Web Development and Core Java

Lab Manual

Vth Semester

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

Prepared By:
Kuldeep Yadav
Assistant Professor,
Department of Computer Science and Engineering,
RPS College of Engineering, Mahendergarh
<table>
<thead>
<tr>
<th>S.no.</th>
<th>Particular</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>General Instructions</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>MDU Syllabus</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>List of Experiments</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Hardware and Software requirement</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Experiment-1</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>Experiment-2</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>Experiment-3</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>Experiment-4</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Experiment-5</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Experiment-6</td>
<td>23</td>
</tr>
<tr>
<td>11</td>
<td>Experiment-7</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>Experiment-8</td>
<td>26</td>
</tr>
<tr>
<td>13</td>
<td>Experiment-9</td>
<td>28</td>
</tr>
<tr>
<td>14</td>
<td>Experiment-10</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>Experiment-11</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>Experiment-12</td>
<td>54</td>
</tr>
</tbody>
</table>
GENERAL INSTRUCTIONS FOR LABORATORY CLASSES

DO’S

➢ Without Prior permission do not enter into the Laboratory.

➢ Students should sign in the LOGIN REGISTER before entering into the laboratory.

➢ Students should come with observation and record note book to the laboratory.

➢ Students should maintain silence inside the laboratory.

➢ After completing the laboratory exercise, make sure to shutdown the system properly.

DONT’S

➢ Students bringing the bags inside the laboratory..

➢ Students using the computers in an improper way.

➢ Students scribbling on the desk and mishandling the chairs.

➢ Students using mobile phones inside the laboratory.

➢ Students making noise inside the laboratory.
Syllabus

CSE-311 Web Development & Core JAVA
Java programs using classes & objects and various control constructs such as loops etc, and data structures such as arrays, structures and functions.
Java programs for creating Applets for display of Images, Texts and Animation
Programs related to interfaces & packages
Input output & Random files programs in java
Java programs using Event driven concept
Programs related to Network Programming
Development of Web site for the college or newspaper agency.
List of Experiment

1. Write a program to print “Hello World” in java.
2. Write a program to implement basic data types and control structure (loop, if-else etc) in java.
3. Write a program to implement array, polymorphism, inheritance using methods in java.
4. Write a program to implement packages and interface in java.
5. Write a program to draw an image using applet in Java.
6. Write a program to read and write data in a file using java.
7. Create a webpage using HTML to describe your department using paragraph and list.
8. Create a table in HTML to show your class time table.
9. Apply CSS(Cascade Style Sheet) to change a certain portion, Bold, Italic and underline certain words in your HTML web page.
10. Create a registration form and put validation checks on values entered by the users using java scripts.
11. Create a JSP page for the form which embedded JSP in HTML.
12. Using idea from the above experiments try to create a website for your college (RPS).
Hardware and Software requirement

HARDWARE REQUIREMENTS:

Pentium Dual Core @ 2.70 Ghz and above with 1GB RAM,

160 GB HARD Disk,

Monitor 1024* 768 color

SOFTWARE REQUIREMENTS:

Windows / Linux operating system

JDK 1.6 (or above)

Netbeans IDE or Eclipse IDE

Oracle or MS-Access or MySQL
Experiment-1

**Aim:** Write a program to print “Hello World” in java

**Source Code:**

```java
Class First {

    public static void main (String arg [])
    {
        System.out.print ("Hello World");
    }
}
```

**Output:**

```
C:\java\javac First.java
C:\java\java First.
Hello World
```
Viva Questions:

1. What are command line arguments?
   The values that are passed to the main method from the command line while executing the program are called as command line arguments.

2. What are the various types of operators available in java?
   Arithmetic operator, Relational operator, Logical operator, Bitwise operator, Increment and decrement operator, Assignment operator, Conditional operator and Special operator.

3. What is a ternary operator?
   The operator that takes three arguments is called as ternary operator.

   The conditional operator is the only ternary operator available in java.

4. What is the use of Integer.parseInt() method?
   This method is used to convert the String object into integer value.

5. What is called as a Boolean expression?
   An expression that returns either true or false value is called a Boolean expression.
Experiment-2

**Aim:** Write a program to implement basic data types and control structure (loop, if-else etc ) in java

**Source Code:**

```java
public class Test {
    public static void main(String args[]) {
        for(int x = 1; x < 10; x = x+1) {
            System.out.print("value of x : "+ x);
            System.out.print("\n");
        }
    }
}
```

**Output:** value of x 1
value of x :2
value of x :3
value of x:4
value of x :5
value of x:6
value of x:7
value of x :8
value of x:9
value of x:10
Viva Questions:

1. What is a control structure?

Control structures are statements that are used to change the flow of the program based on some condition.

2. What are the two types of control structures?

Decision making statements and Looping statements

3. What are decision making statements?

The statements that are used to execute only a block of code and leaving others based on the condition.

4. What is the use of looping statement?

The looping statement is used to execute a block of repeatedly until the condition is true.

5. What is the difference between while and do..while?

In case of while statement the block of code will not be executed at least once if the condition is false at the first run.

In case of do..while statement the block of code will be executed at least once if the condition is false at the first run.
**Experiment-3**

**Aim:** Write a program to implement array, polymorphism, inheritance using methods in java

**Source Code:**

```java
class Array {
    public static void main(String args[]) {
        int a[] = new int[4]; // declaration and instantiation
        a[0] = 1; // initialization
        a[1] = 2;
        a[2] = 3;
        a[3] = 4;

        // printing array
        for (int i = 0; i < a.length; i++) // length is the property of array
            System.out.println(a[i]);
    }
}

(ii). Polymorphism

Public class Car {
    void run() {
        System.out.println("running");
    }
}

class Swift extends Bike {
    void run() {
        System.out.println("running safely with 80km");
    }

    public static void main(String args[]) {
        Car c = new Swift(); // upcasting
        c.run();
    }
}
```

Output: running safely with 80km
(iii). Inheritance

class Employee{
    float salary=40000;
}

class Programmer extends Employee{
    int bonus=10000;
    public static void main(String args[])
    {
        Programmer p=new Programmer();
        System.out.println("Programmer salary is:"+p.salary);
        System.out.println("Bonus of Programmer is:"+p.bonus);
    }
}

Output:

Programmer salary is:40000

Bonus of Programmer is: 10000
Viva Questions:

1. **What is an array?**
   
   An array is a collection of elements of same data type referred by a common name. The elements are of the array are stored in consecutive memory locations.

2. **What are the types of arrays?**
   
   One dimensional array, two dimensional array and multidimensional arrays.

3. **How to declare a two dimensional array?**
   
   Datatype arrayname[][]=new datatype[ row size][column size]

4. **How the individual elements of an array can be accessed?**
   
   The individual elements can be accessed using the index. The index of the first element starts with 0.

5. **What is Vector?**
   
   Vector is just like an array that can store elements of different data types. The elements of the vector can only be objects. So the primitive data type must be converted into object data type before adding to the vector.

6. **What is the difference between capacity and size of the Vector?**
   
   Capacity specifies the maximum number of objects that can be stored in the vector. Size specifies the number of objects that are present in the Vector.

7. **What are the difference between array and Vector?**
   
   The elements of the array are of same data type. The elements of the Vector can be different data type.
   The elements of the array can be of primary data type. The elements of the Vector can only be objects.
The capacity of the array is fixed. The size of Vector can be changed during Run time.

8. **What is class?**

A class is a collection of data and methods that defines an object. The variables and methods (functions) of a class are called members of the class.

9. **What is an object?**

The variable of type class is called object.

**Syntax for defining a class**

```java
class className
{
    //Declaration of instance variables
    //Constructors
    //Instance Methods
}
```

10. **How can we create objects?**

The objects can be created using the new operator.

```java
className objectNname=new className();
```

11. **How the members of a class can be accessed?**

The members of the class can be accessed using the dot operator

```java
ObjectName .variable
```

Or

```java
ObjectName.methodName(Arguments)
```

12. **What is a constructor?**

Constructors are special methods whose name is same as the class name. The
constructors do not return any value.

The constructors are automatically called when an object is created. They are usually used to initialize the member variables.

13. What is a default constructor?

Constructor that does not take any argument is called default constructor.

14. What is meant by constructor overloading?

Defining more than one constructor for a class which differ in the number of arguments / type of arguments or both is called constructor overloading.

15. What is inheritance?

Inheritance is the process of deriving a new class from an existing class. The newly created class is called sub class and the already existing class is called super class.

16. What are the types of inheritance?

Single inheritance: One super class and single sub class.

Multiple inheritance: More than one super class and single subclass.

Hierarchial inheritance: one super class and more than one subclass.

Multilevel inheritance: Deriving a sub class from another sub class

Syntax for deriving a sub class

class subclassName extends superclassname

{
   //define the members

}
**Experiment-4**

**Aim:** Write a program to implement packages and interface in java.

Source code:

```java
package mypack;
public class Test{
    public static void main(String args[]){
        System.out.println("Welcome to First package");
    }
}
```

Save Test.java

Compile: Syntax : javac -d directory javafilename

Eg. javac –d.Test.java

Run: java mypack.Test

Output: Welcome to First package

(ii)

```java
package my;
public class A{
    void show()
    {
        System.out.print("Hello");
    }
}
```

Compile using javac –d A.java
How to use package in another class

import my.*;

class B{
    public static void main(String args[]){
        A obj = new A();
        obj.msg();
    }
}

Output : javac B.java

    java B
    Hello

(iii).
interface printable{
    void print();
}

class A implements printable{
    public void print(){System.out.println("Hello");}
}

public static void main(String args[]){
    A obj = new A();
    obj.print();
}

Output: Hello

Viva Questions:
1. **What is an interface?**

   Interface is just like a class which contains final variables and public abstract methods. It is used to implement multiple inheritances in java.

2. **What is an abstract method?**

   The method which has only declaration in the super class and defined in the subclass is known as abstract method.

   **Syntax for defining an interface**

   ```java
   interface interfacename
   {
       //define the static variables and declare abstract methods
   }
   ```

3. **How to implement interface?**

   The abstract methods should be implemented in a class to use the interface in our program.

   ```java
   class className extends superclassname implements interface1, interface2, ...
   {
       //define the members
   }
   ```

4. **What is package?**
Package is a collection of interfaces and classes.

5. **How a package can be created?**

   A package can be created using the package statement.

6. **How a package can be imported into the program?**

   A package can be imported using the import statement.
**Experiment-5**

**Aim:** Write a program in java to draw image using applet.

To execute the applet by appletviewer tool, create an applet that contains applet tag in comment and compile it. After that run it by: appletviewer First.java. Now Html file is not required but it is for testing purpose only.

```java
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet{

    public void paint(Graphics g){
        g.drawRect(10,10,50,100);
        g.drawRect(10,10,50,100);
    }

} /*
<applet code="First. class" width="300" height="300">
</applet> */
```

**Output:**

To execute the applet by appletviewer tool, write in command prompt:

```
c:\>javac First.java
c:\>appletviewer First.java
```
Viva Questions:

1. What is an applet?
   Applet is a small interactive java programs that are used for web application.

2. What are the two packages that are needed to create an applet?
   Java.awt and java.applet

3. What is an applet tag?
   Applet tag is an HTML tag that is used to run the java applets.

4. What is an event?
   Events is an interruption given to the running program using input devices such as mouse, keyboard etc.

5. What package is needed to handle an event?
   java.awt.event

6. What are the steps needed to handle the event?
   1. Import java.awt.event
   2. Implement the corresponding Event Listener interfaces
   3. Register the Event Listeners with the event source.
   4. Override the appropriate methods of the listener interfaces.
**Experiment-6**

**Aim**: Write a program to read and write data in a file using java

**Source Code:**

```java
import java.io.*;

class Test{
    public static void main(String args[]){
        try{
            FileOutputStream fout=new FileOutputStream("abc.txt");
            String s="Anup Kumar is a good player";
            byte b[]=s.getBytes();//converting string into byte array
            fout.write(b);
            fout.close();
            System.out.println("File created ...");
        }catch(Exception e){System.out.println(e);}
    }
}

import java.io.*;
class SimpleRead{
    public static void main(String args[]){
        try{
            FileInputStream fin=new FileInputStream("abc.txt");
            int i=0;
            while((i=fin.read())!=-1){
                System.out.println((char)i);
            }
            fin.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

Output: Anup Kumar is a good player
Experiment-7

Aim: Create a webpage using HTML to describe your department using paragraph and list.

Source Code:

<html>
<body>
<center> <b><u>Department of Computer Science and Engineering</u></b></center>
<p>The fundamental aim of faculty is to provide academic, professional and research support for students by designing and developing educational programmes that contribute the effective learning and critical thinking with theoretical and practical knowledge about sustainable developing field of Computer Science and Engineering.</p>
<p>The discipline is one of the most leading and demanding professional careers in today's modern era. It involves designs, supports and operations of computer software / hardware systems. Here the Computer Science programme of RPS Group of Institutions provides a balanced blend of software and hardware learning experiences, founded on solid mathematics and science with a definite engineering flavour, and hands on experience with the latest equipments in our laboratories. The department also highlights diversified areas of computer science such as programming, algorithms, hardware, software, testing, networking and simulation ranging from elementary to upper intermediate levels. For practical orientation, highly advanced research laboratories have been installed with the latest configured computer machines.</p>
<p>Some of Our Faculty Members:</p>
<ol>
<li>Mr. Mahesh Kumar</li>
<li>Mr. Jitender</li>
<li>Mr. Mukesh Kumar</li>
<li>Mr. Kuldeep Yadav</li>
</ol>
Approved by AICTE (Govt. of India) & Affiliated to M.D. University, Rohtak

/ol>

</html>

Output:

The fundamental aim of faculty is to provide academic, professional and research support for students for designing and developing educational programmes that contribute to effective learning and critical thinking with theoretical and practical knowledge about sustainable development of Computer Science and Engineering.

The discipline is one of the most leading and demanding professional careers in today’s modern era. It involves designing, supports and operations of computer software / hardware systems. Here the Department of Computer Science and Engineering provides a balanced blend of software and hardware training experiences, grounded on solid mathematics and science with a definite engineering flavour, and hands-on experience with the latest equipment in our laboratories. The department also highlights diversified areas of computer science such as programming, algorithms, hardware, software, testing, networking and simulation ranging from elementary to upper intermediate levels. For practical orientation, highly advanced research laboratories have been installed with the latest configured computer machines.

Some of Our Faculty Members:
1. Mr. Mahesh Kumar
2. Mr. Shinde
3. Mr. Mukesh Kumar
4. Mr. Kuldip Yadav
Experiment-8

Aim: Create a table in HTML to show your class time table.

Source Code:

```html
<html>
<body>
<center><b><u>Time Table</u></b></center><br/>
<center><table border="1">
<tr><th>Day - Time</th><th>9:00 - 10:00</th><th>10:00 - 11:00</th><th>11:00 - 12:00</th><th>12:30 - 1:30</th><th>1:30 - 2:30</th><th>2:30 - 3:30</th></tr>
<tr><td><b>Monday</b></td><td>CAO</td><td>C - Programming</td><td>-</td><td>WD</td><td>DIS</td><td>-</td></tr>
<tr><td><b>Tuesday</b></td><td>-</td><td>C - Programming</td><td>-</td><td>CAO</td><td>-</td><td>-</td></tr>
<tr><td><b>Wednesday</b></td><td>CAO</td><td>C - Programming</td><td>-</td><td>SWD</td><td>-</td><td>-</td></tr>
<tr><td><b>Thrusday</b></td><td>-</td><td>C - Programming</td><td>DIS</td><td>CAO</td><td>-</td><td>WD</td></tr>
<tr><td><b>Friday</b></td><td>-</td><td>C - Programming</td><td>DIS</td><td>-</td><td>WD</td><td>WD Lab</td></tr>
</table></center>
</html>
```
## Time Table

<table>
<thead>
<tr>
<th>Day-Time</th>
<th>9:00-10:00</th>
<th>10:00-11:00</th>
<th>11:00-12:00</th>
<th>12:30-1:30</th>
<th>1:30-2:30</th>
<th>2:30-3:30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>CAO</td>
<td>C-Programming</td>
<td>-</td>
<td>WD</td>
<td>DIS</td>
<td>-</td>
</tr>
<tr>
<td>Tuesday</td>
<td>-</td>
<td>C-Programming</td>
<td>-</td>
<td>CAO</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Wednesday</td>
<td>CAO</td>
<td>C-Programming</td>
<td>-</td>
<td>SWD</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Thursday</td>
<td>-</td>
<td>C-Programming</td>
<td>DIS</td>
<td>CAO</td>
<td>-</td>
<td>WD</td>
</tr>
<tr>
<td>Friday</td>
<td>-</td>
<td>C-Programming</td>
<td>DIS</td>
<td>-</td>
<td>WD</td>
<td>WD Lab</td>
</tr>
</tbody>
</table>
Experiment-9

Aim: Apply CSS (Cascade Style Sheet) to change a certain portion, Bold, Italic, and underline certain words in your HTML web page.

Source: Style.css

body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}

.stylebig {
    font-size: 18px;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}

.stylemedium {
    font-size: 14px;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}

.stylesmall {
    font-family: "Times New Roman", Times, serif;
Home.Html

<%@ page contentType="text/html; charset=iso-8859-1" language="java" errorPage="ErrorPage.jsp" %>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" href="stylesheet/Style.css" type="text/css"/>
<title>Home Page</title>
</head>
<body>

</body>
</html>
**Experiment-10**

**Aim:** Create a registration form and put validation checks on values entered by the users using java scripts.

**Source Code:**

**StudentLogin.jsp**

```html
<%@ page contentType="text/html; charset=iso-8859-1" language="java" errorPage="ErrorPage.jsp" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Login Page</title>
<link rel="stylesheet" href="stylesheet/Style.css" type="text/css"/>
<script type="text/javascript">
function val()
{
    if(document.form1.studentId.value="")
    {
        alert("Please Enter Your Login ID");
        document.form1.studentId.focus();
        return false;
    }
    if(document.form1.studentPassword.value="")
    {
```
alert("Please Enter Your Password");

document.form1.studentPassword.focus();

return false;

}
<td width="30%" align="right"><a href="Student_Registration.jsp" class="stylelink" style="text-decoration:none ; font-weight: bold;">New Student Click Here</a></td>

</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>

<% if(session.getAttribute("message") != null) {
<tr>
<td colspan="4" class="stylegreen" align="center"><%= session.getAttribute("message")%></td>
</tr>
%
session.removeAttribute("message");
%
</tr>
<tr>
<td colspan="4">&nbsp;</td>
</tr>
<tr>
<td colspan="4" align="center"><%= session.getAttribute("message")%></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<table width="40%" border="1" align="center" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
  <tr align="center" bgcolor="#999999">
    <td colspan="2" class="stylebig">Student Login Here</td>
  </tr>
  <tr bgcolor="#E1E1E1" class="stylesmall">
    <td width="35%" class="style7">Login Id: </td>
    <td width="65%"><input name="studentId" type="text" id="studentId"></td>
  </tr>
</table>
<tr bgcolor="#E1E1E1" class="stylesmall">
<td class="style7">Password:</td>
<td><input name="studentPassword" type="password" id="studentPassword"></td>
</tr>

<% if(request.getParameter("valid") != null &
request.getParameter("valid").equals("invalid")) { %>
<tr bgcolor="#E1E1E1">
<td colspan="2" align="center" class="stylered">Invalid Login Id or Password.</td>
</tr>
<% } %>
<tr bgcolor="#E1E1E1">
<td colspan="2" align="center" class="stylered"><input name="login" class="style10" type="submit" id="login" value="Login">
<input name="close" type="button" id="close" class="style10" value="Close" onClick="self.location='Home.jsp'">
</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</tr>
</table>
</tr>
</table>
</tr>
<tr height="5%" align="center">
<%@ include file="Footer.jsp" %></tr>
**Experiment-11**

**Aim:** Create a JSP page for the form which embedded JSP in HTML.

**Source Code:** 1) Create the dynamic web project

For creating a dynamic web project click on File Menu -> New -> dynamic web project -> write your project name e.g. first -> Finish.
2) Create the JSP file in eclipse IDE

For creating a jsp file explore the project by clicking the + icon -> right click on WebContent -> New -> jsp -> write your jsp file name e.g. index -> next -> Finish.
Approved by **AICTE** (Govt. of India) & Affiliated to **M.D. University**, Rohtak
Now JSP file is created, let's write some code.
3) Start the server and deploy the project:

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

If you are using Eclipse IDE first time, you need to configure the tomcat server First. Click for

**How to configure tomcat server in eclipse IDE**
Now start the tomcat server and deploy project

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.
Yes, let's see JSP is successfully running now.
Experiment-12

Aim: Using idea from the above experiments tries to create a website for your college (RPS)