL, DB2 and Oracle.

6.2 : Scalar Types and their Operations

Q.2 Explain the scalar types and their operations in Ruby.

☞ **[SPPU : May-22, Marks 8]**

Ans. : • There are three types of data types used in Ruby and those are Scalar, Arrays and hashes.

- The scalar types are further classified as **numeric** and **character strings**. Refer Fig. Q.2.1.

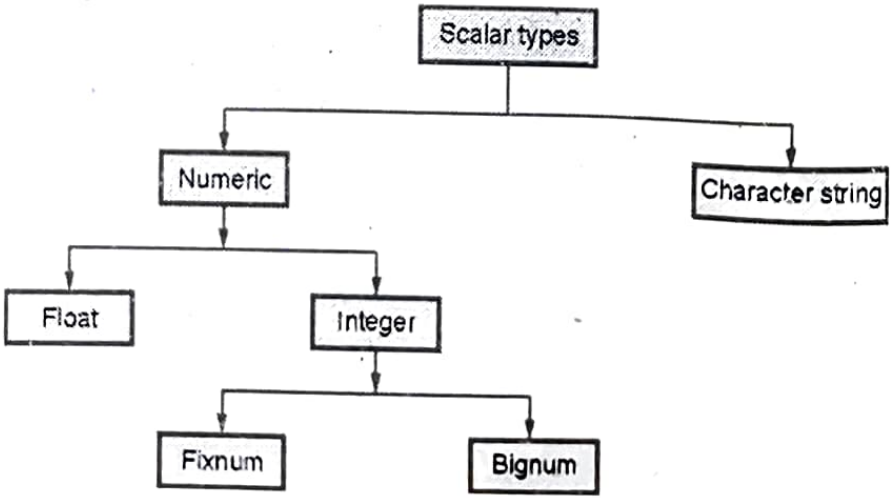


Fig. Q.2.1 Scalar types

Numeric and String Literals

- All the numeric data types in Ruby are derived from the **Numeric** class.
- The immediate children of Numeric class are **Float** and **Integer**.
- The **Integer** class has two child classes - **Fixnum** and **Bignum**.
- The integer objects which lies within the size of 32 bit is called the **Fixnum** object and the object which is greater in size of 32 bit is called the **Bignum** object.
- If a **Fixnum** object whose size grows beyond the 32 bit then it is implicitly typecast to **Bignum** object.
- Similarly in expression evaluation the size of particular Bignum object becomes less than 32 bit then it is converted to the **Fixnum** object.
- The integer literal may contain the underscore character to read out the large integer conveniently. For example 231456981 can be read using the underscore as 231_456_981.
- The numeric literal that contains the floating point is called the **Float** object. It denotes the double precision floating type value.
- The decimal point must be preceded and followed by at least one digit. For example .22 will be illegal literal in Ruby.
- The **string** object consists of the string literals that are nothing but the collection of characters.
- There are two types of strings - single quoted and double quoted string.
- If we want to print some character literally like apostrophe s then the \ is used.

6.3 : Simple Input and Output

Q.3 Write a Ruby program to display addition of two numbers.

Ans. :

```
puts "Enter the value of a"
a=gets.to_i
puts "Enter the value of b"
```

```
b=gets.to_i
c=a+b
puts "The addition of two numbers a and b is #{c}"
```

6.4 : Control Statements

Q.4 Explain how multiple selection constructs are implemented in Ruby.

Ans. : • The control statements are for controlling the flow of control of the program.

- The **if statement** is of following form -

```
Syntax:
If expression
  Statement
end
```

→

```
Example:
If count<10
  put's "Hello"
end
```

- The if **else statement** is -

```
Syntax:
if expression
        Statement
elsif expression
        Statement
else
        Statement
end
```

→

```
Example:
if age<5
        puts "Kid"
elsif age<21
        puts "Youth"
else
        puts "Adult"
end
```

- Ruby allows two kinds of **multiple selection constructs**. Both named **case**. The first form of **case** statement is as follows -

```
Syntax:
case
when value then
        statement sequence
when value then
        statement sequence
...
else Expression
end
```

→

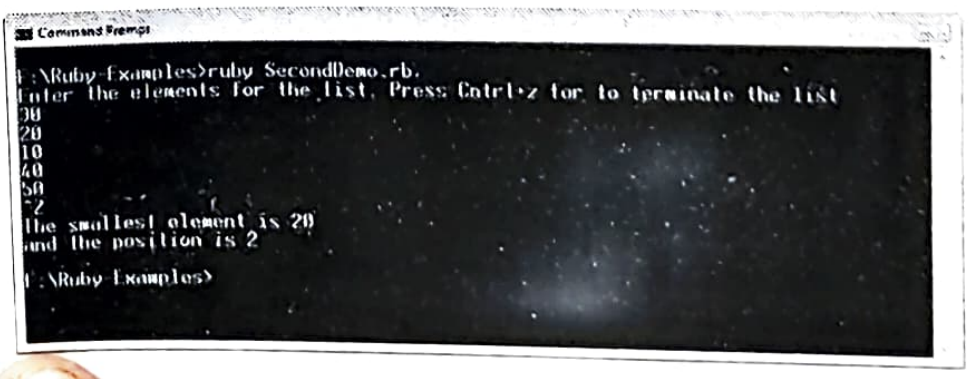
```
Example:
case
when 66 then
        puts "Distinction"
when 60 then
        puts "First Class"
when 50 then
        puts "Pass Class"
else puts "Fail"
end
```

Q.5 Write a Ruby program that inputs the list of numbers from keyboard and finds the second smallest number in the list, along with its position in the list.

Ans. : Ruby Program[SecondDemo.rb]

```
i=0
arr=Array.new
puts "Enter the elements for the list. Press Cntrl+z for to terminate the list"
while(val=gets)
 arr[i]=val
 i+=1
end
first = arr[0]
 second = arr[0]
 i=0
 for value in arr
  if value < first
   second = first
   secondkey=i
   first = value
  elsif value < second
    second = value
    secondkey=i
  end
  i+=1
 end
 puts "The smallest element is #{second}and the position is #{secondkey}"
```

Output



6.5 : Fundamentals of Arrays

Q.6 Explain the concept of arrays in Ruby with the help of example.

Ans. : • In Ruby the arrays are more flexible because of two reasons -

- The arrays in Ruby are **dynamic** that means the array can grow and shrink.
- The arrays in Ruby can store the elements of different data types.

- The arrays can be created in two different ways.
- The first way is to use the keyword **new**. For example :
  - Marks =Array.new
  - Marks=Array.new(10) ← the array of 10 locations gets created.
- The second way is to assign the list of literals to the variables. For example :
  - Student=[10,20,30,"Archana",[ ]]
- The subscript of array elements is integer and every array starts at the subscript 0.
- The **length** method is used to retrieve the length of the array.
- The **at** method returns the element present at the pointed subscript.

Ruby Script [ArrayDemo.rb]

```
A=[10,20,30,40,50]
puts "The array is : #{A}"
num=A.at(3)
puts "The element at 3rd index is :#{num}"
len=A.length
puts "The length of the array is :#{len}"
```

Output

```
The array is : [10, 20, 30, 40, 50]
The element at 3rd index is :40
The length of the array is :5
```

Q.7 Write a Ruby program that sorts the array and then searches the desired element from the array.

Ans. :

```
i=0
A=Array.new
puts "Enter the elements in the array"
while(i<5)
 val=gets
 A[i]=val
 i+=1
end
A.sort!
puts "The sorted array is "
for val in A
 puts val
end
puts "Serching the elements from an array"
puts "Enter the number to be searched"
key=gets
if A.include?(key)
 puts "The element #{key} is present in the list"
else
 puts "The element #{key} is not present in the list"
end
```

6.6 : Hashes

Q.8 What is Hashes in Ruby ? Give difference between Hashes and Arrays.

Ans. : • A hash is a collection of key-value pairs like this : "Roll" =>10

- Hashes are also called as **associative arrays.**
- The hashes are similar to arrays but the only difference between the two is that,

Hashes	Arrays
The hash uses the string values as a subscript to address the elements. These string values are basically the keys.	The arrays use the integer values as a subscript to address the element.

The elements in hash are not ordered. The arrangement of elements in hash resembles the **set.**	The elements in array are ordered by subscript. The arrangement of elements in array resembles the **list.**

- The hash can be created using either the **new** method or by assigning the key-value pair. For example :

Ruby Program [HashDemo1.rb]

```
A = Hash.new( "one" )
puts "\tFirst Method of Hash Creation"
puts "#{A[0]}"
puts "#{A[3]}"
puts "#{A[5]}"
puts "\tSecond Method of Hash Creation"
B=Hash["AAA"=>100,"BBB"=>200,"CCC"=>300]
puts "#{B['AAA']}"
puts "#{B['BBB']}"
puts "#{B['CCC']}"
puts "\t Displaying the keys of the array"
keys=B.keys
puts "#{keys}"
```

6.7 : Methods

Q.9 Explain the concept of methods used in Ruby.

Ans. : • Methods in Ruby are similar to the functions.

- When a block of statements that get repeated many times in the program then the method is used.
- The method is defined outside the class.
- The method definition includes **method header** and sequence of statements.
- The **method header** consists of the reserved word **def.**
- **The method name must begin with lower case.**

- The syntax of method definition is as given below -

```
def method_name ( argument list )
  method body
end
 or
def method_name
  method body
end
```

- If the method has no parameter then the parenthesis can be omitted.

Ruby Program[MethodDemo.php]

```
def mymethod
  puts " This is the method demo"
end
mymethod
```

Output

```
This is the method demo
```

6.8 : Classes

Q.10 Explain classes and objects in Ruby with appropriate examples.

[SPPU : May-22, Marks 9]

Ans. : • Every class contains two things the variables and the functions. The class in the Ruby can be defined as follows -

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

- The class name must begin with the capital letter. The keyword **class** can be used to define the class. The class must be terminated by the keyword **end**.
- In Ruby there are two types of variables in class -
  - **Instance Variables :** The instance variables are defined in the method. The instance variables change from object to object. These variables are available across the methods. The syntax of these variables is ,

```
@variable_name
```

- **Class Variables :** The class variables belongs to the class. They are used to define the characteristics of the class. These are available across the objects. The syntax of these variables is,

  @@variable_name

- **Objects** are the instances of the class and these can be created using the method **new.**

  For example :

  Obj1= Myclass.new

- There is a method named **initialize** is a constructor of the class which initializes the values of the variables. Using the **def** and the **end** the method initialize can be defined.
- The name of the method must be starting with the lower case letter. It must be ended by the keyword **end.**

Example : Create a class for obtaining the roll number,name and marks of the student. Display the records.

```
class Student
  @@TotalStudents=0
  def initialize(roll, name, marks)
    @stud_roll=roll
    @stud_name=name
    @stud_marks=marks
  end
  def display()
    puts "Roll Number: #@stud_roll"
    puts "name: #@stud_name"
    puts "Marks: #@stud_marks"
  end
  def total()
    @@TotalStudents += 1
    puts "\tTotal number of students: #@@TotalStudents"
  end
end
S1=Student.new("1", "Shilpa", 97.11)
S2=Student.new("2", "Rashmi", 89.21)
# Call Methods
```

Class Variables

Instance Variables

```
s1.display()
s1.total()
puts "----------------------------------------"
s2.display()
s2.total()
```

Q.11 Explain the implementation of inheritance using Ruby with suitable example.

Ans. : • Inheritance is the most important feature of any object oriented programming language.

- By inheritance the two classes are created one **base class** and another **derived class.**
- Using the object of derived class one can access the methods of itself and the base class both.
- The syntax for declaring the derived class is,

```
class Derived_class_name < Base_class_name
```

For example :

Ruby Program [InheritanceDemo.rb]

```
class A
 def method1
  puts "I am in method1 of class A."
 end
 def method2
  puts "I am in method2 of class A."
 end
end

class B < A
 def method3
  puts "I am in method3 of class B."
 end
end
obj = B.new
obj.method1
obj.method2
obj.method3
```

Class B is derived from class A

6.9 : Code Blocks and Iterators

Q.12 What is the use of code blocks and iterators in Ruby ?

Ans. : • Block consists of collection of code.

- The block is always enclosed within { and }.
- Blocks can be used with methods.
- Using the iterator **each** we can display each element of the array and apply it to block.
- For example :

Ruby Program [BlockDemo.rb]

```
A=[10,20,30,40]
A.each{|num| puts "The number is #{num}"}
```

Output

```
The number is 10
The number is 20
The number is 30
The number is 40
```

- Using the iterator **step** the specific values can be displayed. For Example

Ruby Program [BlockDemo1.rb]

```
10.step(20,2){|num| puts "The number is #{num}"}
```

Starting value

ending value

Step size

Output

```
The number is 10
The number is 12
```

he number is 14
he number is 16
he number is 18
he number is 20

- Using the **yield** method the block can be invoked.
- A block is always invoked from a function. But the name of the function and the name of the block must be the same in this case.
- For example :

Ruby Program [BlockDemo1.rb]

```
def myfun
  puts "This statement is in method"
  yield
  puts "Returning back to the method"
end
myfun {puts "This statement is in block"}
```

Block is called

Block is defined

6.10 : Pattern Matching

Q.13 Write short note on – pattern matching in Ruby.

Ans. : • For pattern matching the operator =~ is used.

- The pattern can be placed within /
- For example :

```
irb(main):001:0> T="Success"
=> "Success"
irb(main):002:0> T=~/es/
=> 4
irb(main):003:0>
```

- The **split** method is used to separate out the words from the given string. For example following commands will show how to separate out the words from the string using split command.

```
irb(main):005:0> s="hello friends how are you"
=> "hello friends how are you"
irb(main):006:0> w=s.split(/[ .,]\s*/)
=> ["hello", "friends", "how", "are", "you"]
irb(main):007:0>
```

6.11 : Overview of Rails

Q.14 Give an overview of Rails.

Ans. : • Rails is a development framework for web - based applications. As Rails are closely associated with Ruby it is often referred as **Ruby on Rails** or simply RoR.

- Rails is based on **MVC architecture** - i.e. Model View Controller.
- The MVC architecture clearly separates the logical and physical parts of the applications in three parts - Model, View and Controller.
- The **Model** part is for data and constraints on data. The **View** part prepares and presents the result or output. The **Controller** controls the application.
- The most significant characteristic of Rails is its approach to connecting object oriented software with relational database. But both the object oriented software and relational databases are not compatible to each other. Hence Rails make use of **Object-Relational Mapping (ORM) approach**. By this each relational database table is simply mapped to a class. For example if the database has a table named **Student** then it will be mapped to class **Student**. Thus ORM maps **tables to classes, rows to objects** and **columns to field of objects**. The ORM of rails is called **activeRecord**.
- The **View** and **Controller** part of MVC are supported by **ActionPack** components of Rails. The view documents are generated for three categories of web documents XHTML, XML and JavaScript and the Controller part of MVC interacts with data model, user and the view.

- Rails can be more commonly used with Ajax. It makes use of the JavaScript framework named prototype to support Ajax and interaction with JavaScript model.
- Rails uses MVC architecture, consists of designing and building three parts of an MVC system.

6.12 : Document Requests

Q.15 What is the meaning of dynamic document ? How it is implemented using Rails ?

Ans. : • Dynamic document is a kind of web document in which the contents are changing more often. For example if your web page is displaying the current date and time or stack market position or Cricket match score then that document is said to be a dynamic document.

- The **Ruby code** can be **embedded** within the template file by using <% and %> markers.
- If the Ruby code produces some result then the result can be inserted within the template document by <%= and %> marker.
- Ruby's **Time.now** is used to display the current time. It is as follows -

welcome.html

```
<html>
<head>
<title> FIRST RAILS PROGRAM</title>
</head>
<body>
<center>
<h3>Current Date and Time ... </h3>
<p> It is now <%= t = Time.now %> </p>
</center>
</body>
</html>
```

6.13 : Processing Forms

Q.16 Using form handling technique write a sample application in rails.

Ans. : Step 1 : The code for orderform.html file is as follows -

orderform.html

```
<html>
    <head>
      <title> Submitted Order</title>
    </head>
    <body>
    <h3> User Name: <%= @UserName %> </h3>
    <table border="1">
      <tr>
       <td><b>Product Name</b></td>
   <td><b>Price(Rs.)</b></td>
   <td><b>Quantity</b></td>
   <td><b>Cost</b></td>
      </tr>
      <tr>
        <td>Apples</td>
        <td>10 </td>
        <td><%= @qty1 %></td>
        <td><%= @apple_cost %></td>
      </tr>
      <tr>
        <td>Oranges</td>
        <td>7.50 </td>
        <td><%= @qty2 %></td>
        <td><%= @orange_cost %></td>
      </tr>
   </table>
```

```
<br/>
<h2> Total Cost = <%= @total_cost %> </h2>
</body>
</html>
```

Step 2 : Edit the controller file **order_controller.rb** as follows -

order_controller.rb

```
class OrderController < ApplicationController
 def orderform
 end
 #displayorder method - it fetches the data from form and compute
 #total cost

 def displayorder
 @UserName =params[:UserName]
 @qty1 = params[:qty1].to_i
 @qty2 = params[:qty2].to_i

 #Computing the cost of each item
 @apple_cost = 10 * @qty1
 @orange_cost = 7.50 * @qty2

 #computing the total cost of purchased items
 @total_cost = @apple_cost+@orange_cost

 #Converting the values into strings with two digits
 @total_cost = sprintf("%4.2f",@total_cost)
 @apple_cost = sprintf("%4.2f",@apple_cost)
 @orange_cost = sprintf("%4.2f",@orange_cost)
 end
 end
```

Invokes orderform.html

Invokes displayorder.html

Reading the input from the orderform.html and preparing the computed cost variables for displaying the result.

- Using the **params** symbol the parameters on the input rhtml file can be invoked. The **.to_i** method is used to convert the string to integers.

- The **sprintf** function of C can be used to format the numbers in two decimal places. This is done for displaying the floating point data in formatted manner.
- Save this file and close it. By the above code we have actually written the response methods namely **orderform** and **displayorder**. The **orderform.html** and **displayorder.html** files will be invoked as a response to these action methods. Out of these the **orderform.html** is already created.

Step 3 : Create a file named **displayorder.html**.

It is as follows -

displayorder.html

```
<html>
    <head>
      <title> Submitted Order</title>
    </head>
    <body>
    <h3> User Name: <%= @UserName %> </h3>
    <table border="1">
      <tr>
       <td><b>Product Name</b></td>
    <td><b>Price(Rs.)</b></td>
    <td><b>Quantity</b></td>
    <td><b>Cost</b></td>
      </tr>
      <tr>
        <td>Apples</td>
        <td>10 </td>
        <td><%= @qty1 %></td>
        <td><%= @apple_cost %></td>
      </tr>
```

```
    <tr>
      <td>Oranges</td>
      <td>7.50 </td>
      <td><%= @qty2 %></td>
      <td><%= @orange_cost %></td>
    </tr>
  </table>
  <br/>
  <h2> Total Cost = <%= @total_cost %> .</h2>
  </body>
</html>
```

6.14 : Rails Applications and Databases

Q.17 Develop a sample application in Rails to demonstrate the connectivity with the database.

Ans. : Step 1 : Now create a database named **bookinfo_development** using CREATE DATABASE command.

```
C:\Windows\System32\cmd.exe - mysql -u root -p
mysql> CREATE DATABASE bookinfo_development;
Query OK, 1 row affected (0.01 sec)

mysql>
```

Then create tables **books** and **subject**.

```
C:\Windows\System32\cmd.exe - mysql -u root -p
mysql> CREATE TABLE books (
    -> accNo INT(5) NOT NULL AUTO_INCREMENT,
    -> title CHAR(20),
    -> author CHAR(20),
    -> edition INT(3),
    -> publication CHAR(15),
    -> title_id INT(4) NOT NULL,
    -> PRIMARY KEY(accNo)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

```
mysql> CREATE TABLE subject (
    -> id INT(4) NOT NULL AUTO_INCREMENT,
    -> subject_name CHAR(20),
    -> PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Step 2 : Generate model files namely book.rb and subject.rb.

book.rb

```
class Book < ActiveRecord::Base
 belongs_to :subject
end
```

subject.rb

```
class Subject < ActiveRecord::Base
 has_many :books
end
```

The **has_many** and **belongs_to** are the relationships used in MySql for specifying the classes associated with the database tables. When these relationships are provided then Rails generate the capabilities to manipulate the given relationship.

Step 3 : Now insert the data into tables **books** and **subject.**

Step 4 : Edit the controller as follows -

mybook_controller.rb

```
class MybookController < ApplicationController
    def welcome
    end
    def show
    @titlename = params[:title]
    @bookarr = book.find(:all, :conditions => ["title =?",@titlename])
    @subarr = subject.find(:all)

End
```

Step 5 : The **welcome.rhtml file** can be as follows -

```
<html>
<head><title> RAILS-MYSQL DEMO </title></head>
<body>
<form method="post" action="show">
<input type="text" name="title" size="20"/>
<input button="submit" name="Display" />
</form>
</body>
</html>
```

The **show.rhtml** file can as follows -

```
<html>
  <head>
    <title> Book Display</title>
   </head>
   <body>
   <table border="1">
     <tr>
      <td><b>AccessNo</b></td>
   <td><b>Title</b></td>
   <td><b>Author</b></td>
   <td><b>Edition</b></td>
   <td><b>Publication</b></td>
   <td><b>Subject</b></td>

     </tr>
       <% @bookarr.each do |b|
      @accno=b.accNo
      @title=b.tilte
       @author=b.author
      @edition=b.edition
      @pub=b.publication
      @subject_row=@subarr[b.title_id]
      @subjectname=@subject_row. subject_name %>
     <tr>
      <td><b><%= accno %></b></td>
      <td><b><%= title %></b></td>
      <td><b><%= author %></b></td>
```

```
        <td><b><%= edition %></b></td>
        <td><b><%= pub %></b></td>
        <td><b><%= subjectname %></b></td>
      </tr>
      <% end %>
    </table>
    </body>
<html>
```

6.15 : Layouts

Q.18 Introduce the concept of Rails application. Describe layouts and stylesheet in Rail. ☞ **[SPPU : May-22, Marks 8]**

Ans. : Rails Application : Refer Q.17.

Layouts in Rail :

- In rails, the **app/View** directory of each application has two directories, one is for the controller and another is named as **layouts.**
- A layout defines the surroundings of an HTML page. In layout directory you can place a file for common look and feel of your final output. For that purpose layout template is created.
- Rails must be informed about the name of the layout document by placing the **layout** command in the **ApplicationController** class of the application.
- In the **controller** directory of **app** there is a file named **application_controller.rb** in this file the following code can be placed -

```
class ApplicationController < ActionController::Base
 layout "mainlayout"
end
```

- Now if we add **mainlayout.rhtml** document to layout directory then the basic layout of the application can be defined.

6.16 : Rails with Ajax

Q.19 Write short note on - Rails with Ajax.

Ans. : • JavaScript can also make requests to the server, and parse the response. It also has the ability to **update information** on the page. Combining these two powers, a JavaScript writer can make a web page that can **update just parts of itself**, without needing to get the full page data from the server. This is a powerful technique is called as **Ajax**.

• For example -

```
fetch("/index")
 .then((data) => data.text())
 .then((html) => {
  const results = document.querySelector("#compute");
  results.insertAdjacentHTML("before_exit", data);
 });
```

• This code fetches data from "/index" and then appends the result to the element with an id of **compute.**

• Rails provides 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 the DOM. This is a technique by which we can mix JavaScript code into HTML. With this, we can easily add behavior to any link by adding data attribute.

6.17 : Introduction to EJB

Q.20 What is EJB ? Explain the two types of EJB.

Ans. : • J2EE application container contains the components that can be used by the clients for executing the business logic. These components are known as Enterprise Java Beans (EJB).

• EJB mainly contains the business logic and business data.

• Using the technology like EJB the server side, distributed Java components can be built.

- EJBs provide a core set of system services to Java component. **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 and this class implements business logic.

Entity Bean

- Entity beans are used for modelling the business concepts. Therefore entity beans represent the data. For example entity bean can represent the bank account or a customer. The **nouns** that are used in the problem description refer to the entity beans. Basically the entity beans represent the **real world objects**.
- Such beans then act as a **persistent record** in the database.

Session Bean

- Session beans are used for managing processes or tasks. For example *Compute net profit* or *transaction* or *browsing for some information on the web browser*. Hence session beans are used for managing activities. Basically session beans represent the **activities**. Hence session beans are **transient**, because activities generally exist for a short period. Session bean does not represent anything in the database but it can access the database. Session beans are client specific.

Q.21 Give the difference between entity bean and session bean.

Ans. :

Sr. No.	Entity bean	Session bean
1.	These are **persistent.**	These are **transient.**
2.	It refers to the **records** in the database.	These refers to the **activities** for the database.
3.	These can **survive** on server crash. (After server gets online these can be reconstructed from the underlying data).	Session objects generally **cannot survive** on server crashes.

4.	Entity beans are **client-independent** entities.	Session beans are **client-specific** entities.

Q.22 Draw and explain each tier of three tiers architecture using EJB.

[SPPU : May-19, Marks 8]

Ans. : Components of EJB

- The EJB container provides following services to EJB component -
  - Security
  - Transactions
  - Messaging
  - Persistence
  - Distributions
  - Connectivity
- The idea behind developing EJB is to provide a framework for the components that can be plugged into a server. This ultimately helps to increase the functionality of the server.
- EJB is a managed component that can be created, controlled and managed by its container J2EE.
- There can be multiple EJBs in the container J2EE. The container J2EE provides the resources (such as memory and database connections) required by the EJB.
- Thus J2EE maintains a pool of EJB instances. These instances can be assigned to the clients.
- Client makes use of these EJB components and when it is no longer required by the client, these instances can be returned to the pool so that other clients can used these instances. The client who is using the EJB instance does not need to know about its container.

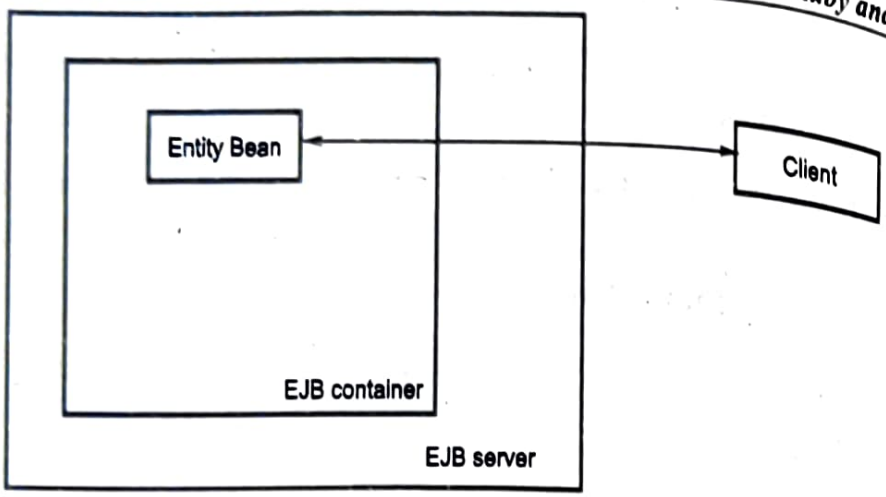


Fig. Q.22.1

- EJB has a technique that keeps business logic separate from the database. In other words, the business logic can deal with business data without caring about how data is stored in database.
- There are two main purposes of the container firstly control the EJB it contains and secondly provide the services to the EJB that are currently present in the container.
- The EJBs are typically one part of a three tier architecture as shown below.

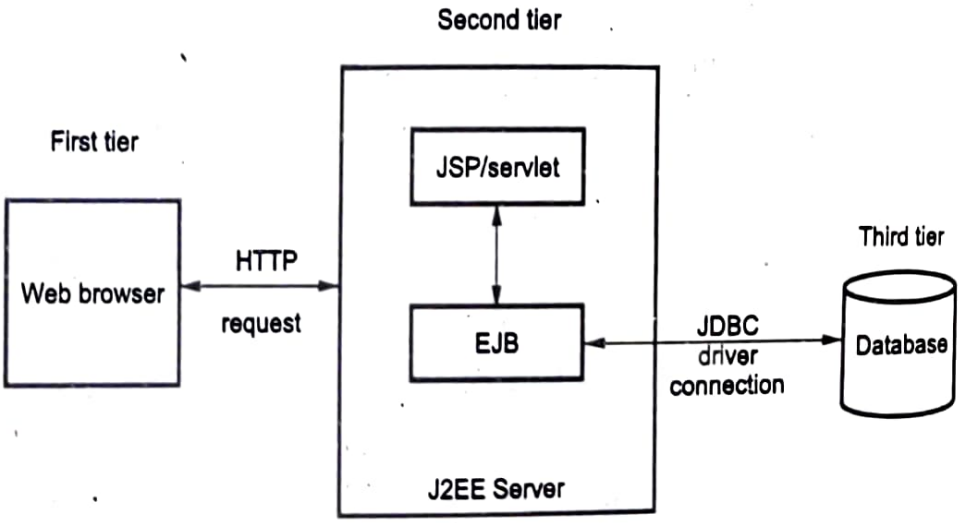


Fig. Q.22.2 3 Tier architecture

- **The following are the flows of the EJB architecture.**
  1. The client is working on a web browser.
  2. There is a database server that hosts a database, like MySQL / Oracle.

3. The J2EE server machine is running on an application server.
4. The client interface is provided with JSP / Servlet. The enterprise beans reside in the business tier providing to the client tier.
5. The Application Server manages the relationships between the client and database machines.

Q.23 Explain architecture of EJB and explain types of EJB in detail.

☞ **[SPPU : May-22, Marks 9]**

OR What is enterprise Java Bean ? Draw and explain main components of EJB architecture. ☞ **[SPPU : May-18, Marks 8]**

OR Draw and explain the role of EJB container in enterprise applications. ☞ **[SPPU : May-18, Marks 6]**

Ans. : Architecture : Refer Q.22.

Types : Refer Q.20.

Q.24 Decide and explain various aspects while deciding between local and remote interface can be considered. ☞ **[SPPU : May-18, Marks 8]**

Ans. : In EJB, the client can access the session bean using the business interface. Depending upon the location of client, there are two types of interfaces used -

1) **Remote Interface :** The remote interface is used by client that may be running on different machine than the bean. Using Remote Method Interface (RMI) the access to the bean can be made.
2) **Local Interface :** The local interface can be used by the client which is running on the same machine where the bean is residing.

Various aspects while deciding between local and remote interface are represented in the following table -

Aspect	Use of Remote Interface	Use of Local Interface
Type of client	If the clients are application clients, web components or EJBs executing on different machine.	If the clients are either web components or EJBs executing on the same machine.

Protocol used	If the RMI used for communication.	If the local function calls are used for communication.
Component distribution	If the component in the application are distributed, that means present on different machines.	If the components are located on the same machine where bean is present.
Loose or tight coupling	If the client and bean are coupled loosely.	If there is a requirement of tight coupling between EJB components.
Performance	Slow performance due to RMI calls.	Better performance due to local calls.
Location transparency	If the client need not know the location of EJB and access it transparently.	If the client wants to know that the current machine is the location of EJB.
Communication	If multiple objects are passed by value so that client and bean can operate on different copies of the object.	If objects are passed by reference and the client or bean can change the value of the same object.
Serialization	If the serialized objects are passed.	If the objects are not serialized.

Q.25 Draw and explain scenario of client accessing remote EJB. List some of the EJB clients. **[SPPU : Dec.-18, May-19, Marks 8]**

Ans. : Scenario of client accessing remote EJB : Refer Q.22.

EJB clients :

1) Java Applications
2) Other EJBs on same or another application server.
3) Java Applets
4) A web service can access EJB
5) Servlets, JSP- some server side technologies.

END...

May-2022 [END SEM] [5870] - 1134	Solved Paper

Course 2019

Time : $2\frac{1}{2}$ Hours] [Maximum Marks : 70

Instructions to the candidates :

1) *Solve Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.*

2) *Neat diagrams must be drawn wherever necessary.*

3) *Figures to the right side indicate full marks.*

Q.1 **a)** *What is difference between server side scripting language and client side scripting language.* **(Refer Q.1 of Chapter - 3)** [5]

b) *Describe servlet architecture in detail.* **(Refer Q.2 of Chapter - 3)** [4]

c) *Explain DTD in XML with schemes, elements and attributes.* **(Refer Q.39 of Chapter - 3)** [9]

OR

Q.2 **a)** *What is session ? How cookies and URL rewriting for session management in servlet.* **(Refer Q.16 of Chapter - 3)** [9]

b) *Write short note on :*

i) AJAX **(Refer Q.41 of Chapter - 3)**

ii) XML transformation. **(Refer Q.33 of Chapter - 3)** [9]

Q.3 **a)** *Explain life cycle of JSP. Write advantages of JSP over servlet.* **(Refer Q.2 of Chapter - 4)** [9]

b) *Write a short notes on :*

i) SOAP **(Refer Q.17 of Chapter - 4)**

ii) WSDL. **(Refer Q.16 of Chapter - 4)** [8]

OR

Q.4 **a)** *Explain struts framework with its components. Also explain interceptors.* **(Refer Q.21 of Chapter - 4)** [9]

b) *Explain JSP support for MVC i.e. model, view controller for developing web application.* **(Refer Q.10 of Chapter - 4)** [8]

Q.5 a) *Identify and explain steps involved in connecting to mySQL with PHP.* **(Refer Q.26 of Chapter - 5)** [9]

b) *Write short notes on :*

i) Overview of ASP. NET **(Refer Q.30 of Chapter - 5)**

ii) Overview of C#. **(Refer Q.33 of Chapter - 5)** [9]

OR

Q.6 a) *Explain in detail WAP Architecture and WML.* **(Refer Q.29 of Chapter - 5)** [9]

b) *Explain functions in PHP with example and session management.* **(Refer Q.24 of Chapter - 5)** [9]

Q.7 a) *Explain classes and objects in Ruby with appropriate examples.* **(Refer Q.10 of Chapter - 6)** [9]

b) *Introduce the concept of Rails application. Describe layouts and stylesheet in Rail.* **(Refer Q.18 of Chapter - 6)** [8]

OR

Q.8 a) *Explain the scalar types and their operations in Ruby.* **(Refer Q.2 of Chapter - 6)** [8]

b) *Explain architecture of EJB and explain types of EJB in detail.* **(Refer Q.23 of Chapter - 6)** [9]

END ...