LAB ASSIGNMENT

Name: Deepankar Sharma
Student ID: 233512013

Course Code: OMC209
Course Title: Advanced Java Programming Laboratory

Last Date of Submission: 31/07/24
Maximum Marks: 30

Program 1:

Create a Java class "Employee" with the attributes - employee number, name, designation and salary.

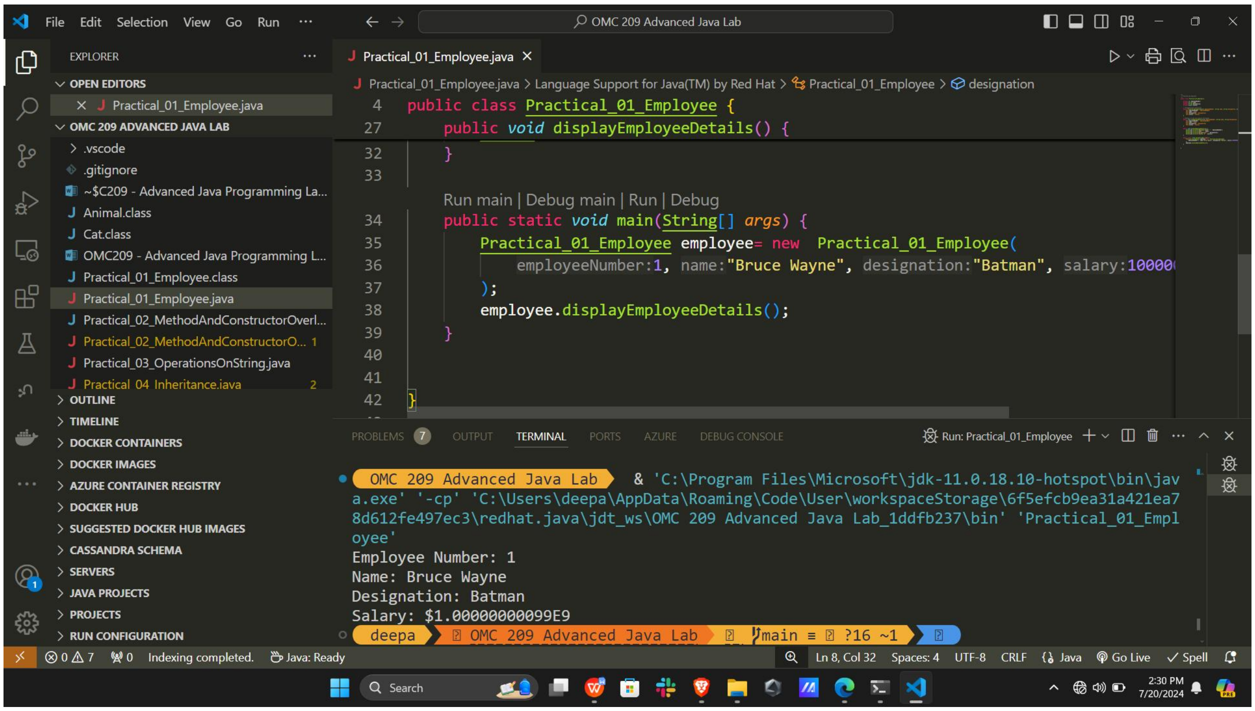
a. Implement methods to set and display these attributes

b. Implement parameterized constructors for initializing these attributes.

Use "this" keyword to illustrate the difference between instance variable and constructor parameters. Display these attributes.

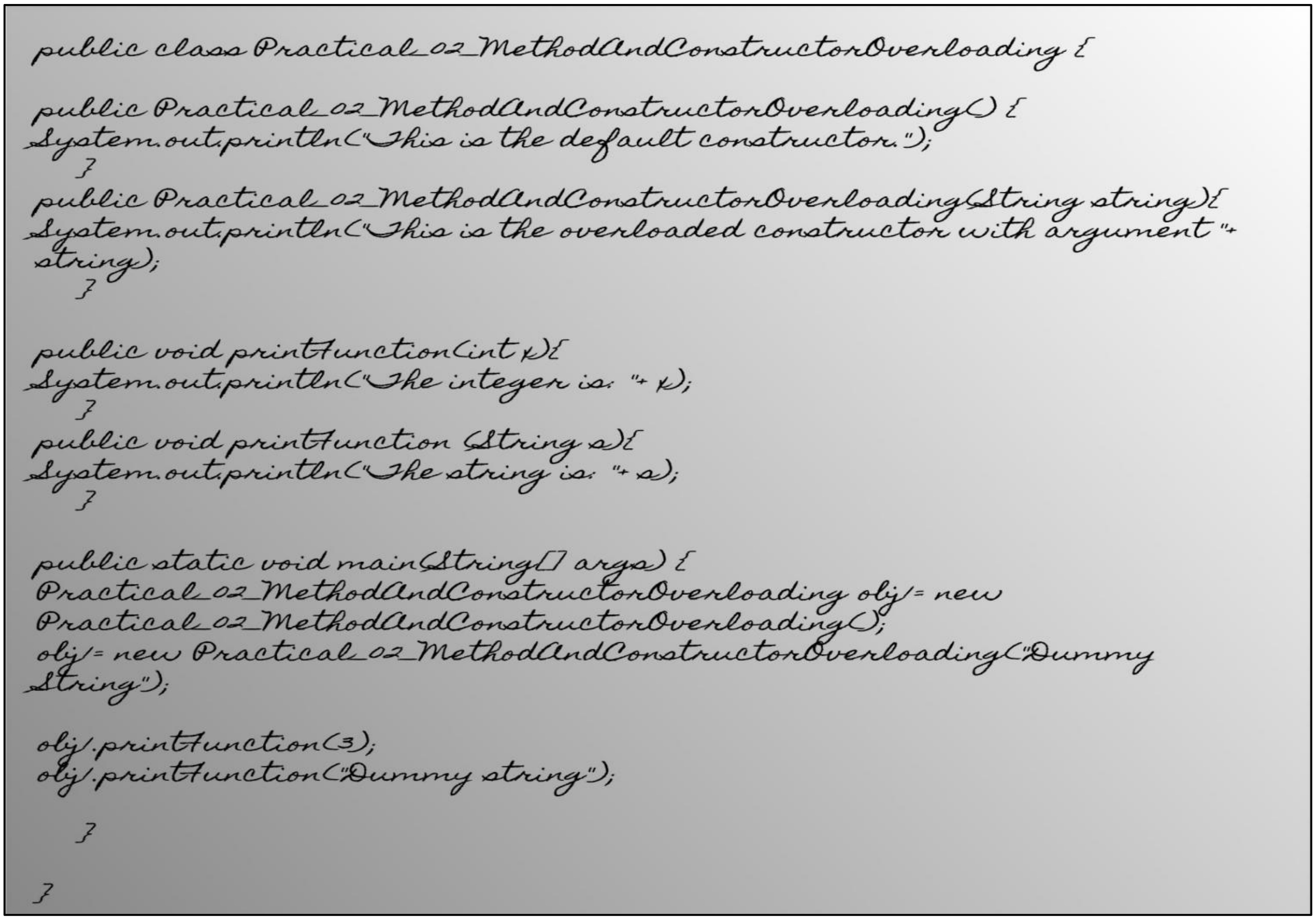
```
public class Practical_01_Employee {
private int employeeNumber;
private String name;
private String designation;
private double salary;
public Practical_01_Employee(int employeeNumber, String name, String
designation, double salary) {
this.employeeNumber = employeeNumber;
this.name = name;
this.designation = designation;
this.salary = salary;
}
public void setEmployeeDetails(int employeeNumber, String name, String
designation, double salary) {
this.employeeNumber = employeeNumber;
this.name = name;
this.designation = designation;
this.salary = salary;
}
public void displayEmployeeDetails() {
System.out.println("Employee Number: " + employeeNumber);
System.out.println("Name: " + name);
System.out.println("Designation: " + designation);
System.out.println("Salary: ₹" + salary);
}
public static void main(String[] args) {
Practical_01_Employee employee= new Practical_01_Employee(
1, "Bruce Wayne", "Batman", 1000000000.99
);
employee.displayEmployeeDetails();
}
}
```

Output:


Practical_01_Employee.java
public class Practical_01_Employee {
public void displayEmployeeDetails() {
}
Run main | Debug main | Run | Debug
public static void main(String[] args) {
Practical_01_Employee employee= new Practical_01_Employee(
employeeNumber:1, name:"Bruce Wayne", designation:"Batman", salary:10000
);
employee.displayEmployeeDetails();
}
}
OMC 209 Advanced Java Lab & 'C:\Program Files\Microsoft\jdk-11.0.18.10-hotspot\bin\java.exe' '-cp' 'C:\Users\deepa\AppData\Roaming\Code\User\workspaceStorage\6f5efcb9ea31a421ea78d612fe497ec3\redhat.java\jdt_ws\OMC 209 Advanced Java Lab_1ddfb237\bin' 'Practical_01_Employee'
Employee Number: 1
Name: Bruce Wayne
Designation: Batman
Salary: $1.00000000099E9

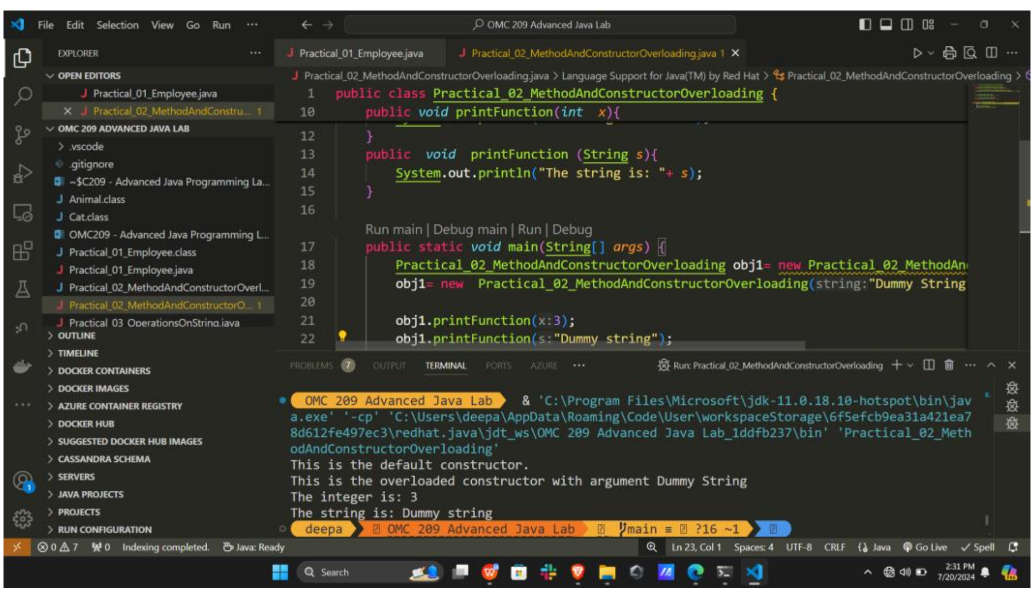
Program 2:

Write a Java program to demonstrate method overloading and constructor overloading.



```
public class Practical_02_MethodAndConstructorOverloading {

public Practical_02_MethodAndConstructorOverloading() {
System.out.println("This is the default constructor.");
  }
public Practical_02_MethodAndConstructorOverloading(String string){
System.out.println("This is the overloaded constructor with argument "+
string);
  }

public void printFunction(int x){
System.out.println("The integer is: "+ x);
  }
public void printFunction (String s){
System.out.println("The string is: "+ s);
  }

public static void main(String[] args) {
Practical_02_MethodAndConstructorOverloading obj1= new
Practical_02_MethodAndConstructorOverloading();
obj1= new Practical_02_MethodAndConstructorOverloading("Dummy
String");

obj1.printFunction(3);
obj1.printFunction("Dummy string");

  }

}
```

Output:



Program 3:

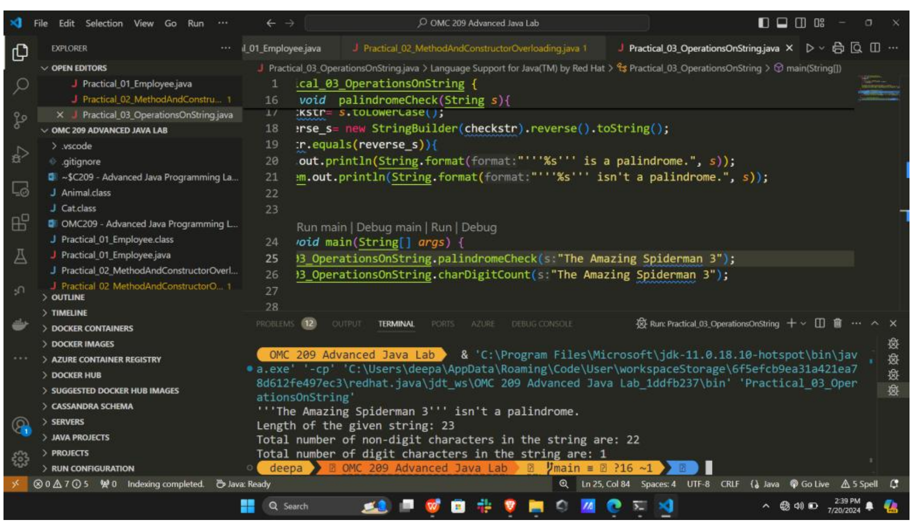
Write a Java program to perform the following operations on a string.

a. Count the number of characters and digits.

b. Check whether the given string is palindrome or not.

```
public class Practical_03_OperationsOnString {
public static void charDigitCount(String s){
 int char_count= 0;
 int digits= 0;

for (int i = 0; i < s.length(); i++) {
 if (s.charAt(i)>=48 && s.charAt(i)<=57) digits++;
 else char_count++;
      }

System.out.println("Length of the given string: "+ s.length());
System.out.println("Total number of non-digit characters in the string
 are: "+ char_count);
System.out.println("Total number of digit characters in the string are: "+
 digits);
   }

public static void palindromeCheck(String s){
String checkstr= s.toLowerCase();
String reverse_s= new StringBuilder(checkstr).reverse().toString();
if (checkstr.equals(reverse_s)){
System.out.println(String.format("'''%s''' is a palindrome.", s));
 else System.out.println(String.format("'''%s''' isn't a palindrome.", s));
   }

public static void main(String[] args) {
 Practical_03_OperationsOnString.palindromeCheck("The Amazing
Spiderman 3");
 Practical_03_OperationsOnString.charDigitCount("The Amazing Spiderman
 3");
   }
 }
```

Output:

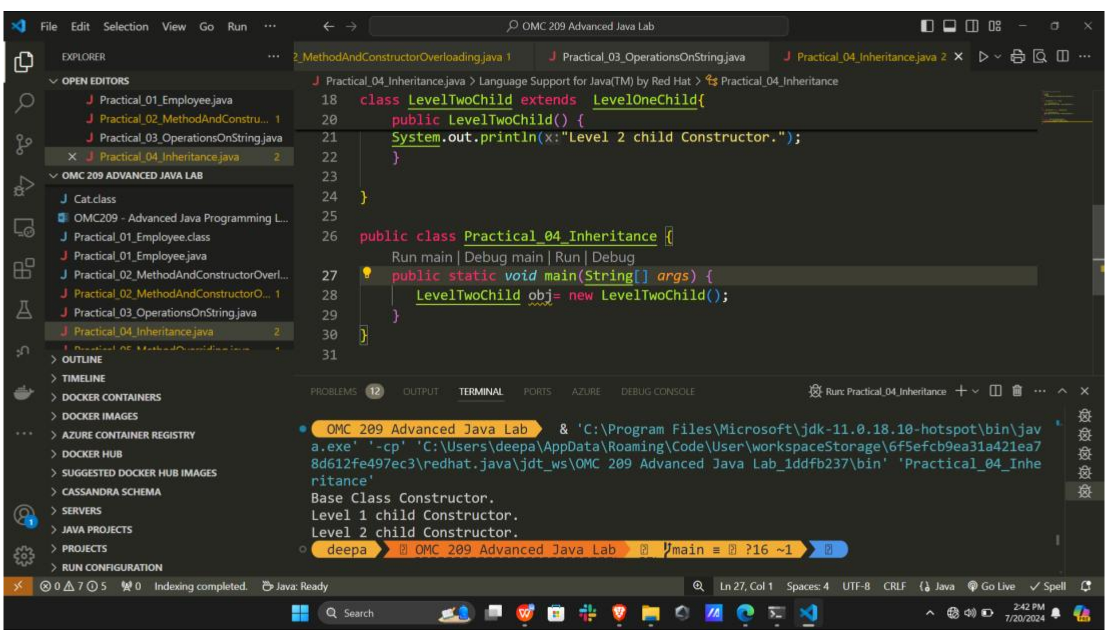


Program 4:

Write a Java program to demonstrate single and multi-level inheritance. Display the order of execution of constructors in multi-level inheritance.

```
class Base{
Base(){
System.out.println("Base Class Constructor.");
  }
}

class LevelOneChild extends Base{

public LevelOneChild() {
System.out.println("Level 1 child Constructor.");
  }

}

class LevelTwoChild extends LevelOneChild{

public LevelTwoChild() {
System.out.println("Level 2 child Constructor.");
  }

}

public class Practical_04_Inheritance {
public static void main(String[] args) {
LevelTwoChild obj= new LevelTwoChild();
  }
}
```

Output:

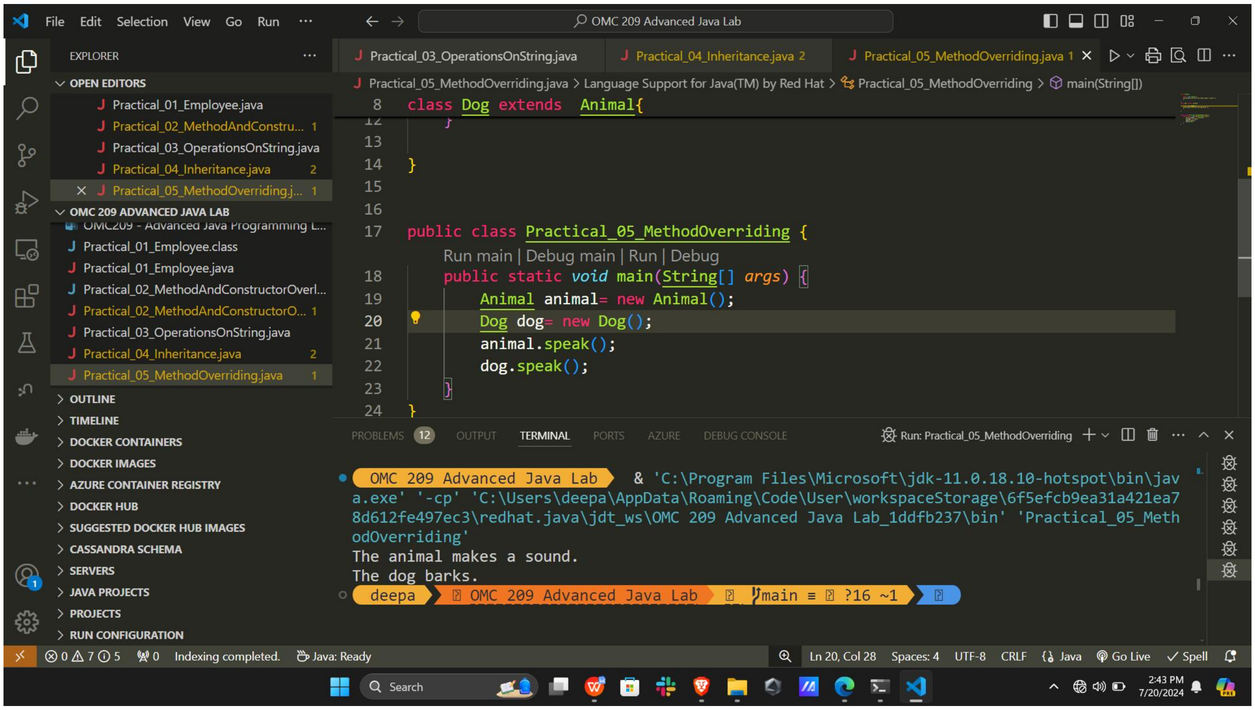


Program 5:

Write a Java program to demonstrate method overriding.

```java
class Animal{

public void speak() {
System.out.println("The animal makes a sound.");
   }


}
class Dog extends Animal{

public void speak() {
System.out.println("The dog barks.");
   }


}


public class Practical_05_MethodOverriding {
public static void main(String[] args) {
Animal animal= new Animal();
Dog dog= new Dog();
animal.speak();
dog.speak();
   }
}
```

Output:



OMC 209 Advanced Java Lab
Practical_05_MethodOverriding.java
class Dog extends Animal{
}
public class Practical_05_MethodOverriding {
Run main | Debug main | Run | Debug
public static void main(String[] args) {
Animal animal= new Animal();
Dog dog= new Dog();
animal.speak();
dog.speak();
}
}
PROBLEMS OUTPUT TERMINAL PORTS AZURE DEBUG CONSOLE
OMC 209 Advanced Java Lab & 'C:\Program Files\Microsoft\jdk-11.0.18.10-hotspot\bin\java.exe' '-cp' 'C:\Users\deepa\AppData\Roaming\Code\User\workspaceStorage\6f5efcb9ea31a421ea78d612fe497ec3\redhat.java\jdt_ws\OMC 209 Advanced Java Lab_1ddfb237\bin' 'Practical_05_MethodOverriding'
The animal makes a sound.
The dog barks.
Ln 20, Col 28 Spaces: 4 UTF-8 CRLF Java

Program 6:

Write a Java program to illustrate the use of

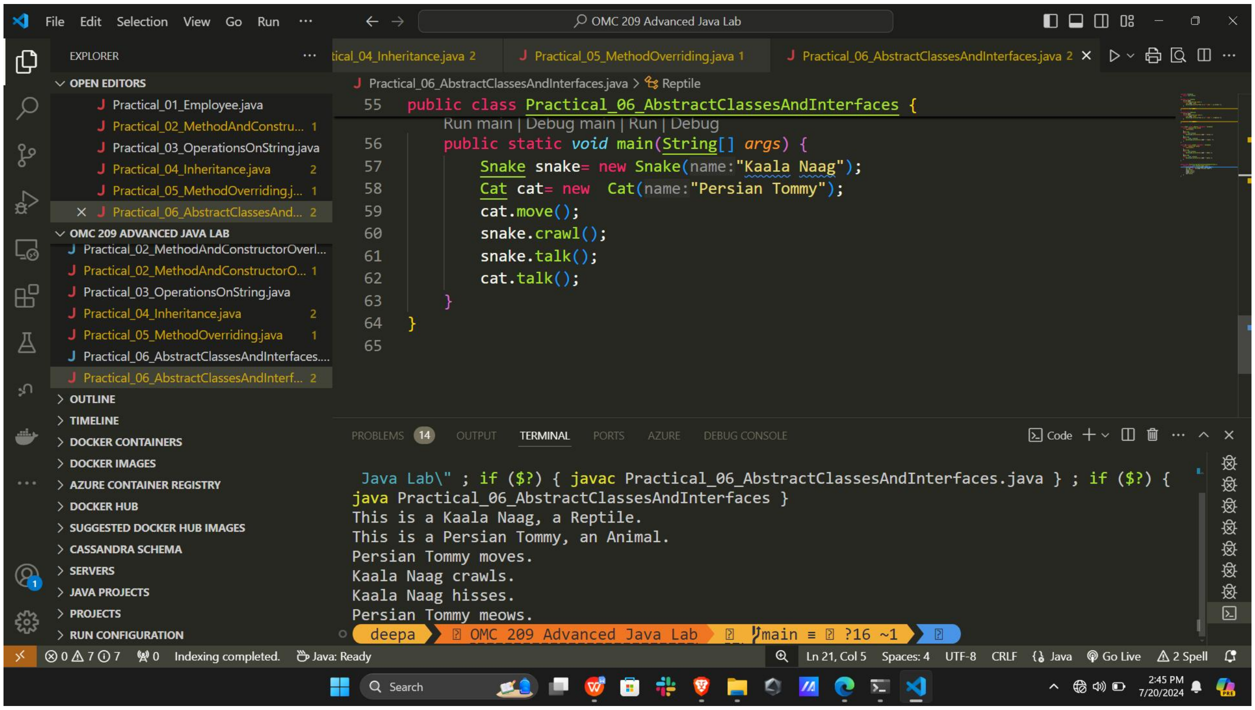
a. Abstract class

b. Interfaces in Java

```
interface Talkable{
public void talk();
}

abstract class Animal{
String name;
public Animal(String name) {
this.name= name;
System.out.println("This is a "+ name+ ", an Animal.");
    }

abstract void move();
}

abstract class Reptile{
String name;
public Reptile(String name) {
this.name= name;
System.out.println("This is a "+ name+ ", a Reptile.");
    }

abstract void crawl();
}


class Snake extends Reptile implements Talkable{
public Snake(String name){
super(name);
    }
@Override
public void talk(){
System.out.println(this.name+ " hisses.");
    }
@Override
public void crawl(){
System.out.println(this.name+ " crawls. ");
```

```
}
}

class Cat extends Animal implements Talkable{
public Cat(String name){
super(name);
    }
@Override
public void talk(){
System.out.println(this.name+ " meows.");
    }
@Override
public void move(){
System.err.println(this.name+ " moves.");
    }
}


public class Practical_06_AbstractClassesAndInterfaces {
public static void main(String[] args) {
Snake snake= new Snake("Kaala Naag");
Cat cat= new Cat("Persian Tommy");
cat.move();
snake.crawl();
snake.talk();
cat.talk();
    }
}
```

Output:

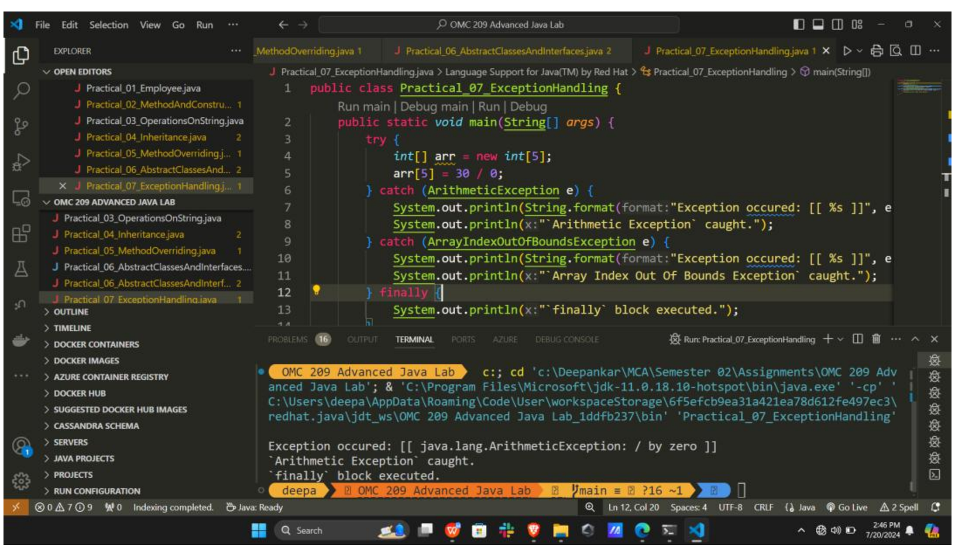

Java Lab\" ; if ($?) { javac Practical_06_AbstractClassesAndInterfaces.java } ; if ($?) {
java Practical_06_AbstractClassesAndInterfaces }
This is a Kaala Naag, a Reptile.
This is a Persian Tommy, an Animal.
Persian Tommy moves.
Kaala Naag crawls.
Kaala Naag hisses.
Persian Tommy meows.

Program 7:

Write a Java program to demonstrate exception handling. Show the order of execution of “try”, “catch” and “finally” blocks when an exception occurs and when it does not occur during the execution by providing appropriate inputs during execution and displaying messages.

```
public class Practical_07_ExceptionHandling {
public static void main(String[] args) {
try {
int[] arr = new int[5];
arr[5] = 30 / 0;
 catch (ArithmeticException e) {
System.out.println(String.format("Exception occured: [[ %s ]]", e));
System.out.println("`Arithmetic Exception` caught.");
 catch (ArrayIndexOutOfBoundsException e) {
System.out.println(String.format("Exception occured: [[ %s ]]", e));
System.out.println("`Array Index Out Of Bounds Exception` caught.");
 finally {
System.out.println("`finally` block executed.");
     }
   }
}
```

Output:

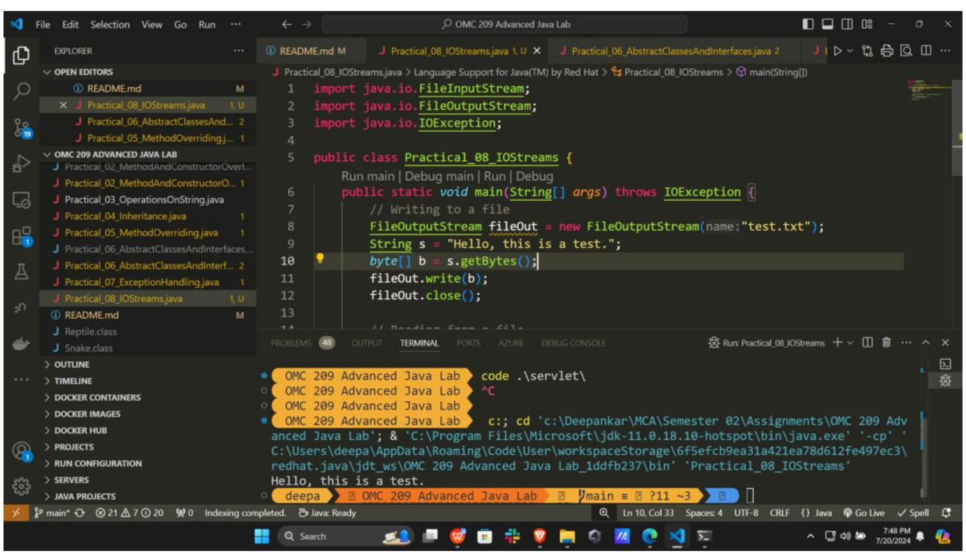


Program 8:

Write a Java program to illustrate the use of I/O streams.

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class Practical_08_IOStreams {
public static void main(String[] args) throws IOException {
  Writing to a file
FileOutputStream fileOut = new FileOutputStream("test.txt");
String s = "Hello, this is a test.";
byte[] b = s.getBytes();
fileOut.write(b);
fileOut.close();

  Reading from a file
FileInputStream fileIn = new FileInputStream("test.txt");
int i;
while ((i = fileIn.read()) != -1) {
System.out.print((char) i);
      }
fileIn.close();
   }
}
```

Output:



Program 9:

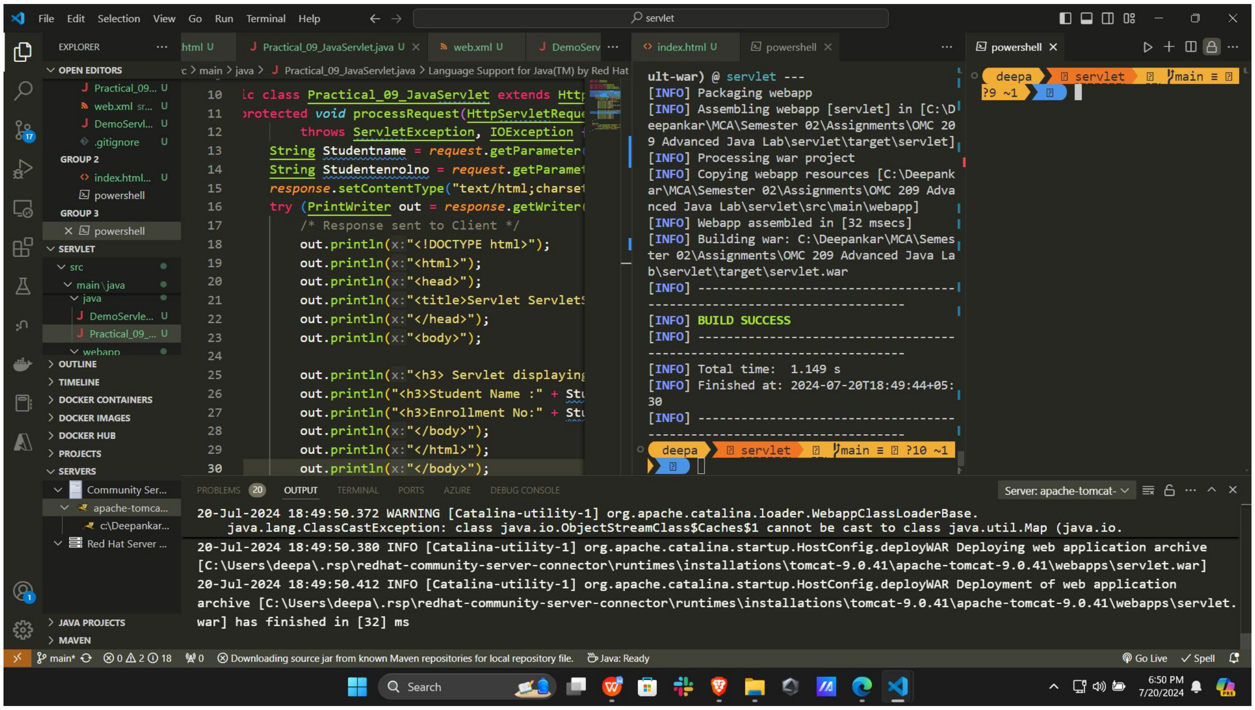
Write a Java Servlet program to implement a dynamic HTML using Servlet(Student name and enrolment number should be accepted using HTML and displayed using a Servlet).

```
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;

public class Practical_09_JavaServlet extends HttpServlet {
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
String Studentname = request.getParameter("name");
String Studentenrolno = request.getParameter("enrlno");
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
 *Response sent to Client */
out.println("<!DOCTYPE html>"); out.println("<html>"); out.println("<head>");
out.println("<title Servlet ServletStudent</title>"); out.println("</head>");
out.println("<body>"); out.println("<h3> Servlet displaying accepted Student
Name and Enrolment No. </h3>"); out.println("<h3>Student Name :" +
Studentname + "</h3><br>"); out.println("<h3>Enrollment No:" + Studentenrolno
+ "</h3>"); out.println("</body>"); out.println("</html>"); out.println("</body>");
out.println("</html>");     }
    }

@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
    }
}
```

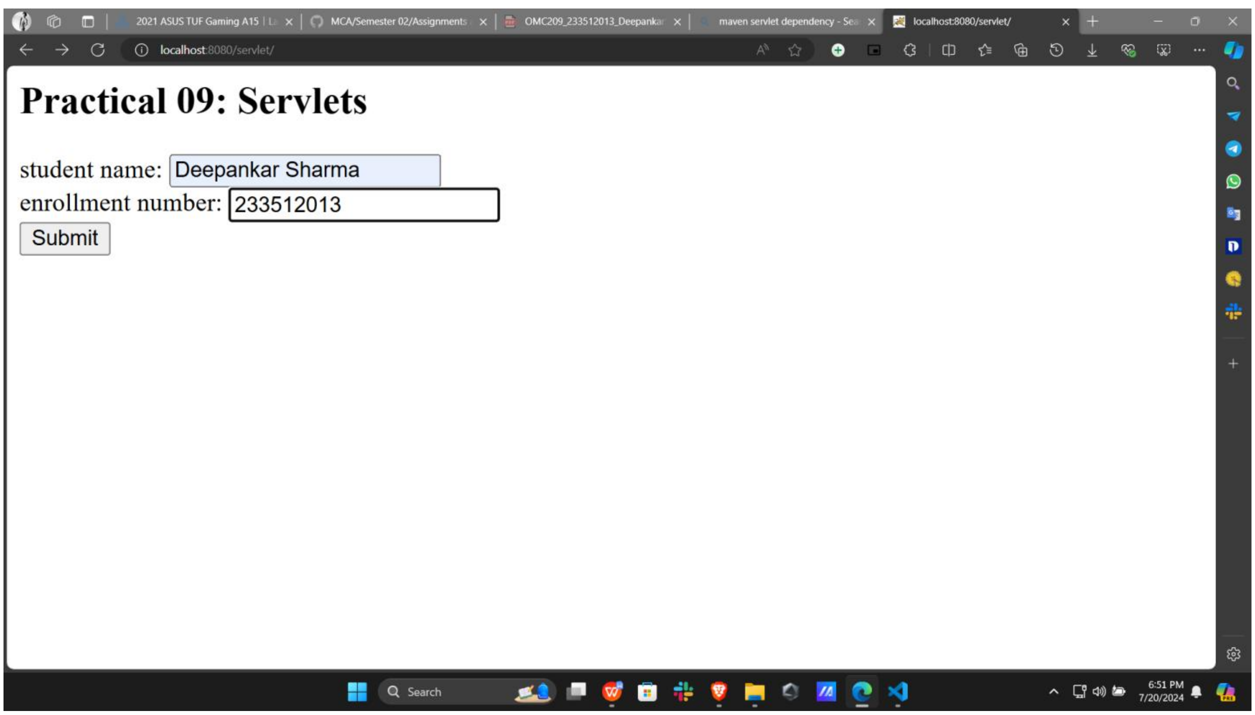
```
<web-app>
<servlet>
      <servletname>s1</servletname>
      <servlet-class>DemoServlet</servlet-class>
</servlet>
 <servlet>
      <servletname>practical09</servletname>
   <servlet-class>Practical_09_JavaServlet</servlet-
class>
 </servlet>

 <servlet>
  <servletname>practical10_FetchServlet</servlet-
name>
 <servlet-
class>Practical_10_FetchCookieServlet</servlet-
class>
 </servlet>
 <servlet>
    <servletname>practical10_SetServlet</servlet-
name>
 <servlet-
class>Practical_10_SetCookieServlet</servlet-class>
 </servlet>
 <servlet>
  <servlet-
name>practical11_GetPostServlet</servletname>
<servlet-class>Practical_11_GetPostServlet</servlet-
class> </servlet>
 <servletmapping>
     <servletname>s1</servletname>
    <url-pattern>/display</url-pattern>
 </servletmapping>
 <servletmapping>
```

```
10pt;">      <servletname>practical09</servletname>      <url-
pattern>/practical_09</url-pattern> </servletmapping>
 <servletmapping>
   <servletname>practical10_FetchServlet</servlet
name>
 <url-pattern>/FetchCookieServlet</url-pattern>
</servletmapping>
 <servletmapping>
     <servletname>practical10_SetServlet</servlet
name>
       <url-pattern>/SetCookieServlet</url-pattern>
</servletmapping>
 <servletmapping>
   <servlet
name>practical11_GetPostServlet</servletname>
<url-pattern>/GetPostServlet</url-pattern> </servlet
mapping>
 <session-config>
    <session-timeout> 30 </session-timeout>
 </session-config>
</web-app>
```

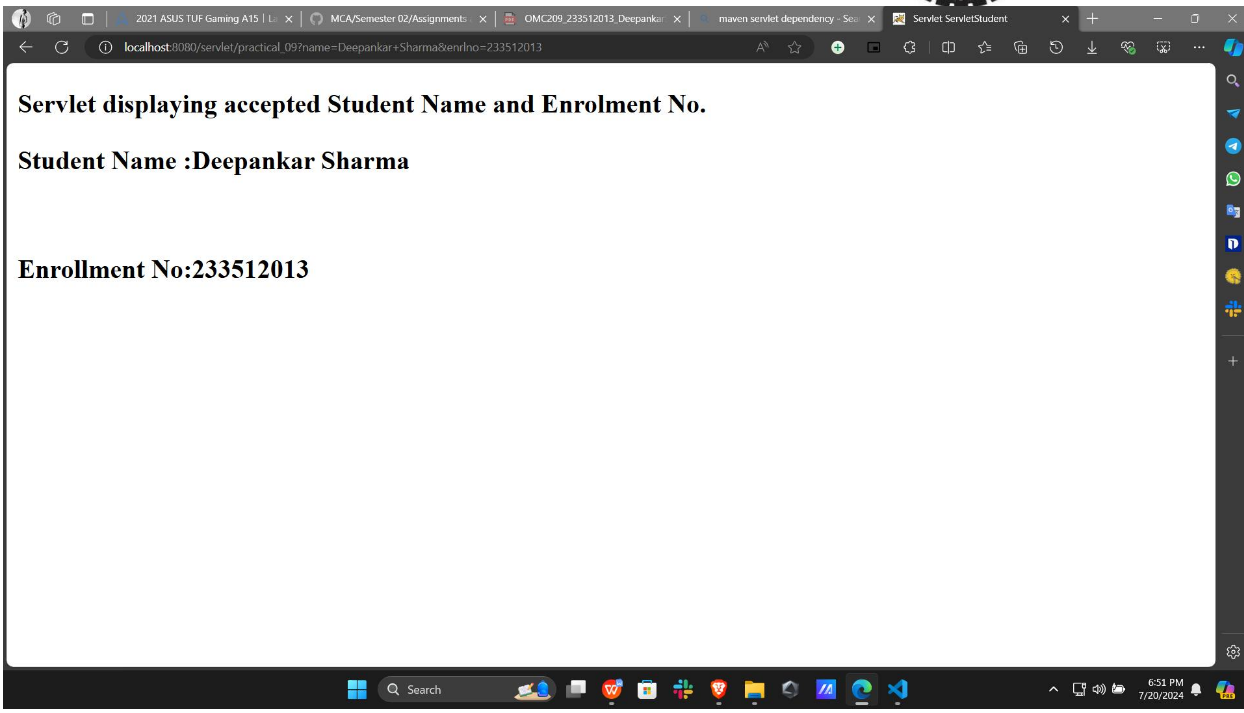
Output:



File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
SERVLET
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
PROJECTS
SERVERS
JAVA PROJECTS
MAVEN
String Studentname = request.getParameter
String Studentenrolno = request.getParamet
response.setContentType("text/html;charset
try (PrintWriter out = response.getWriter(
/* Response sent to Client */
out.println(x:"<!DOCTYPE html>");
out.println(x:"<html>");
out.println(x:"<head>");
out.println(x:"<title>Servlet Servlet
out.println(x:"</head>");
out.println(x:"<body>");
out.println(x:"<h3> Servlet displaying
out.println("<h3>Student Name :" + Stu
out.println("<h3>Enrollment No:" + Stu
out.println(x:"</body>");
out.println(x:"</html>");
out.println(x:"</body>");
[INFO] Packaging webapp
[INFO] Processing war project
[INFO] BUILD SUCCESS
[INFO] Total time: 1.149 s
[INFO] Finished at: 2024-07-20T18:49:44+05:30
20-Jul-2024 18:49:50.372 WARNING [Catalina-utility-1] org.apache.catalina.loader.WebappClassLoaderBase.
java.lang.ClassCastException: class java.io.ObjectStreamClass$Caches$1 cannot be cast to class java.util.Map (java.io.
20-Jul-2024 18:49:50.380 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive
[C:\Users\deepa\.rsp\redhat-community-server-connector\runtimes\installations\tomcat-9.0.41\apache-tomcat-9.0.41\webapps\servlet.war]
20-Jul-2024 18:49:50.412 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application
archive [C:\Users\deepa\.rsp\redhat-community-server-connector\runtimes\installations\tomcat-9.0.41\apache-tomcat-9.0.41\webapps\servlet.
war] has finished in [32] ms



localhost:8080/servlet/
Practical 09: Servlets
student name: Deepankar Sharma
enrollment number: 233512013
Submit


2021 ASUS TUF Gaming A15
MCA/Semester 02/Assignments
OMC209_233512013_Deepankar
maven servlet dependency - Sea
Servlet ServletStudent
localhost:8080/servlet/practical_09?name=Deepankar+Sharma&enrlno=233512013
Servlet displaying accepted Student Name and Enrolment No.
Student Name :Deepankar Sharma
Enrollment No:233512013
Search
6:51 PM
7/20/2024

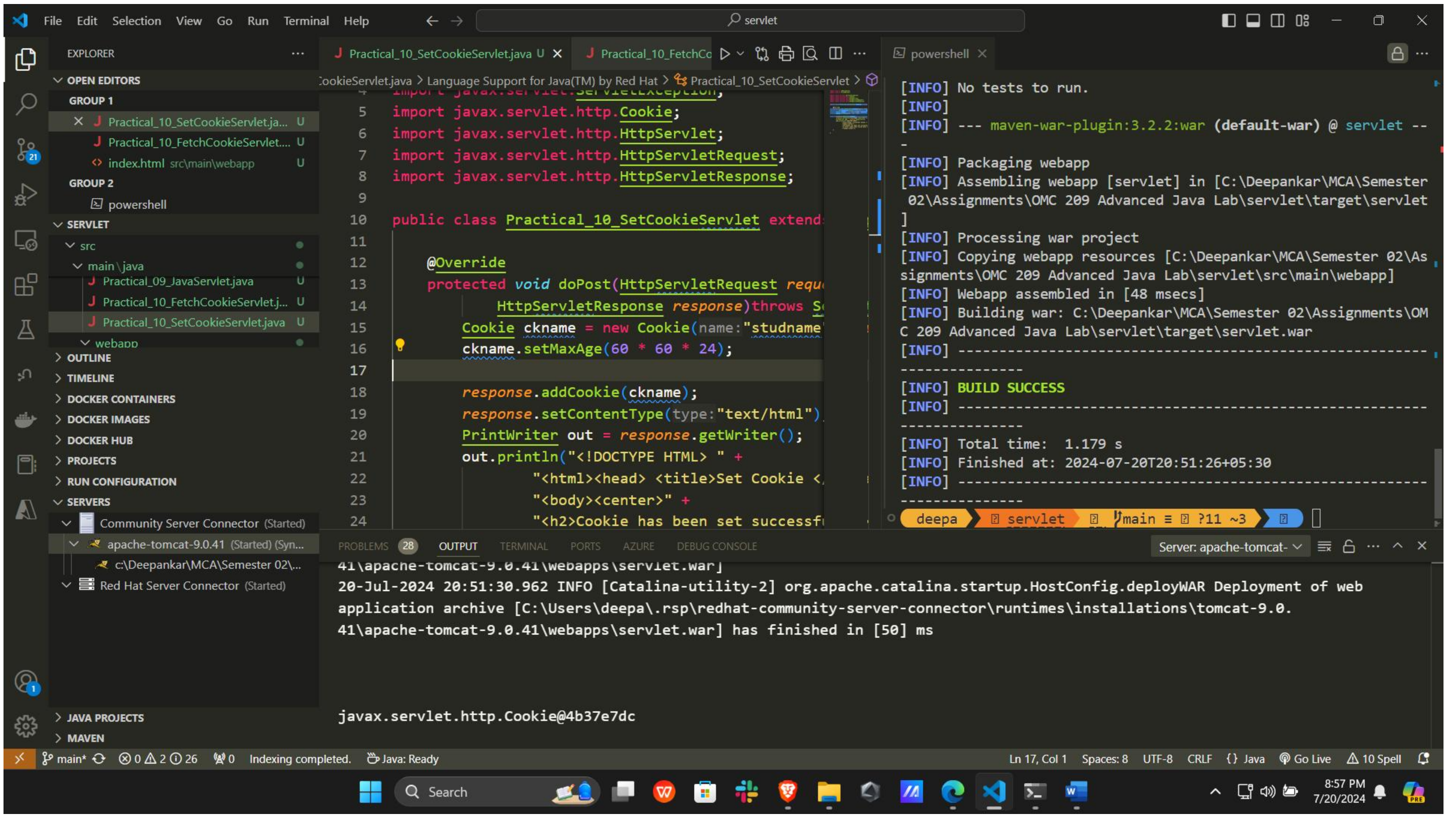
Program 10:

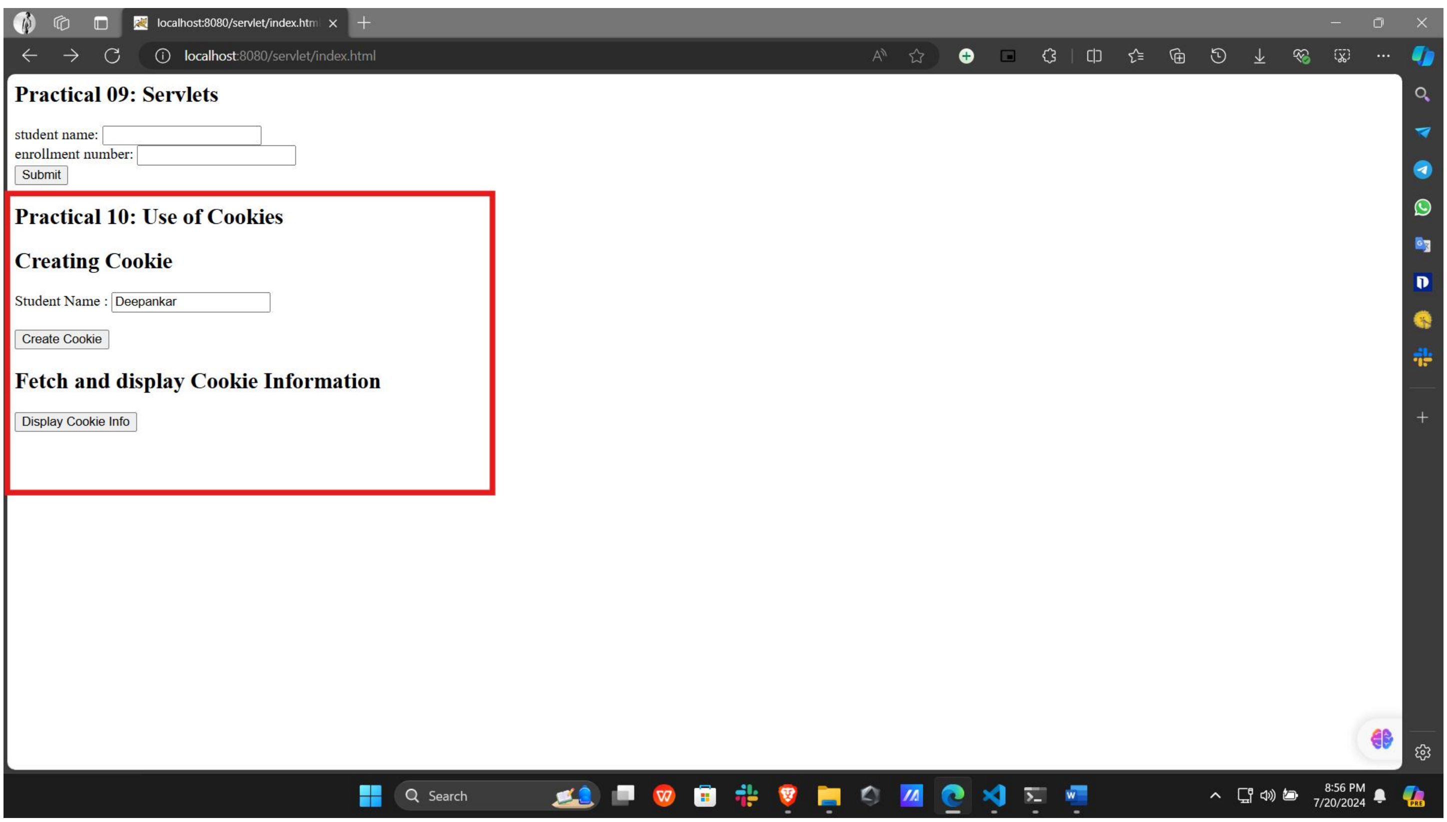
Write a Java Servlet program to demonstrate the use of cookies.

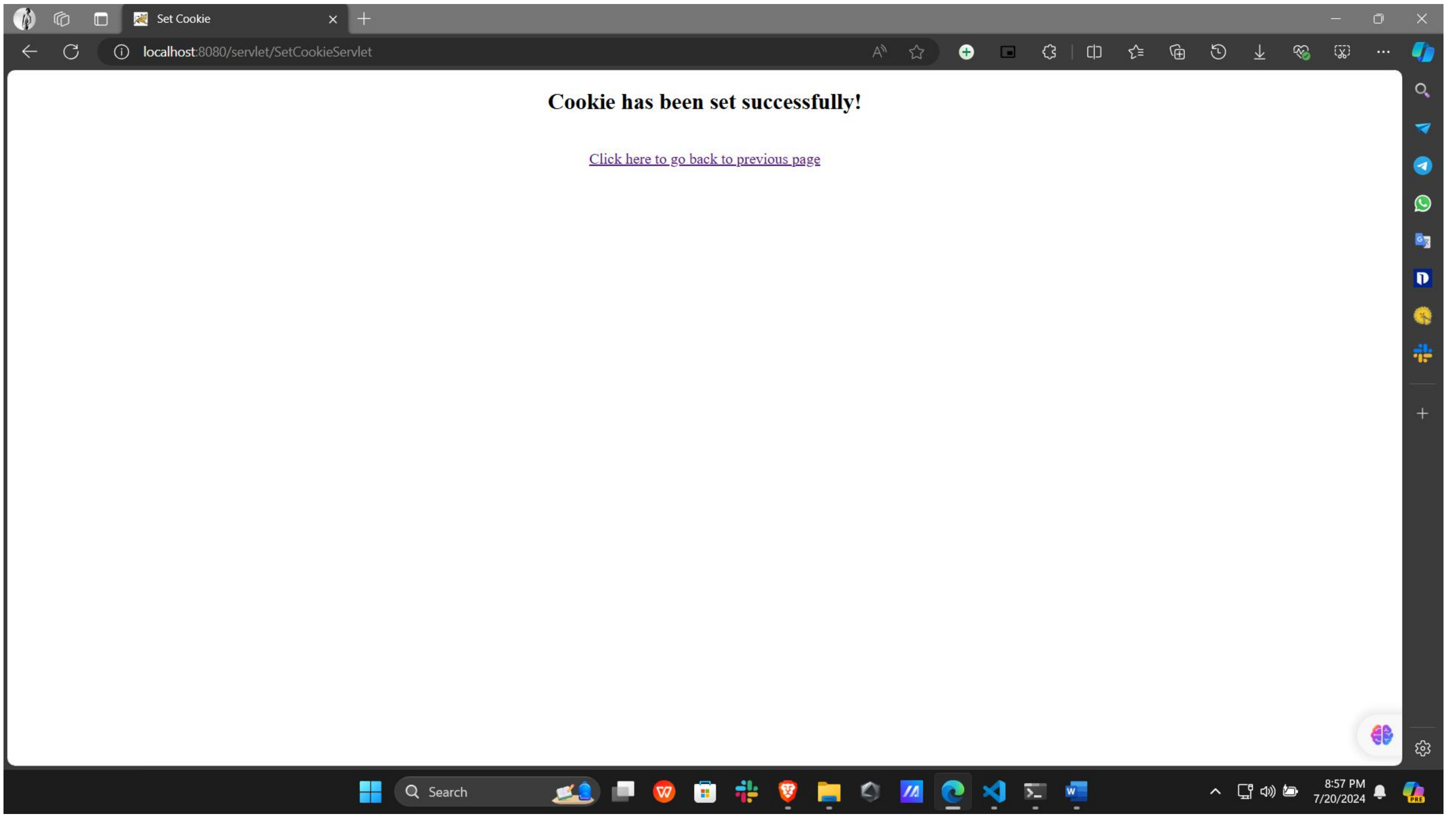
```
import java.io.IOException;
import java.io.PrintWriter;

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

public class Practical_10_SetCookieServlet extends HttpServlet {

@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)throws ServletException, IOException {
Cookie ckname = new
Cookie("studname",request.getParameter("txtstudname"));
ckname.setMaxAge(60 * 60 * 24);

response.addCookie(ckname);
response.setContentType("text/html");
out.println("<!DOCTYPE HTML> " + "<html><head> <title>Set Cookie </title>
</head> " + "<body><center>" + "<h2>Cookie has been set successfully!</h2><br/>"
+ "<a href='index.html'>Click here to go back to previous page </a>" + "</body>
</html>");
}
 }


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

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

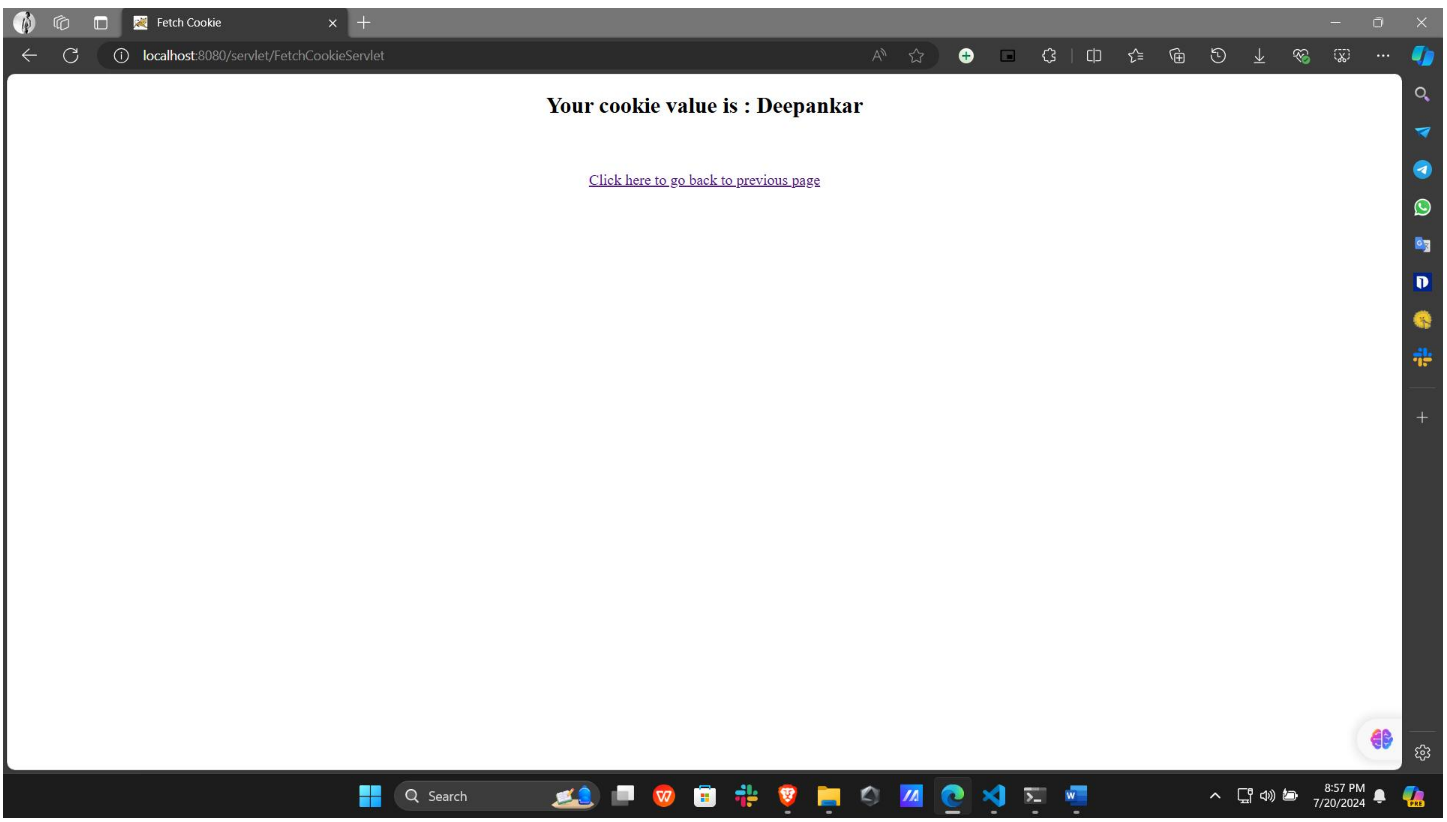
```
public class Practical_10_FetchCookieServlet extends HttpServlet {
Cookie ck = null;
Cookie[] ckary = null;

private String searchCookie(String s) {
for (Cookie temp  ckary) {
ck = temp;
if ((ck.getName()).compareTo(s) == 0)
return ck.getValue();
            }
return "";
        }

@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
ckary = request.getCookies();
response.setContentType("text/html");
PrintWriter out;
out = response.getWriter();
out.println("<!DOCTYPE HTML><html> <head> <title> Fetch Cookie </title>
</head><body><center>"); if (ckary  =!null) { out.println("<h2>Your cookie
value is  :" + searchCookie("studname") + " </h2><br/>");   else { out.println
(h2>No Cookies found!</h2>");   out.println("<br/> " + "<a href='index.html'>Click
here to go back to previous page</a> " + "</center></body></html>");
 }
 }
```

Output:




Practical 09: Servlets
student name:
enrollment number:
Submit
Practical 10: Use of Cookies
Creating Cookie
Student Name : Deepankar
Create Cookie
Fetch and display Cookie Information
Display Cookie Info


Set Cookie
localhost:8080/servlet/SetCookieServlet
Cookie has been set successfully!
Click here to go back to previous page
8:57 PM
7/20/2024

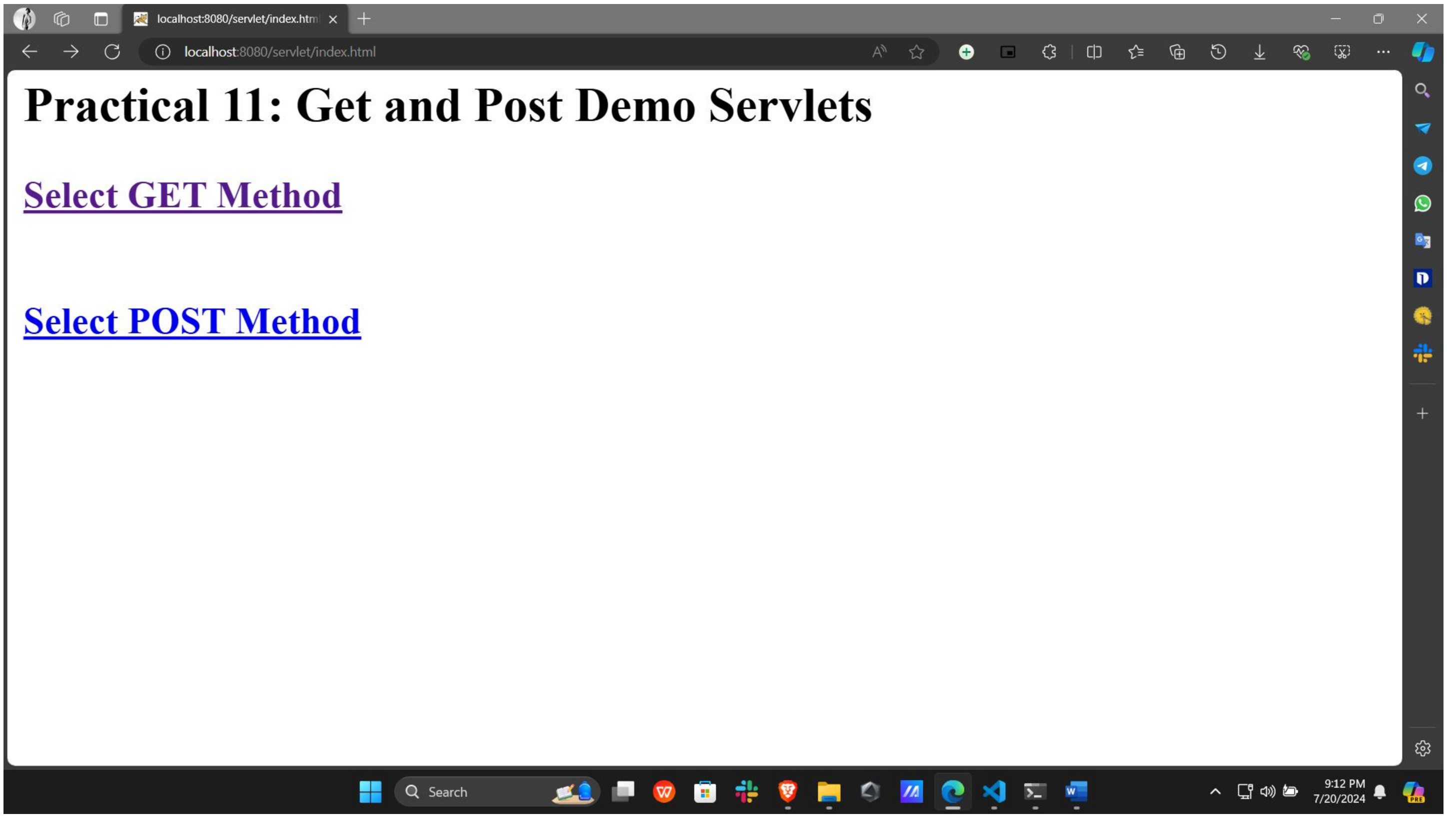

Fetch Cookie
localhost:8080/servlet/FetchCookieServlet
Your cookie value is : Deepankar
Click here to go back to previous page
8:57 PM
7/20/2024

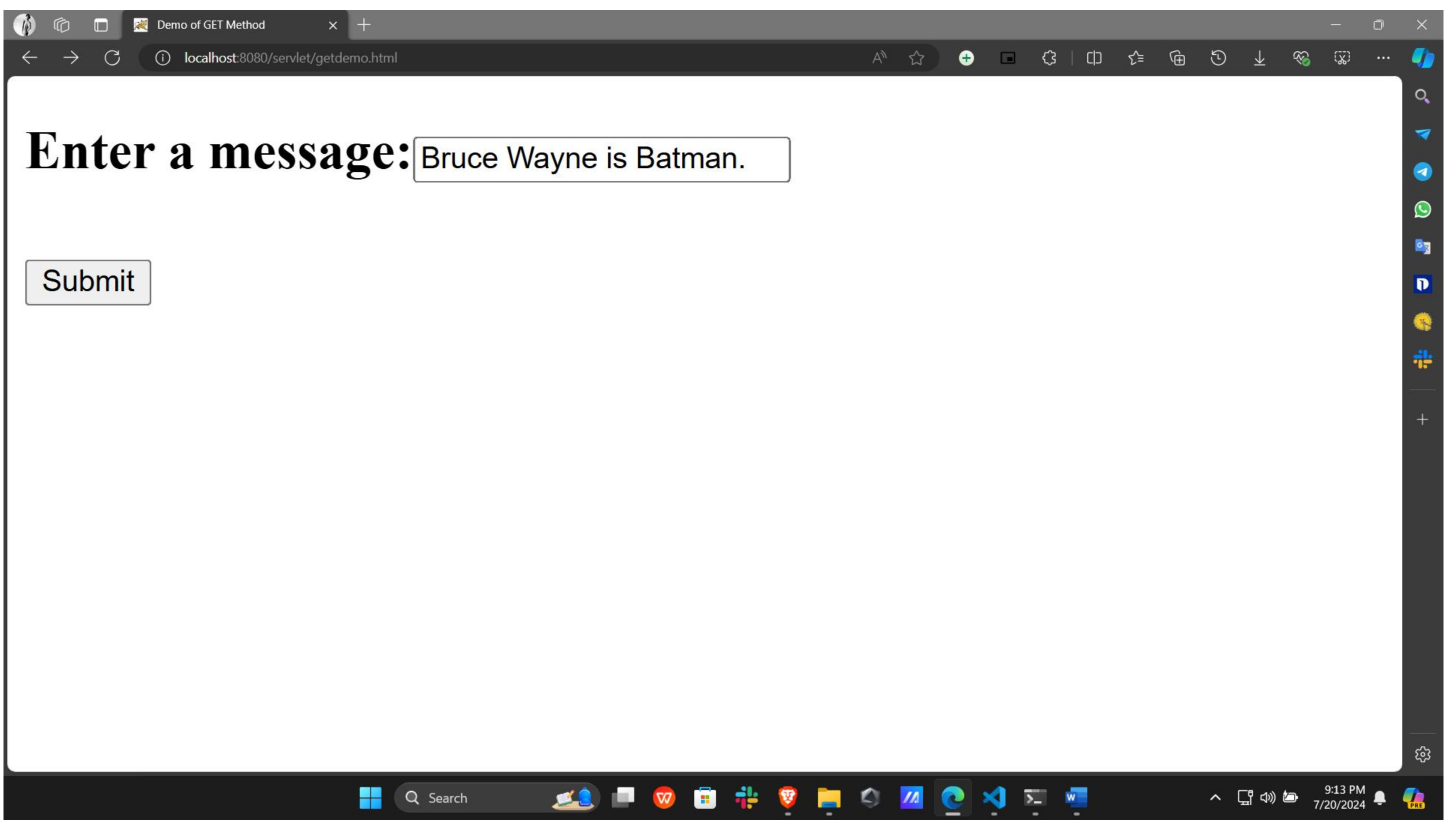
Program 11:

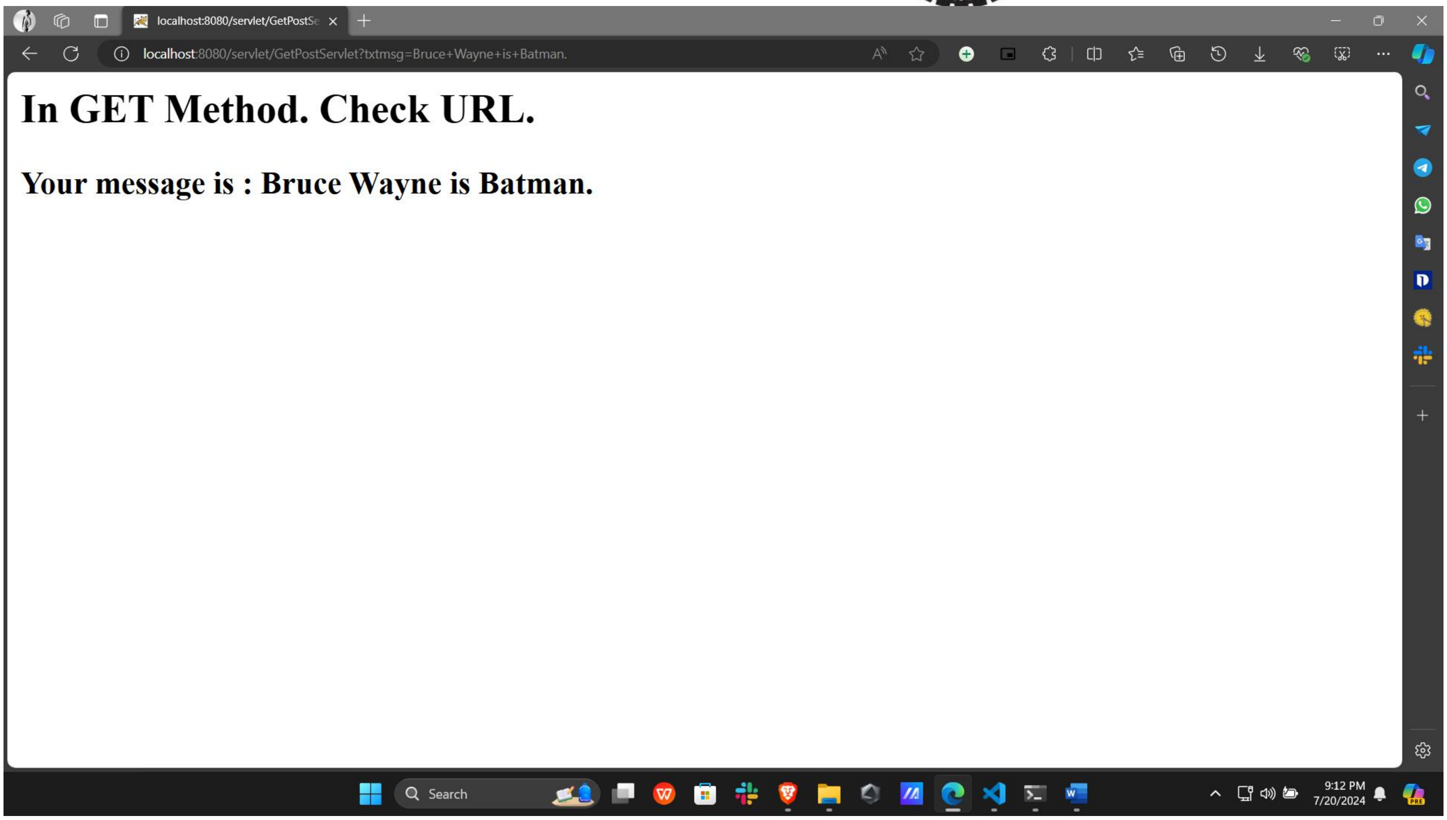
Write a Java Servlet program to demonstrate the use of GET and POST methods for handling HTTP client requests and server responses.

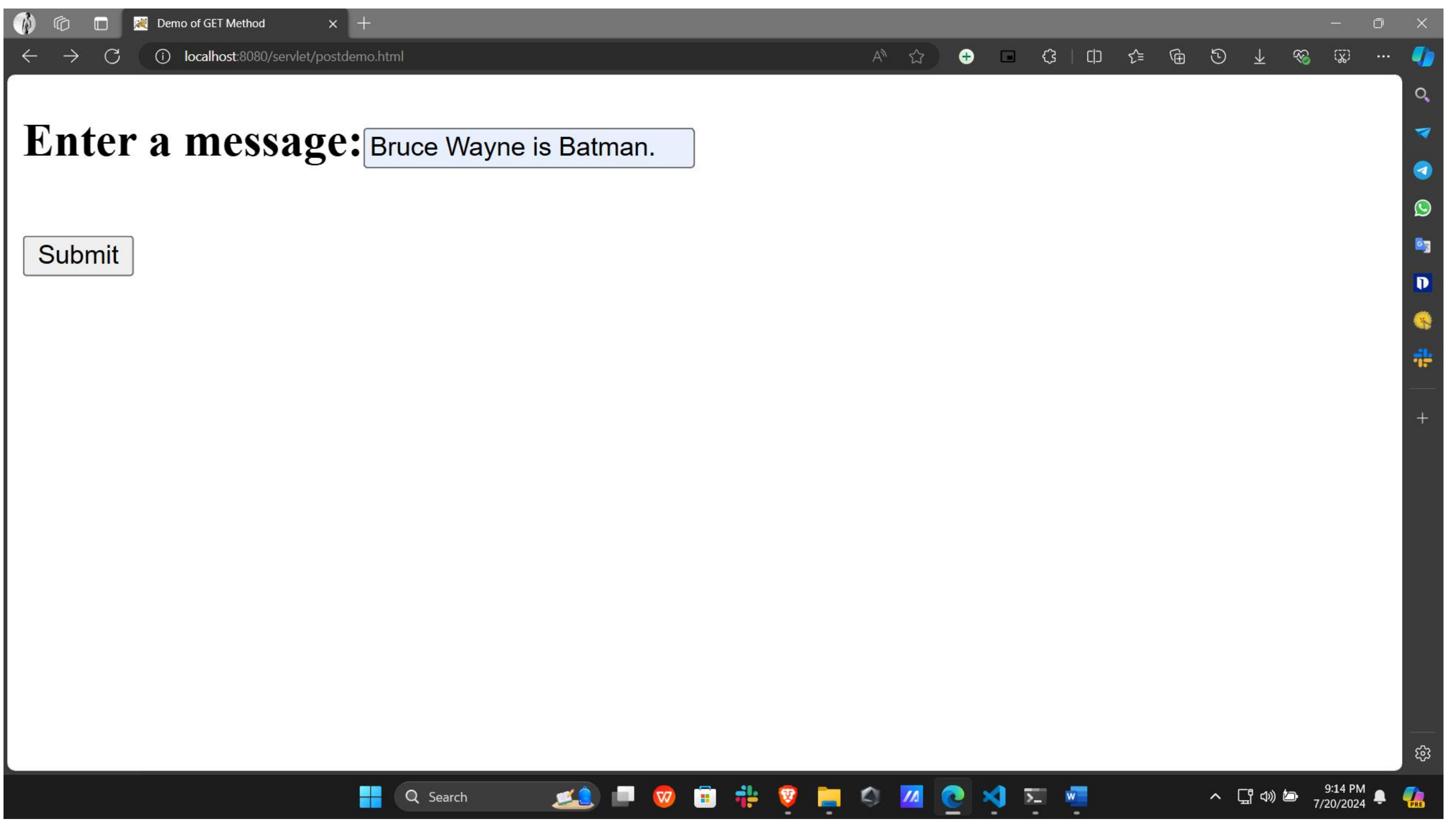
```java
import java.io.IOException;
import java.io.PrintWriter;

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

public class Practical_11_GetPostServlet extends HttpServlet {
@Override
protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
                PrintWriter out = response.getWriter();
                response.setContentType("text/html");
               String code = request.getParameter("txtmsg");
              out.println("<h2> In GET Method. Check URL.</h2>");
               out.println("<h3> Your message is : " + code + "</h3>");
}


 @Override protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
      PrintWriter out = response.getWriter();
     response.setContentType("text/html");
      String code = request.getParameter("txtmsg");
        out.println("<h2> In POST Method. Check URL.</h2>");
       out.println("<h3> Your message is : " + code + "</h3>");
}}
```

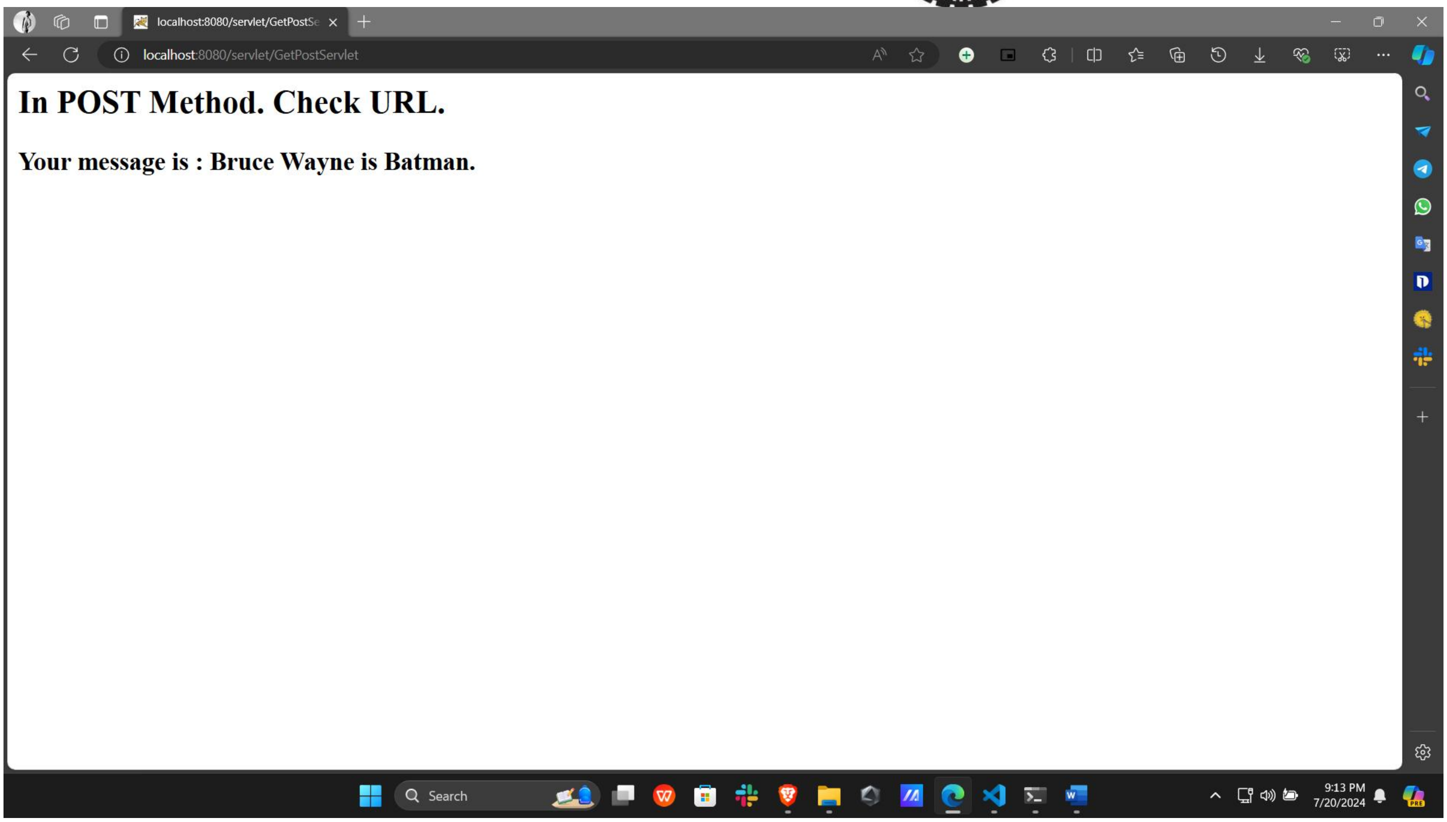
Output:


localhost:8080/servlet/index.html
Practical 11: Get and Post Demo Servlets
Select GET Method
Select POST Method


Demo of GET Method
localhost:8080/servlet/getdemo.html
Enter a message:
Bruce Wayne is Batman.
Submit


localhost:8080/servlet/GetPostSe
localhost:8080/servlet/GetPostServlet?txtmsg=Bruce+Wayne+is+Batman.
In GET Method. Check URL.
Your message is : Bruce Wayne is Batman.
Search
9:12 PM
7/20/2024


Demo of GET Method
localhost:8080/servlet/postdemo.html
Enter a message:
Bruce Wayne is Batman.
Submit
Search
9:14 PM
7/20/2024

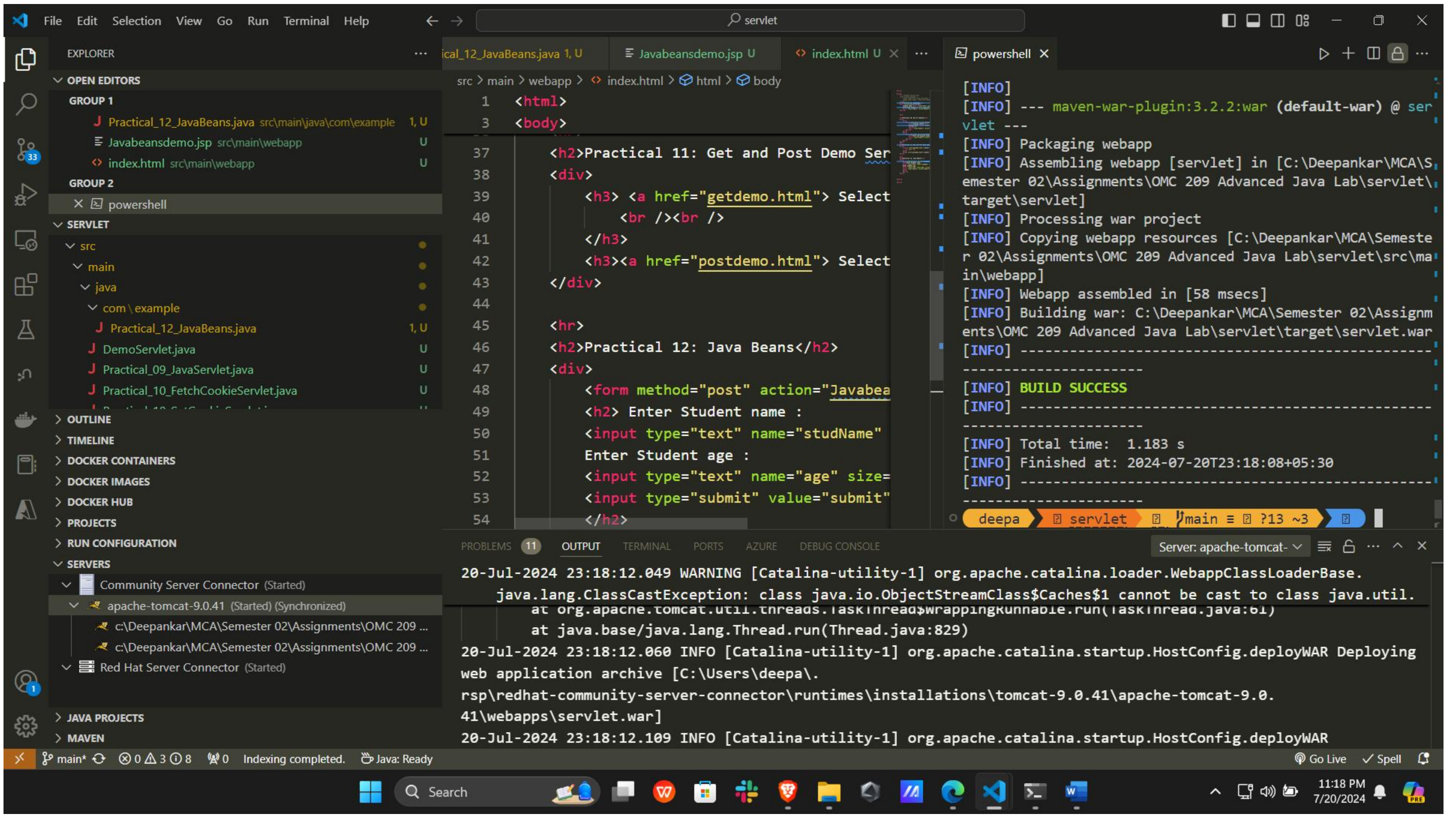

localhost:8080/servlet/GetPostServlet
In POST Method. Check URL.
Your message is : Bruce Wayne is Batman.
Search
9:13 PM
7/20/2024

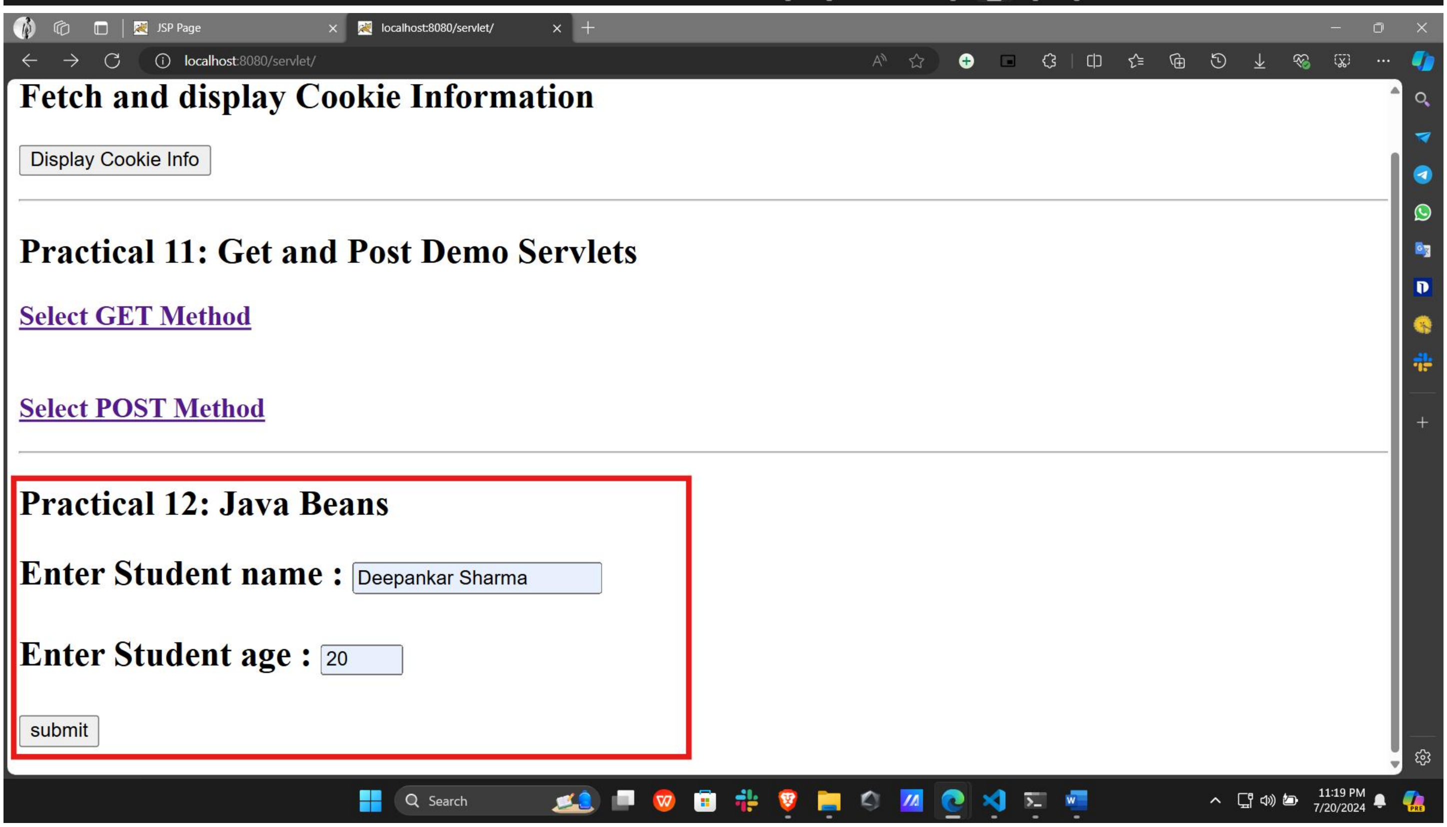
Program 12:

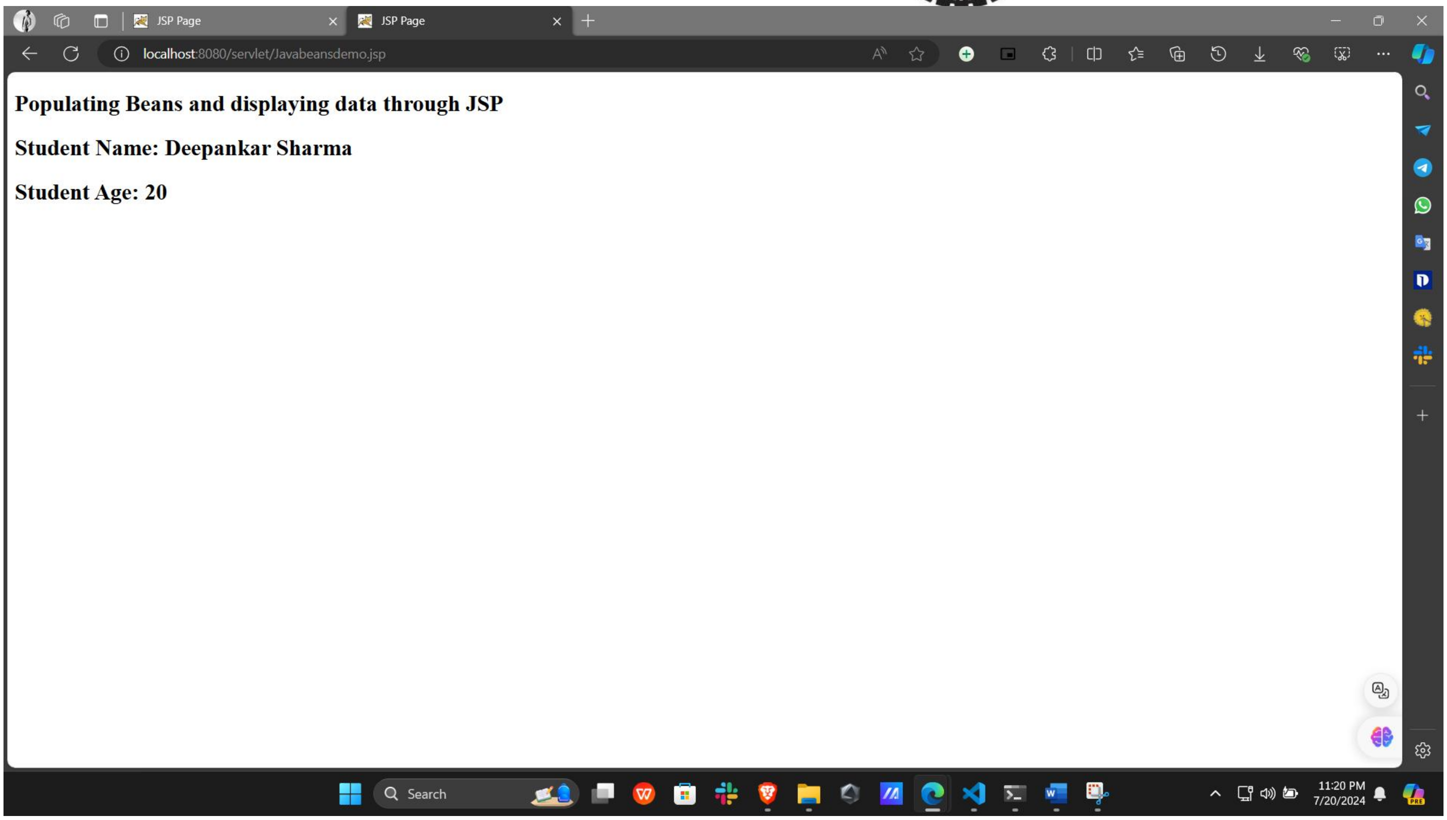
Write a JSP program to demonstrate the use of Java Beans.

```
<!DOCTYPE html>
<html>
 <head>
</head>
 <body>
        <h2>Populating Beans and displaying data through JSP </h2>
      <jsp:useBean id="myid" class="com.example.Practical_12_JavaBeans"
scope="request">
       <jsp:setProperty name="myid" property="studName" />
     <jsp:setProperty name="myid" property="age" />
      </jsp:useBean> <p>
     <h2>Student Name: <jsp:getProperty name="myid"
property="studName" />

      </h2> </p> <p> <h2>Student Age: <jsp:getProperty name="myid"
property="age" /> </h2> </p>
</body> </html>

package com.example;
public class Practical_12_JavaBeans {
String studName, age;
public void Practical_12_JavaBeans(){}

public String getStudName() {
return studName;
  }

public void setStudName(String studName) {
this.studName = studName;
  }
public String getAge() {return age;}
public void setAge(String age) {
this.age = age;
 } }
```

Output:


File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
GROUP 1
Practical_12_JavaBeans.java
Javabeansdemo.jsp
index.html
GROUP 2
powershell
SERVLET
src
main
java
com \ example
Practical_12_JavaBeans.java
DemoServlet.java
Practical_09_JavaServlet.java
Practical_10_FetchCookieServlet.java
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
PROJECTS
RUN CONFIGURATION
SERVERS
Community Server Connector (Started)
apache-tomcat-9.0.41 (Started) (Synchronized)
Red Hat Server Connector (Started)
JAVA PROJECTS
MAVEN
<h2>Practical 11: Get and Post Demo Ser
<h3> <a href="getdemo.html"> Select
<h3><a href="postdemo.html"> Select
<h2>Practical 12: Java Beans</h2>
<form method="post" action="Javabea
<h2> Enter Student name :
<input type="text" name="studName"
Enter Student age :
<input type="text" name="age" size=
<input type="submit" value="submit"
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ servlet ---
[INFO] Packaging webapp
[INFO] Processing war project
[INFO] Webapp assembled in [58 msecs]
[INFO] BUILD SUCCESS
[INFO] Total time: 1.183 s
[INFO] Finished at: 2024-07-20T23:18:08+05:30
20-Jul-2024 23:18:12.049 WARNING [Catalina-utility-1] org.apache.catalina.loader.WebappClassLoaderBase.
java.lang.ClassCastException: class java.io.ObjectStreamClass$Caches$1 cannot be cast to class java.util.
at java.base/java.lang.Thread.run(Thread.java:829)
20-Jul-2024 23:18:12.060 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive [C:\Users\deepa\.rsp\redhat-community-server-connector\runtimes\installations\tomcat-9.0.41\apache-tomcat-9.0.41\webapps\servlet.war]
20-Jul-2024 23:18:12.109 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.deployWAR
11:18 PM
7/20/2024


JSP Page
localhost:8080/servlet/
Fetch and display Cookie Information
Display Cookie Info
Practical 11: Get and Post Demo Servlets
Select GET Method
Select POST Method
Practical 12: Java Beans
Enter Student name : Deepankar Sharma
Enter Student age : 20
submit
11:19 PM
7/20/2024


JSP Page
JSP Page
localhost:8080/servlet/Javabeansdemo.jsp
Populating Beans and displaying data through JSP
Student Name: Deepankar Sharma
Student Age: 20
Search
11:20 PM
7/20/2024