1a. Write a Java program to implement linear search.

```
import java.util.Scanner;
Class LinearS
{
Public static void main(String args[])
{
int counter, num, item, array[];
Scanner input =new Scanner(System.in);
System.out.println("Enter number of elements");
num =input.nextInt();
array = new int[num];
System.out.println("Enter "+ num +" integers");
for(counter =0; counter < n; counter++)
array[counter]=input.nextInt();
System.out.println("Enter the search value");
item=input.nextInt();
for(counter =0; counter < n; counter++)
{
if(array[counter]== item)
{
System.out.println(item +" is present at location "+(counter+1));
break;
}
}
if(counter == num)
System.out.println(item +" is not present in array.");
}
}
```

**********OUTPUT**********

```
Enter the number of elements 4
Enter 4 integer 20 52 17 92
Enter search value 17
17 is present at location 3
```

1b. Write a java program for sorting a given list of names.

```
import java.io.*;
class SortingNames
{
public static void main(String[] args) throws IOException
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.print("\nEnter The number of Names :");
int n = Integer.parseInt(br.readLine());
String names[] = new String[n];
System.out.println();
for (int i = 1; i <= n; i++)
{
System.out.print("Enter Name " + i + ":");
names[i-1] = br.readLine();
}
System.out.println("\nNames in Ascending Order");
System.out.println();
for (int j = 0; j < names.length; j++)
{
for (int i = j + 1; i < names.length; i++)
{
if (names[i].compareToIgnoreCase(names[j]) < 0)
{
String temp = names[j];
names[j] = names[i];
names[i] = temp;
}
}
System.out.println(names[j]);
}
}
}
```

**********OUTPUT**********

```
Output:
Enter the number of names:3
Enter name 1: ab
Enter name 2: abc
Enter name 3:a
Names in Ascending Order
a
ab
abc
```

2a. Write a java program that illustrates the multilevel inheritance.

```
class Car{
public Car()
{
System.out.println("Class Car");
}
public void vehicleType()
{
System.out.println("Vehicle Type: Car");
}
}
class Maruti extends Car{
public Maruti()
{
System.out.println("Class Maruti");
}
public void brand()
{
System.out.println("Brand: Maruti");
}
public void speed()
{
System.out.println("Max: 90Kmph");
}
}
public class Maruti800 extends Maruti{

public Maruti800()
{
System.out.println("Maruti Model: 800");
}
public void speed()
{
System.out.println("Max: 80Kmph");
}
public static void main(String args[])
{
Maruti800 obj=new Maruti800();
obj.vehicleType();
obj.brand();
obj.speed();
```

```
}
}
```

**********OUTPUT**********

```
Class Car
Class Maruti
Maruti Model: 800
Vehicle Type: Car
Brand: Maruti
Max: 80Kmph
```

2b.To write a program to implement multiple inheritance using interfaces.

```
import java.io.*;
class student
{
int rno;
void getno(int n)
{
rno = n;
}
void putno()
{
System.out.println("RegNo:"+rno);
}
}
class Test extends student
{
float m1,m2;
void getmarks(float a,float b)
{
m1=a;
m2=b;
}
void putmarks()
{
System.out.println("M1 :"+m1);
System.out.println("M2 :"+m2);
}}
interface sports
{
float sportwt =6.0f;
void putwt();
}
class Results extends Test implements sports
{
float tot;
public void putwt()
{
```

```
System.out.println("Sports wt :"+sportwt);
}
void display()
{
tot = m1+ m2 + sportwt;
putno();
putmarks();
putwt();
System.out.println("Total: " +tot);
}}
class clsmultiple
{public static void main(String args[])
{
Results r = new Results();
r.getno(1001);
r.getmarks(79f,95f);
r.display();
}
}
```

**********OUTPUT**********

```
RegNo:1001
M1 :79.0
M2 :95.0
Sports wt :6.0
Total: 180.0
D:\JAVA>
```

40

3a.Write a Java program to implement the concept of importing classes from user defined package and creating packages.

```
/*Source code of package p1 under the directory C:\jdk1.6.0_26\bin>p1\edit Student.java */
package p1;
public class Student
{
int regno;
String name;
public void getdata(int r,String s)
{
regno=r;
name=s;
}
public void putdata()
{
System.out.println("regno = " +regno);
System.out.println("name = " + name);
}
}

/* Source code of the main function under C:\jdk1.6.0_26\bin>edit StudentTest.java */
import p1.*;
class StudentTest
{
public static void main(String arg[])
{
student s=new student();
s.getdata(123,"xyz");
s.putdata();
}
}
```

**********OUTPUT**********

```
regno = 123
name = xyz
```

3b. Write a Java program that reads a line of integers and then displays each integer and the sum of all integers. (use StringTokenizer class)

```
import java.util.*;
class StringTokenizerDemo {
public static void main(String args[])
{
int n;
int sum = 0;
Scanner sc = new Scanner(System.in);
System.out.println("Enter integers with one space gap:");
String s = sc.nextLine();
StringTokenizer st = new StringTokenizer(s, " ");
while (st.hasMoreTokens()) {
String temp = st.nextToken();
n = Integer.parseInt(temp);
System.out.println(n);
sum = sum + n;
}
System.out.println("sum of the integers is: " + sum);
sc.close();
}
}
```

**********OUTPUT**********

```
Enter integers with one space gap:
10 20 30 40 50
10
20
30
40
50
sum of the integers is: 150
```

4a.To write a program to perform arithmetic operations using static members.

```
import java.util.Scanner;
class stclass
{
static int num1, num2;
static int add( int a, int b)
{
return a+b;
}
static int sub( int a, int b)
{
return a-b;
}
static int mul( int a, int b)
{
return a*b;
}
static int div( int a, int b)
{
return a/b;
}
static int modulus( int a, int b)
{
return a%b;
}
static int increment( int a)
{
return ++a;
}
static int decrement( int a)
{
return --a;
}

public static void main(String args[])
{
Scanner input = new Scanner(System.in);
System.out.println("Enter two numbers:");
num1  =  input.nextInt();
num2  =  input.nextInt();
System.out.println("\nThe two numbers are: " + num1 + "," + num2 +"\nAdditon: " + add(num1, num2) + "\nSubstraction: " + sub(num1, num2)+"\nMultiplication: " +mul(num1, num2) + "\nDivision: " + div(num1, num2) +"\nModulus: " + modulus(num1, num2) +"\n Increment of 1st num: "+ increment(num1) +"\nDecrement of 2nd num" + decrement(num2));
```

```
}
}
```

**********OUTPUT**********

```
Enter two numbers:
40
4
The numbers are : 40,4
Addition: 44
Substraction: 36
Multiplication: 160
Division: 10
Modulus: 0
Incremented 1st num: 41
Decremented 2nd num: 3
```

4b.To write a program to read and print n numbers using arrays.

```
import java.io.*;
class clsarray1
{
public static void main(String args[])throws IOException
{
BufferedReader br= new BufferedReader(new
InputStreamReader(System.in));
int i,n;
int a[] = new int[100];
System.out.println("Enter the value of n");
n=Integer.parseInt(br.readLine());
System.out.println("Enter the values");
for(i=0;i<n;i++)
{
a[i]=Integer.parseInt(br.readLine());
}
System.out.println("The values are");
for(i=0;i<n;i++)
{
System.out.println(a[i]);
}
}
}
```

**********OUTPUT**********

```
Enter the value of n
3
Enter the values
1
2
3
The values are
1
2
3
30
```

5a. Write a Java program to practice using String class and its methods.

```
import java.lang.String;
class stringdemo
{
public static void main(String arg[])
{
String s1=new String("gpt gulbarga");
String s2="GPT GULBARGA";
System.out.println(" The string s1 is : " +s1);
System.out.println(" The string s1 is : " +s2);
System.out.println(" Length of the string s1 is : " +s1.length());
System.out.println(" The first accurence of r is at the position : " +s1.indexOf('r'));
System.out.println(" The String in Upper Case : " +s1.toUpperCase());
System.out.println(" The String in Lower Case : " +s1.toLowerCase());
System.out.println(" s1 equals to s2 : " +s1.equals(s2));
System.out.println(" s1 equals ignore case to s2 : " +s1.equalsIgnoreCase(s2));
int result=s1.compareTo(s2);
System.out.println("After compareTo()");
if(result==0)
System.out.println( s1 + " is equal to "+s2);
else if(result>0)
System.out.println( s1 + " is greather than to "+s2);
else
System.out.println( s1 + " is smaller than to "+s2);
System.out.println(" Character at an index of 6 is :" +s1.charAt(6));
String s3=s1.substring(4,12);
System.out.println(" Extracted substring is :"+s3);
System.out.println(" After Replacing g with a in s1 : " + s1.replace('g','a'));
String s4=" This is a book ";
System.out.println(" The string s4 is :"+s4);
System.out.println(" After trim() :"+s4.trim());
}
}
```

**********OUTPUT**********

```
The string s1 is : gpt gulbarga
The string s1 is : GPT GULBARGA
Length of the string s1 is : 12
The first accurence of r is at the position : 9
The String in Upper Case : GPT GULBARGA

The String in Lower Case : gpt gulbarga
s1 equals to s2 : false
s1 equals ignore case to s2 : true
After compareTo()
gpt gulbarga is greather than to GPT GULBARGA
Character at an index of 6 is :l
Extracted substring is :gulbarga
```

After Replacing g with a in s1 : apt aulbaraa
The string s4 is : This is a book
After trim() :This is a book

5b. Write a Java program to practice using String Buffer class and its methods.

```
import java.lang.String;
class stringbufferdemo
{
public static void main(String arg[])
{
StringBuffer sb=new StringBuffer("This is my college");
System.out.println("This string sb is : " +sb);
System.out.println("The length of the string sb is : " +sb.length());
System.out.println("The capacity of the string sb is : " +sb.capacity());
System.out.println("The character at an index of 6 is : " +sb.charAt(6));
sb.setCharAt(3,'x');
System.out.println("After setting char x at position 3 : " +sb);
System.out.println("After appending : " +sb.append(" in gulbarga "));
System.out.println("After inserting : " +sb.insert(19,"gpt "));
System.out.println("After deleting : " +sb.delete(19,22));
}
}
```

**********OUTPUT**********

```
This string sb is : This is my college
The length of the string sb is : 18
The capacity of the string sb is : 34
The character at an index of 6 is : s
After setting char x at position 3 : Thix is my college
After appending : Thix is my college in gulbarga
After inserting : Thix is my college gpt in gulbarga
After deleting : Thix is my college in gulbarga
```

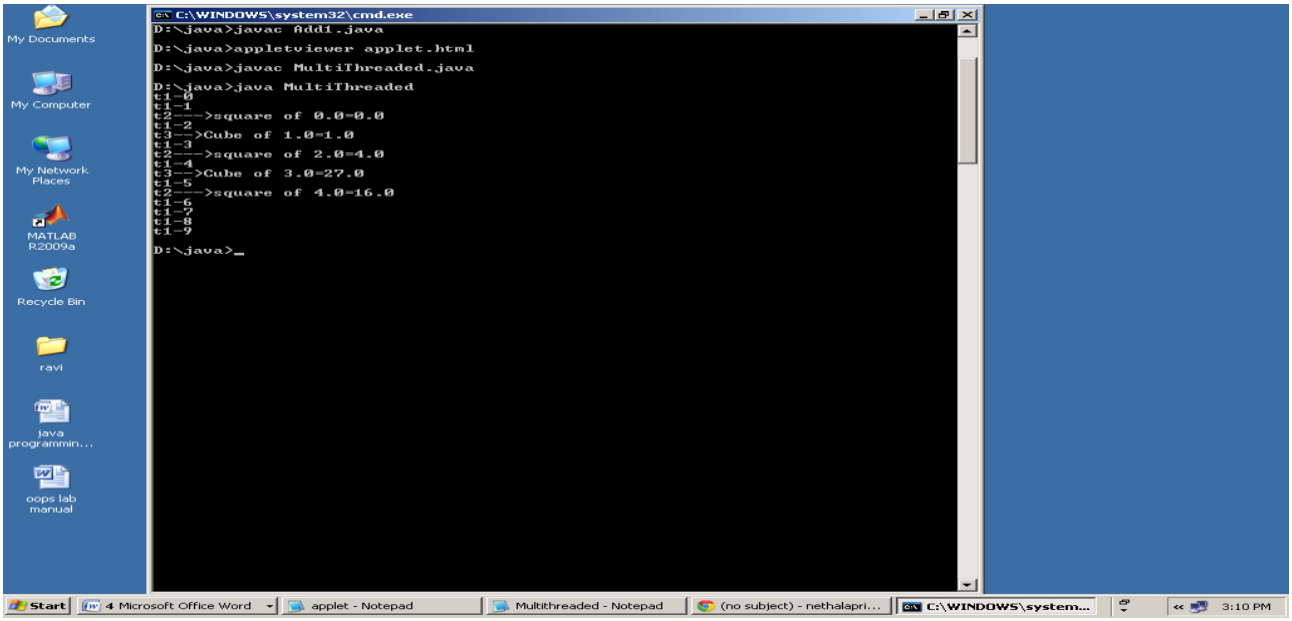
6. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

```
class RandomGenThread implements Runnable
{
double num;
public void run()
{
try {
SquareThread sqt = new SquareThread();
Thread squareThread = new Thread(sqt);
CubeThread cbt = new CubeThread();
Threadcube Thread = new Thread(cbt);
squareThread.start();
cubeThread.start();
for(int i=0;i<10;i++)
{
System.out.println("t1-"+i);
if(i%2 == 0)
{
sqt.setNum(new Double(i));
}
else
{
cbt.setNum(new Double(i));
}
Thread.sleep(1000);
}
} catch (InterruptedException e)
{
e.printStackTrace();
}
}

}
class SquareThread implements Runnable
{
Double num;
public void run()
{
try {

int i=0;
do{
```

```
i++;
if(num != null&&num %2 ==0)
{
System.out.println("t2--->square of "+num+"="+(num*num));
num = null;
}
Thread.sleep(1000);
}while(i<=5);
}
catch (Exception e)
{
e.printStackTrace();
}
}

public Double getNum()
{
return num;
}
public void setNum(Double num)
{
this.num = num;
}
}

class CubeThread implements Runnable
{
Double num;
public void run()
{
try {
int i=0;
do{
i++;
if(num != null&&num%2 !=0)
{
System.out.println("t3-->Cube of "+num+"="+(num*num*num));
num=null;
}
Thread.sleep(1000);
}
while(i<=5);

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

```
}
}
public Double getNum()
{
return num;
}
public void setNum(Double num)
{
this.num = num;
}
}
public class MultiThreaded
{
public static void main(String[] args) throws InterruptedException
{
Thread randomThread = new Thread(new RandomGenThread());
randomThread.start();
}
}
```

**********OUTPUT**********



7. Write a program to implement the concept of Exception handling by creating user defined exceptions.

```
import java.lang.Exception;
import java.lang.*;
import java.lang.Exception;
import java.io.DataInputStream;
class MyException extends Exception
{
MyException(String message)
{
super(message);
}
}
class userdef
{
public static void main(String a[])
{
int age;
DataInputStream ds=new DataInputStream(System.in);
try
{
System.out.println("Enter the age (above 15 abd below 25) :");
age=Integer.parseInt(ds.readLine());
if(age<15 || age> 25)
{
throw new MyException("Number not in range");
}
System.out.println(" the number is :" +age);
}
catch(MyException e)
{
System.out.println("Caught MyException");
System.out.println(e.getMessage());
}
catch(Exception e){ System.out.println(e); }
}
}
```

**********OUTPUT 1**********

```
Enter the age (above 15 abd below 25) :
6
Caught MyException
Number not in range
```

**********OUTPUT 2**********

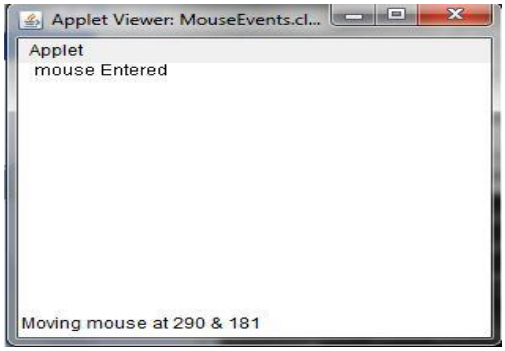
```
Enter the age (above 15 abd below 25) :
20
the number is :20
```

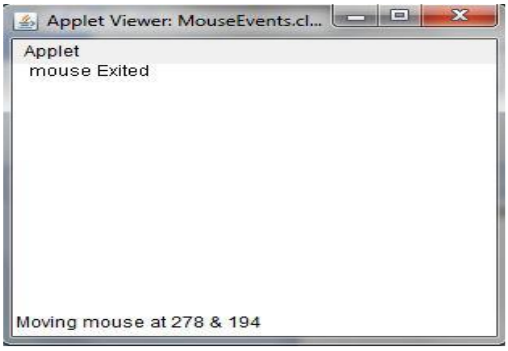
8. Write a Java Program to demonstrate Mouse events

```
import java.applet.*;
import java.awt.event.*;
import java.awt.*;
/* <applet code="MouseEvents.class" width=300 height=200> </applet> */
public class MouseEvents extends Applet implements MouseListener,MouseMotionListener
{
String msg =" ";
int x=0,y=0;
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent m)
{
x=10;
y=10;
msg ="mouse clicked";
repaint();
}
public void mouseEntered(MouseEvent m)
{
x=10;
y=10;
msg ="mouse Entered";
repaint();
}
public void mouseExited(MouseEvent m)
{
x=10;
y=10;
msg ="mouse Exited";
repaint();
}
public void mousePressed(MouseEvent m)
{
x=m.getX();
y=m.getY();
msg ="Down";
repaint();

}
public void mouseReleased(MouseEvent m)
{
x=m.getX();
```

```
y=m.getY();
msg ="Up";
repaint();
}
public void mouseDragged(MouseEvent m)
{
x=m.getX();
y=m.getY();
msg ="*";
showStatus("Draggeg mouse at " +x+ " & "+y);
repaint();
}
public void mouseMoved(MouseEvent m)
{
showStatus("Moving mouse at " +m.getX()+ " & "+m.getY());
}
public void paint(Graphics g)
{
g.drawString(msg,x,y);
}
}
```

**********OUTPUT**********





9. Write a Java program with Servlets to create a dynamic HTML form to accept and display user name and password with the help of 'get()' and 'post()' methods.

loginForm.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>LoginInfo</title>
</head>
<h1>Welcome....!!!</h1>

<body>
<form method = "post"  action = "loginInfo">
Login Id: <input type = "text" name = "name"/>
<br/><br/>
Password: <input type = "password" name = "password"/>
<br/><br/>
<input type = "submit" value = "Login"/>
</form>

</body>
</html>
```

loginInfo.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


/**
 * Servlet implementation class loginInfo
 */
public class loginInfo extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
```

```
    public loginInfo() {
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                doPost(request, response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                System.out.println("inside doPost() mwethod");
                String id = request.getParameter("name");
                String pass = request.getParameter("password");
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                RequestDispatcher rd = null;
                request.setAttribute(id, "name");
                if(id.equals("name") && pass.equals("pass"))
                {
                        out.println("<b>Welcome </b><br>" + id);
                        request.setAttribute("Uname", id);
                        rd = request.getRequestDispatcher("/Welcome");
                        rd.forward(request, response);
                }
                else
                {
                        out.println("<b>Invalid Login Info.</b><br>");
                        rd = request.getRequestDispatcher("/loginForm.html");
                        rd.include(request, response);
                }
                out.close();

        }

}
```

Welcome.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Welcome
 */
public class Welcome extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public Welcome() {
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                doPost(request, response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                String id = (String) request.getAttribute("Uname");
                out.println("<h1>Valid Login Info.</h1> </b><br>");
                out.println("<b> Welcome </b> " + id);
        }

}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>ServDemo</display-name>
  <welcome-file-list>
    <welcome-file>loginForm.html</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <servlet-name>loginInfo</servlet-name>
    <servlet-class>loginInfo</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>loginInfo</servlet-name>
    <url-pattern>/loginInfo</url-pattern>
  </servlet-mapping>
  <servlet>
    <description></description>
    <servlet-name>Welcome</servlet-name>
    <servlet-class>Welcome</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Welcome</servlet-name>
    <url-pattern>/Welcome</url-pattern>
  </servlet-mapping>
</web-app>
```

10. Write a Java Servlet program to demonstrate session tracking.

login.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>LoginInfo</title>
</head>
<h1>Welcome....!!!</h1>

<body>
<form method = "get"  action = "sessionTrack">
Login Id: <input type = "text" name = "name"/>
<br/><br/>
Password: <input type = "password" name = "password"/>
<br/><br/>
<input type = "submit" value = "Login"/>
</form>

</body>
</html>
```

sessionTrack.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class sessionTrack
 */
public class sessionTrack extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public sessionTrack() {
        // TODO Auto-generated constructor stub
```

```
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // TODO Auto-generated method stub
                HttpSession session = request.getSession(true);
                Date createTime = new Date(session.getCreationTime());
                Date lastAccessTime = new Date(session.getLastAccessedTime());
                String title = "Welcome";
                Integer visitCount = new Integer(0);
                String visitCountKey = new String("visitCount");
                //String userIDKey = new String("userID");
                String userIDKey = request.getParameter("name");
                //String userID = new String("ABCD");
                String userID = request.getParameter("password");
                if (session.isNew())
                {
                        title = "Welcome to my website";
                        session.setAttribute(userIDKey, userID);
                }
                 else
                 {
                        visitCount = (Integer)session.getAttribute(visitCountKey);
                        visitCount = visitCount + 1;
                        userID = (String)session.getAttribute(userIDKey);
                }
                session.setAttribute(visitCountKey,  visitCount);
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();

        String docType =
        "<!doctype html public \"-//w3c//dtd html 4.0 " +
        "transitional//en\">\n";

        out.println(docType +
        "<html>\n" +
        "<head><title>" + title + "</title></head>\n" +

        "<body bgcolor = \"#f0f0f0\">\n" +
        "<h1 align = \"center\">" + title + "</h1>\n" +
        "<h2 align = \"center\">Session Infomation</h2>\n" +
        "<table border = \"1\" align = \"center\">\n" +
```

```
            "<tr bgcolor = \"#949494\">\n" +
            "  <th>Session info</th><th>value</th></tr>\n" +

            "<tr>\n" +
            "  <td>id</td>\n" +
            "  <td>" + session.getId() + "</td></tr>\n" +

            "<tr>\n" +
            "  <td>Creation Time</td>\n" +
            "  <td>" + createTime + "  </td></tr>\n" +

            "<tr>\n" +
            "  <td>Time of Last Access</td>\n" +
            "  <td>" + lastAccessTime + "  </td></tr>\n" +

            "<tr>\n" +
            "  <td>User ID</td>\n" +
            "  <td>" + userIDKey + "  </td></tr>\n" +

            "<tr>\n" +
            "  <td>Number of visits</td>\n" +
            "  <td>" + visitCount + "</td></tr>\n" +
            "</table>\n" +
            "</body></html>"
            );

            }

            /**
             * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
             */
            protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                        // TODO Auto-generated method stub
                        doGet(request, response);
            }

}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>SessDemo</display-name>
  <welcome-file-list>
    <welcome-file>login.html</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>sessionTrack</display-name>
    <servlet-name>sessionTrack</servlet-name>
    <servlet-class>sessionTrack</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>sessionTrack</servlet-name>
    <url-pattern>/sessionTrack</url-pattern>
  </servlet-mapping>
</web-app>
```

11. Write a JAVA JSP program to implement verification of a particular user login and display a welcome page.

login.jsp

```
<html>
<head> <title> Login page </title>
</head>
<body>
<form action="validation.jsp">
<table border="0">
<tr>
<td> USER ID: </td>
<td>
<input type="text" name="uname" /> <br>
</td>
</tr>
<tr>
<td> PASSWORD: </td>
<td>
<input type="password" name="password" /> <br>
</td>
</tr>
<tr> <td align ="center">
<input type="submit" value="submit" >
<input type="reset" value="reset">
</td>
</tr>
</form>
</body>
</html>
```

validation.jsp

```
<html>
<body>
<%! String uid="student"; %>
<%! String pass="aitmca"; %>
<%! String id, password; %>
<% id=request.getParameter("uname"); %>
<% password=request.getParameter("password"); %>
<% if(uid.equals(id)&&pass.equals(password))
{
%>
<jsp:forward page="welcome.jsp"/>
<%
}
else
```

```
%>
<jsp:forward page="error.jsp" />
<%
}
%>
</body>
</html>
```

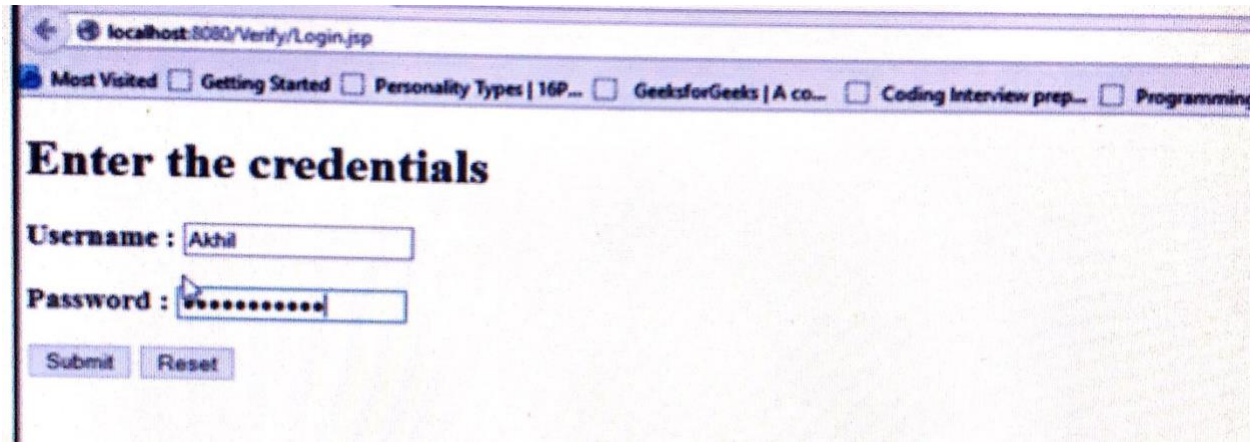
welcome.jsp

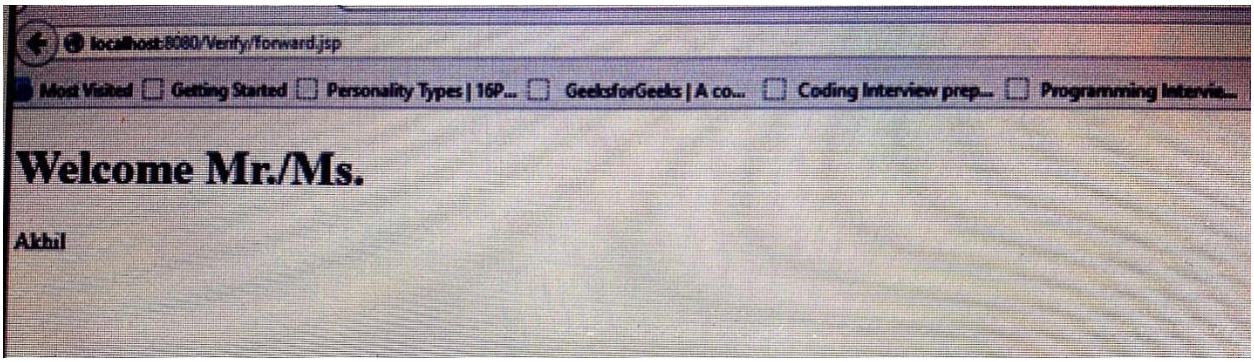
```
<html>
<body bgcolor="pink">
<center>
<%! String id; %>
<% id=request.getParameter("uname"); %>
<h1> welcome <%=id%> to the home page </h1>
</center>
</body>
</html>
```

error.jsp

```
<html>
<head> <title>JSP Page</title> </head>
<body bgcolor="pink">
<h1>INVALID ENTRY</h1>
</body>
</html>
```

**********OUTPUT**********




localhost:8080/Verify/forward.jsp
Most Visited
Getting Started
Personality Types | 16P...
GeeksforGeeks | A co...
Coding Interview prep...
Programming Intervie...
Welcome Mr./Ms.
Akhil

Viva Questions

1. What is JVM?
2. What is the most important feature of java?
3. What do you mean by platform independence?
4. What are the access modifiers in java?
5. What is an abstract class?
6. Why is java architectural neutral?
7. Why is java considered dynamic?
8. List two java IDE's
9. What is a constructor?
10. What is a bytecode?
11. Why new operator is used?
12. When ParseInt () method is used?
13. What is an exception?
14. Define inheritance.
15. What is polymorphism.
16. What is abstraction.
17. Define packages in Java.
18. What is an applet?
19. Explain garbage collection in Java.
20. List primitive Java types.
21. What is the difference between error and exception?
22. What is type casting?
23. What is JSP?
24. What is event handling?
25. What is the difference between an inner class and subclass?
26. What is Final keyword in Java used for?
27. What is ternary operator?
28. What is multi-threading?
29. What is an instance variable?
30. What is singleton class?
31. When throws keyword is used?
32. What is an interface?
33. Define immutable object.
34. What is a JAR file?
35. Difference between overloading and overriding.
36. What are wrapper classes?
37. What is the difference between swing and AWT components?
38. Which Java operator is right associative?
39. What are class loaders?
40. What is a locale class?

41. What is dynamic binding?
42. Difference between java and C++.
43. What is JVM?
44. Explain JRE and JDK.
45. What is a servlet?
46. What is request dispatcher?
47. What is the life cycle of a servlet?
48. What are the different methods of session management in servlets?
49. How do cookies work in servlets?
50. What are the differences between get and post methods.