op code: 9810

1

a. Differentiate between GET and POST. (5)

Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to server.

GET	POST
1) In case of Get request, only **limited amount of data** can be sent because data is sent in header.	In case of post request, **large amount of data** can be sent because data is sent in body.
2) Get request is **not secured** because data is exposed in URL bar.	Post request is **secured** because data is not exposed in URL bar.
3) Get request **can be bookmarked.**	Post request **cannot be bookmarked.**
4) Get request is **idempotent** . It means second request will be ignored until response of first request is delivered	Post request is **non-idempotent.**
5) Get request is **more efficient** and used more than Post.	Post request is **less efficient** and used less than get.

b. Explain web services. (5)

A **Web Service** is can be defined by following ways:

- Is a client server application or application component for communication
- Method of communication between two devices over network.
- Is a software system for interoperable machine to machine communication
- Is a collection of standards or protocols for exchanging information between two devices or application

There are three major web service components.

- SOAP
- WSDL
- UDDI

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of −

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

Example

Consider a simple account-management and order processing system. The accounting personnel use a client application built with Visual Basic or JSP to create new accounts and enter new customer orders.

The processing logic for this system is written in Java and resides on a Solaris machine, which also interacts with a database to store information.

The steps to perform this operation are as follows −

- The client program bundles the account registration information into a SOAP message.
- This SOAP message is sent to the web service as the body of an HTTP POST request.
- The web service unpacks the SOAP request and converts it into a command that the application can understand.
- The application processes the information as required and responds with a new unique account number for that customer.
- Next, the web service packages the response into another SOAP message, which it sends back to the client program in response to its HTTP request.
- The client program unpacks the SOAP message to obtain the results of the account registration process.

c. Explain 3 tier architecture of web application. (5)

- In 3 tier architecture, there are 3 components: Client PC, An Application server and A Database Server.
- The work of server is distributed among application server and database server.
- Application server has the required communication functions.
- The data required by the business logic exists in database server.
- The required data is returned to public servers and then to client PC.

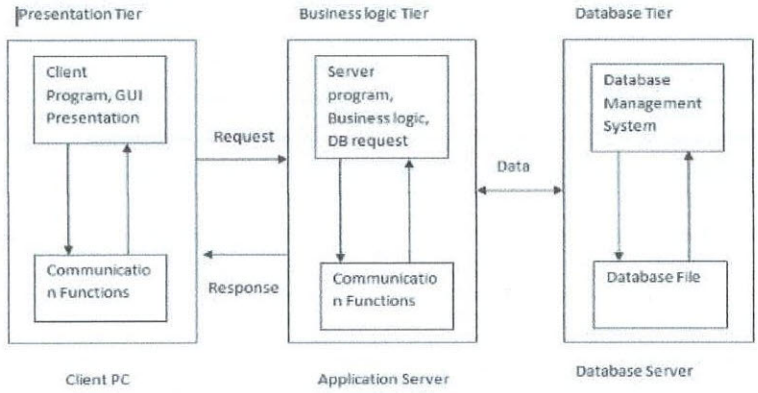


Figure 1.1: 3-tier architecture

Advantages:

- Improved Data Integrity
- High Degree of Flexibility in deployment platform and configurations
- Improved security
- High Performance and persistent objects
- Architecture is scalable, adding users and resources in future would be easy
- Maintenance and modifications can be done effectively
- Code and data reusability can be achieved

Disadvantages:

- 3 tier architecture is complex compared to 1 tier and 2 tier
- Cost of network maintenance and deployment is greater than 1 tier and 2 tier

d. Differentiate between HTML and XML. (5)

HTML	XML
HTML is an abbreviation for HyperText Markup Language.	XML stands for eXtensible Markup Language.
HTML was designed to display data with focus on how data looks.	XML was designed to be a software and hardware independent tool used to transport and store data, with focus on what data is.
HTML is a markup language itself.	XML provides a framework for defining markup languages.
HTML is a presentation language.	XML is neither a programming language nor a presentation language.
HTML is case insensitive.	XML is case sensitive.
HTML is used for designing a web-page to be rendered on the client side.	XML is used basically to transport data between the application and the database.
HTML has it own predefined tags.	While what makes XML flexible is that custom tags can be defined and the tags are invented by the author of the XML document.
HTML is not strict if the user does not use the closing tags.	XML makes it mandatory for the user the close each tag that has been used.
HTML does not preserve white space.	XML preserves white space.
HTML is about displaying data,hence static.	XML is about carrying information,hence dynamic.

2	a. Write a JavaScript program to validate a form which accepts name, age, email and phone no. of a student `<html>` `<head>` `<title> FormData</title>` `<script>` `function myFunction()` `{` `message("The form was submitted");` `}` `</script>` `</head>` `<body>` `<h1> Enter Student's information</h1>` `<form onsubmit="myFunction()">` `Name: <input type="text" name="studentname" required>` ` ` `Age: <input type="text" name="Age" pattern="[0-9]{2}" title="Please enter valid age" required>` ` ` `Email id: <input type="text" name="Emailid" pattern="^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$" title="Please enter valid email id" required>` ` ` `Phone Number: <input type="number" name="PhoneNo" pattern="[0-9]{10}" title="Please enter valid phone number" required>` ` ` `<input type="submit" value="Submit">` `</form>` `</body>` `</html>` b. Write CSS for an HTML page to: • Set the background color • Set the size of the font • Create a hyperlink without an underline `<!DOCTYPE html>` `<html>` `<head>` `<style>` `body {` `background-color: coral;` `}` `h1 {` `font-size: 250%;` `}` `h2 {` `font-size: 200%;` `}`	(10) (10)

```
p {
   font-size: 100%;
}
a:link {
   text-decoration: none;
}
a:visited {
   text-decoration: none;
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<a href="default.asp" target="_blank">This is a link</a>
</body>
</html>
```

3 a. Write HTML code to draw following table. (10)

	Average		Red Eyes
	Height	**Weight**	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

```
<table>
 <tr>
  <th rowspan="2"></th>
  <th colspan="2">Average</th>
  <th rowspan="2">Red <br>Eyes</th>
 </tr>
 <tr>
  <td>Height</td>
  <td>Weight</td>
 </tr>
 <tr>
  <td>males</td>
  <td>1.9</td>
  <td>0.003</td>
  <td>40%</td>
 </tr>
 <tr>
  <td>females</td>
  <td>1.7</td>
  <td>0.002</td>
  <td>43%</td>
 </tr>
</table>
```

b. Write an ASP.NET program to insert a new record in database. (10)

Default.aspx

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:GridView ID="GridView1" runat="server"
OnRowCommand="gridview1_RowCommand" >
</asp:GridView>
 <br />
<asp:Button ID="Button2" runat="server" Text="Insert" />
  
<br />

Roll_No :            

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<br />
<br />
    Student_Name  :    
<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
<br />
<br />
    Phone_Number :    
<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
<br />

<br />

Adderss :

<asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
<br />
<br />

    Comparing value :  
<asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
<br />
<br />
</div>
</form>
</body>
</html>
```

Default.aspx.vb

```
Imports System.Data
Imports System.Data.SqlClient
Partial Class _Default
   Inherits System.Web.UI.Page
   Dim mycom As SqlCommand
   Dim mycon As SqlConnection
   Dim ds As DataSet
   Dim adp As SqlDataAdapter
```

```
   Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Button1.Click
mycon = New SqlConnection("server=webserver1;user id =sa;password=sa;database=VUW")
mycon.Open()
mycom = New SqlCommand("select * from t1", mycon)
ds = New DataSet
adp = New SqlDataAdapter(mycom)
adp.Fill(ds, "t1")
      GridView1.DataSource = ds
GridView1.DataBind()
mycon.Close()
   End Sub
   Protected Sub Button2_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Button2.Click
mycon = New SqlConnection("server=webserver1;user id =sa;password=sa;database=VUW")
mycon.Open()
mycom = New SqlCommand("insert into t1 (Roll_No,Student_Name,Phone_No,Address) values ('" &
TextBox1.Text & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "')",
mycon)
mycom.ExecuteNonQuery()
mycon.Close()
   End Sub

   Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewCommandEventArgs) Handles GridView1.RowCommand
      If e.CommandName = "Select" Then
         Dim index As Integer = Convert.ToInt32(e.CommandArgument)
         Dim r As GridViewRow = GridView1.Rows(index)
Session.Add("Roll_No", r.Cells(1).Text)
Session.Add("Student_Name", r.Cells(2).Text)
Session.Add("Phone_No", r.Cells(3).Text)
Session.Add("Address", r.Cells(4).Text)
      End If
      TextBox1.Text = Session("Roll_No")
      TextBox2.Text = Session("Student_Name")
      TextBox3.Text = Session("Phone_No")
      TextBox4.Text = Session("Address")
   End Sub
End Class
```

4 a. What is jQuery? Illustrate the use of jQuery for form validation. (10)

- JQuery is a light, "write less, do more", Javascript library.
- The purpose of JQuery is to make it much easier to use JavaScript on your website. It simplifies HTML document traversing, event handling, animating and AJAX interactions for rapid web development.
- The jQuery library is a single JavaScript file, and you reference it with the HTML

<script> tag (notice that the <script> tag should be inside the <head> section):

```
<head>
<script src="jquery-1.11.3.min.js"></script>
</head>
```

jQuery Syntax

- The jQuery syntax is tailor made for selecting HTML elements and performing some action on the element(s).
- Basic syntax is: $(selector).action()

1. A $ sign to define/access jQuery
2. A (selector) to "query (or find)" HTML elements
3. A jQuery action() to be performed on the element(s)
   Examples:
4. $(this).hide() - hides the current element.
5. $("p").hide() - hides all
   elements.
6. $(".test").hide() - hides all elements with class="test".
7. $("#test").hide() - hides the element with id="test".

Query Selectors

- jQuery selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: $().

The element Selector

- The jQuery element selector selects elements based on the element name.
- You can select all elements on a page like this: $("p")
- example

JQuery for Form Validation:

```
<html>
<head>
 <script src="jquery.js"></script>
  <script src="jquery_validate.js"></script>
  <script>
    $(document).ready(function(){
    $("#registerform").validate({
       rules: {
                              firstname: "required",
                              lastname: "required",
                              email:
                              {       required: true,
                                      email: true
                              },
                              contact:
                              {       required:true,
                                      number:true,
                                      minlength:10,
                                      maxlength:10
                              },
                              birth:
                              {       required:true,
                                      date:true
                              },
                              username:
                              {       required:true,
                                       maxlength:6
                              },
                              password:
                              {       required: true,
                                      minlength: 5,
                                      maxlength:10
                               },
                              gender:"required",
                              hobby:"required",
```

(10)

```
                        country:"required"
                        },
        // Specify the validation error messages
    messages:
                {
      firstname: "Please enter ur firstname",
      lastname: "Please enter ur lastname",
      username:
                        {       required:"Enter your username",
                                minlength:"Your username must be at least 6 characters long"
                        },
       password:
                        {  required:"Enter your password",
                        minlength: "Your password must be at least 5 characters long"
                        },
       contact:
                        {       number:"please enter numbers",
                                minlength:"Please enter valid contact no."
                        },
       birth:
                        {date:"Please enter valid date"
                        },
       email: "Please enter a valid email address"
                        }
       })
 })
 </script> </head><body>
 <h1>Register here</h1>
 <form id="registerform" method="post" >
<table>
<tr><td>Firstname</td><td><input type=text id="firstname" name="firstname"></td></tr>
<tr><td>Lastname</td><td><input type=text id="lastname" name="lastname"></td></tr>
<tr><td>Email</td><td><input type=text id="email" name="email"></td></tr>
<tr><td>Contact No.</td><td><input type=text id="contact" name="contact"></td></tr>
<tr><td>Date of Birth</td><td><input type=text id="birth" name="birth"></td></tr>
<tr><td>Username</td><td><input type=text id="username" name="username"></td></tr>
<tr><td>Password</td><td><input type=password id="password" name="password"></td></tr>
<tr><td>Gender</td><td>Male<input type=radio name="gender" id="male"></td></tr>
<tr><td></td><td>Female<input type=radio name="gender" id="female"</td></tr>
<tr><td>Hobby</td><td>Reading<input type=checkbox name="hobby" id="hobby"</td>
<td>Dancing<input type=checkbox name="hobby" id="hobby"</td></tr>
<tr><td></td><td>Painting<input type=checkbox name="hobby" id="hobby"</td>
<td>Singing<input type=checkbox name="hobby" id="hobby"</td></tr>
<tr><td>Country</td><td><select name="country" id="country"><option disabled=true
selected=selected>None
<option>USA <option>India
<option>Australia</select></td></tr>
<tr><td><input type=submit id="submit" value=Submit name=submit></td>
<td><input type=reset id="reset" value=Reset name=reset></td></tr>
</table>
</form>
</body>
</html>
```

b. Write a JSP program to read the students data like roll number, name, email and marks from the database and display in tabular format on web page.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Student details </title>
</head>
<body>

</body>
<form method="post">
<table border="2">
<tr>
<td>ROLL NUMBER</td>
<td>NAME</td>
<td>EMAIL</td>
<td>MARKS</td>
</tr>

<%

try
{
Class.forName("com.mysql.jdbc.Driver");
String url="jdbc:mysql://localhost/test";
String username="root";
String password="root";
String query="select * from jsp1";
Connection conn=DriverManager.getConnection(url,username,password);
Statement stmt=conn.createStatement();
ResultSet rs=stmt.executeQuery(query);
while(rs.next())
{
%>
<tr>
<td><%=rs.getInt("ROLL NUMBER ") %></td>
<td><%=rs.getString("NAME") %></td>
<td><%=rs.getString("EMAIL") %></td>
<td><%=rs.getString("MARKS") %></td>
</tr>
<%
}
%>
</table>
<%
rs.close();
stmt.close();
conn.close();
   }

catch(Exception e)
   {
e.printStackTrace();
   }
```

```
%>
</form>
</html>
```

5 a. What is session? What are the ways to do session tracking? (10)

- When a user logs in to a application, performs some activity and close the application, this entire scenario is called as session.
- A web server needs to track all the activities of users. It needs to store required information about the user. This whole process is called as session tracking.
- There exists a session array which often stores the unique session ID for the session.
- PHP keep tracks of session by using a function called session_start().
- When session_start() function is invoked , session ID is created and recorded.

Following is a PHP code:

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

- Another way to store information is using Cookies. Cookies is a small file that server embeds in a user's machine.
- Cookies are used to identify users. It consists of a name and textual value. It can be created by some software system on the server.
- In every HTTP communication between browser and server, a header is included.
- The header part contains the information stored in cookies, about the message. -- In PHP, a function setcookie() can be used to create cookie.

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

- An information from the cookie can be retrieved in a following way:

```
<?php
if(isset($_COOKIE["Myname"])    echo "Welcome" .$_COOKIE["Myname"]."....!!";?>
```

b. Write a program to accept username and password fields from user in form; store and display it using PHP-MySQL (10)

Registration FORM

```
<div class="container">
  <form class="form-signin" method="POST">
    <h2 class="form-signin-heading">Please Register</h2>
    <div class="input-group">
```

```
  <span class="input-group-addon" id="basic-addon1">@</span>
  <input type="text" name="username" class="form-control" placeholder="Username" required>
</div>
      <label for="inputPassword" class="sr-only">Password</label>
      <input type="password" name="password" id="inputPassword" class="form-
control" placeholder="Password" required>
      <button type="submit">Register</button>
      <a href="login.php">Login</a>
    </form>
</div>
```

Connection

```
<?php
$connection = mysqli_connect('localhost', 'root', 'Rvm@i[9)0?~=');
if (!$connection){
   die("Database Connection Failed" . mysqli_error($connection));
}
$select_db = mysqli_select_db($connection, 'test');
if (!$select_db){
   die("Database Selection Failed" . mysqli_error($connection));
}
```

PHP Registration

```
<?php
require('connect.php');
  // If the values are posted, insert them into the database.
  if (isset($_POST['username']) && isset($_POST['password'])){
     $username = $_POST['username'];
$email = $_POST['email'];
     $password = $_POST['password'];
     $query = "INSERT INTO `user` (username, password, email) VALUES ('$username', '$password',
'$email')";
     $result = mysqli_query($connection, $query);
     if($result){
        $smsg = "User Created Successfully.";
     }else{
        $fmsg ="User Registration Failed";
     }
  }
  ?>
```

6 a. Write an XML to accept student details (Name, ID, Branch, and CGPA). Write an XSL to display the list of students in descending order of their CGPA. (10)

xml code:

```
<?xml version="1.0"?>
<class>
<student>
<name> ABC </name>
<id> 001 </id>
<branch> IT </branch>
<cgpa> 9 </cgpa>
</student>
<student>
```

<name> PQR </name>
<id> 004 </id>
<branch> IT </branch>
<cgpa> 8 </cgpa>
</student>
<student>
<name> XYZ </name>
<id> 006 </id>
<branch> Computer </branch>
<cgpa> 10 </cgpa>
</student>
</class>

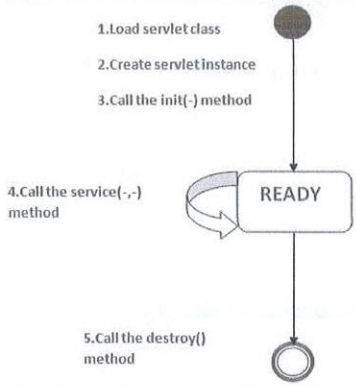
xsl code:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
 <h2>Student list</h2>
 <table border="1">
  <tr bgcolor="#9acd32">
   <th>ID</th>
   <th>Name</th>
   <th>Branch</th>
   <th>CGPA</th>
  </tr>
  <xsl:for-each select="class/student">
   <xsl:sort select="cgpa"/ order=”descending” data-type=”number”>
   <tr>
    <td><xsl:value-of select="id"/></td>
    <td><xsl:value-of select="name"/></td>
    <td><xsl:value-of select="branch"/></td>
    <td><xsl:value-of select="cgpa"/></td>
   </tr>
  </xsl:for-each>
 </table>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

(5)

b. Explain Servlet life cycle.

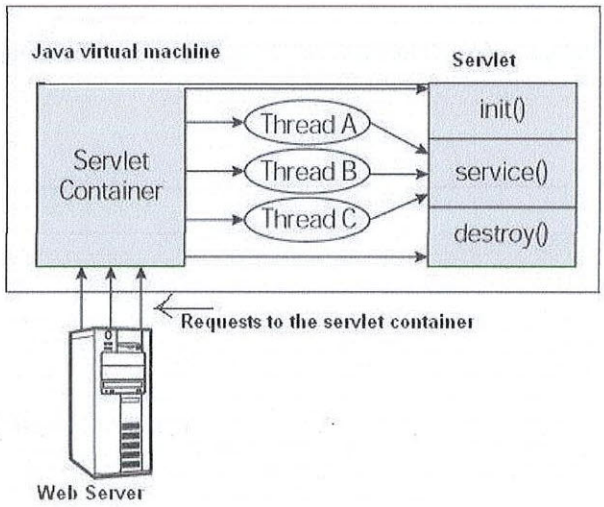
(5)



A servlet life cycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servlet.

- The servlet is initialized by calling the init() method.
- The servlet calls service() method to process a client's request.
- The servlet is terminated by calling the destroy() method.
- Finally, servlet is garbage collected by the garbage collector of the JVM.

Describe each step in 1-2 lines.



c. Write JavaScript program to change background color continuously.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Background effect by using loop </title>
<script language="javascript">
var COLOR = 999999
var woot = 0
function stoploop()
{
document.bgColor = '#fff';
clearTimeout(loopID);
}
function loopBackground()
 {
if (COLOR > 0)
{
```

```
document.bgColor = '#' + COLOR
COLOR -= 111111
loopID = setTimeout("loopBackground()",1)
}
else
{
document.bgColor = '#000000'
woot += 10
COLOR = 999999
COLOR -= woot
loopID = setTimeout("loopBackground()",1)
}
}
</script>

</head>
<body style="width:800px; margin:0 auto">
<center>
<h3> continuously change the background color by using loop </h3>
<FORM NAME="background">
<INPUT TYPE="button" VALUE="Start bgColor WARP"
onClick="loopBackground()">
<br>
<input type="button" value="Stop bgColor WARP" onClick="stoploop()">
</FORM>
</center>
</body>
</html>
```