



**UNIVERSITY WITH A PURPOSE**

## **OBJECT ORIENTED PROGRAMMING LAB**

**Submitted By-**

**Name-ROHAN NYATI**

**Roll.No-R177219148**

**SAP ID-500075940**

**Branch- B.Tech CSE AI&ML(2019-2023)**

**Batch-B5**

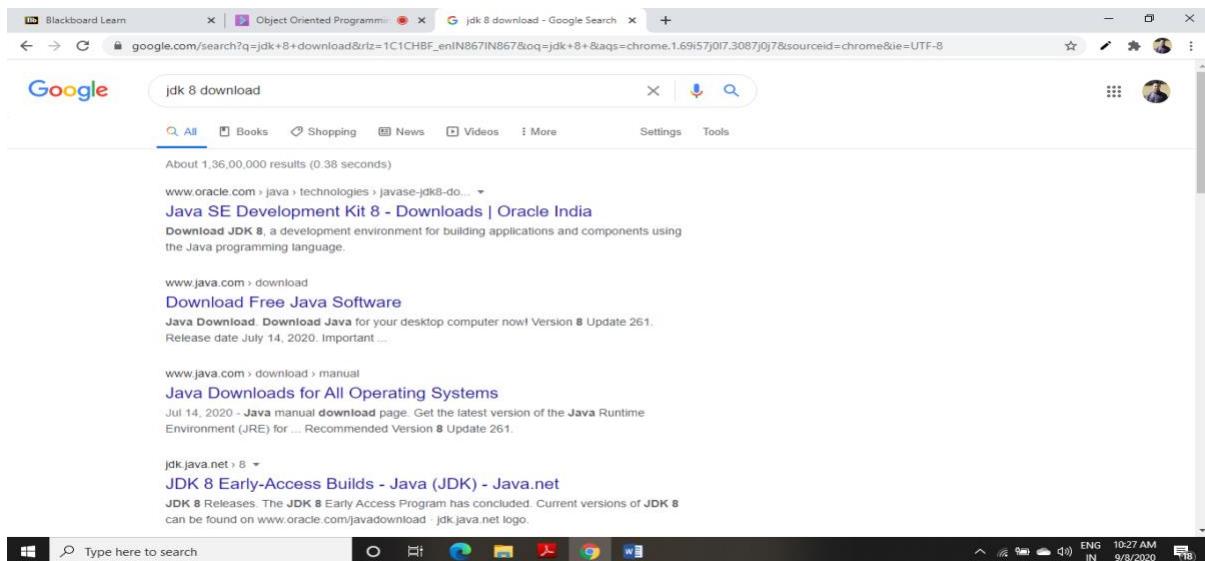
**SEMESTER- 4th**

### **EXPERIMENT NO-1**

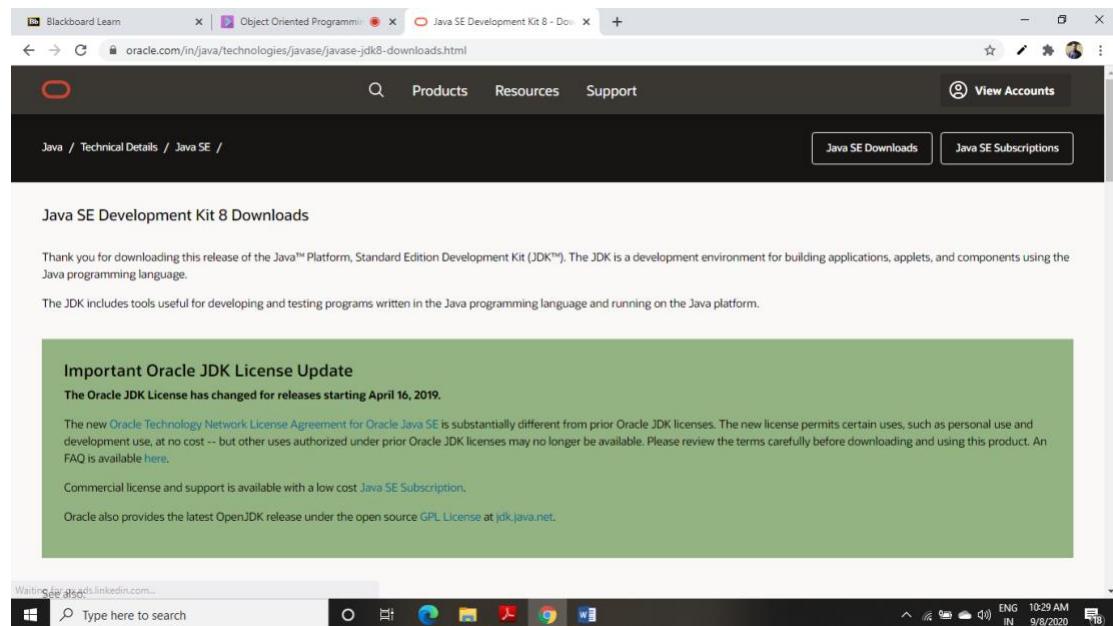
**TITLE:** Introduction to JAVA Environment

**1) Installation of JDK**

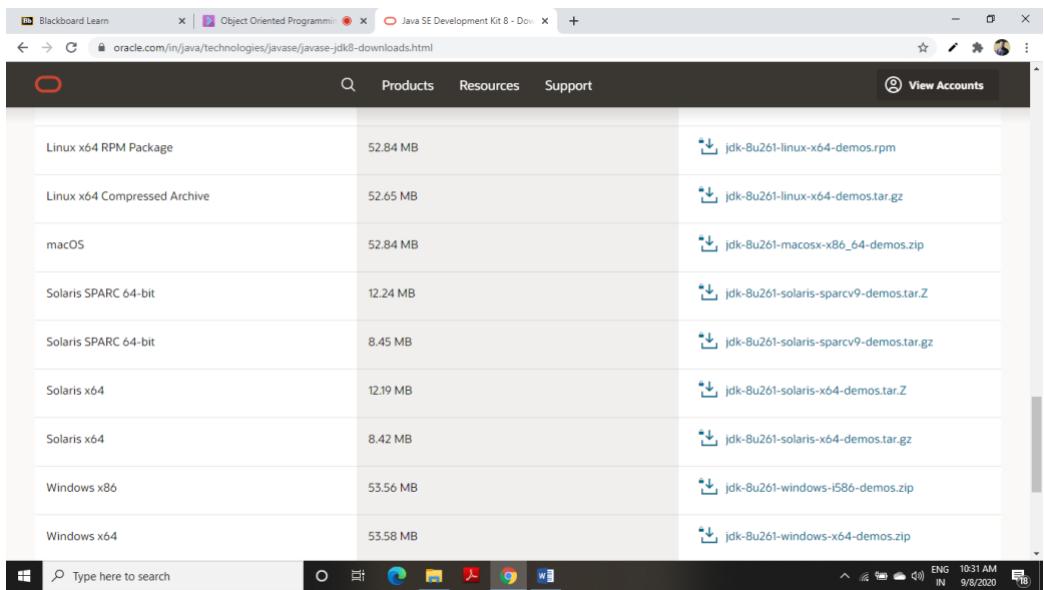
First, we will search **JDK 8 Download** and then we will click the Second link:  
**JAVA SE Development Kit 8 – Downloads**



## 2) A Window will open of JAVA SE Development Kit 8 Downloads

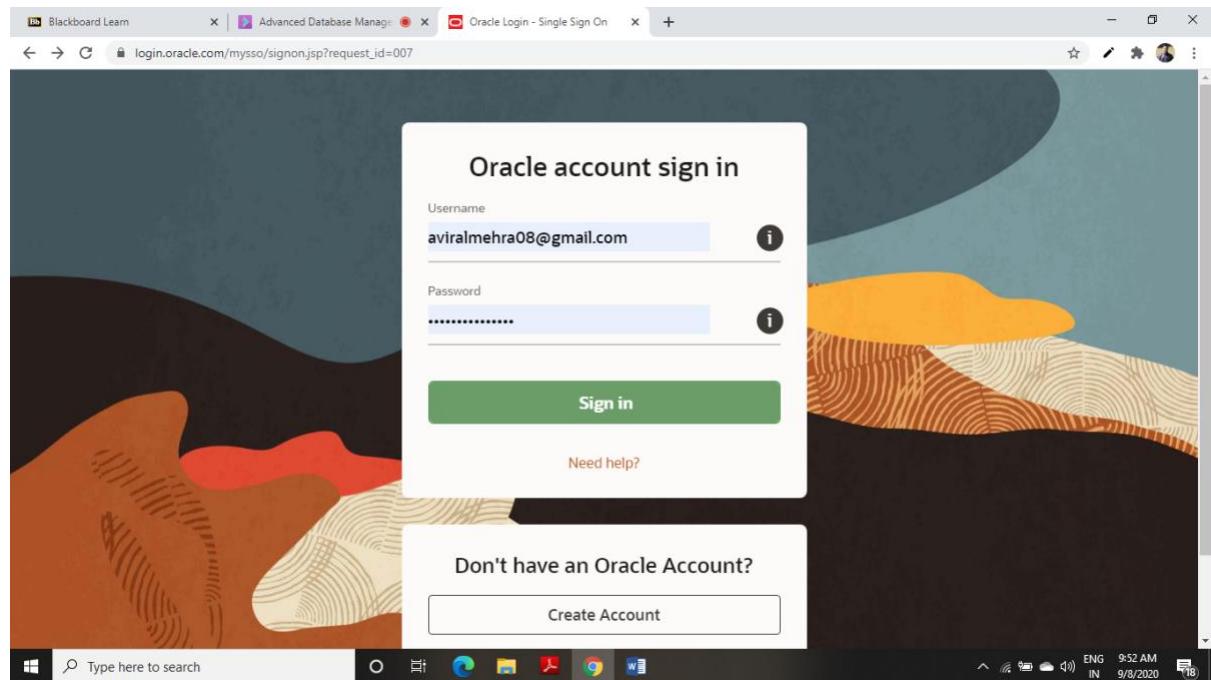


## 3) Now Scroll Down and Select Windows \*64 and Download



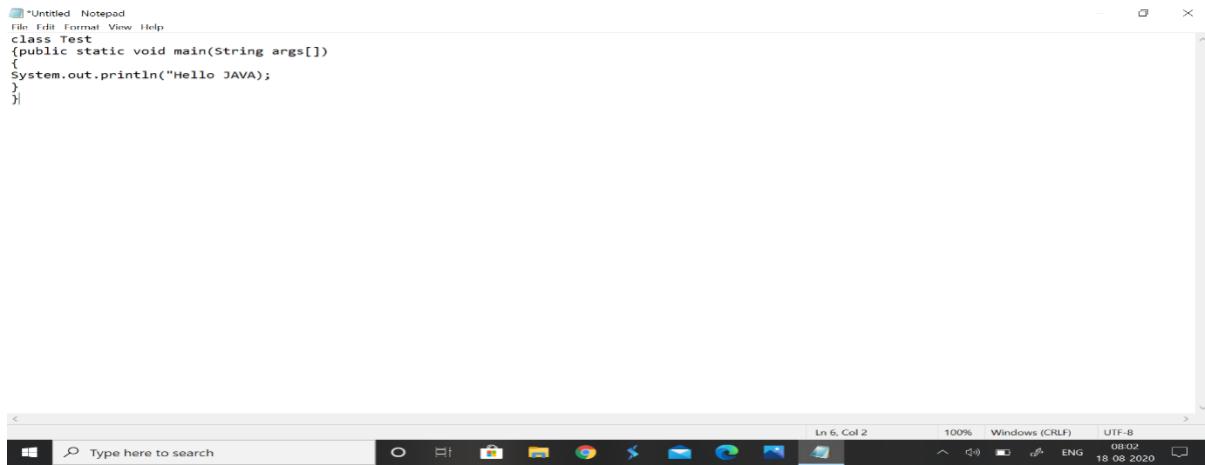
#### 4) After Downloading JDK

Make Account on **ORACLE** and SIGN IN



## 2) Setting of PATH and CLASSPATH

Code a Program in NOTEPAD :



```
Untitled - Notepad
File Edit Format View Help
class Test
{
public static void main(String args[])
{
System.out.println("Hello JAVA");
}
}
```

The screenshot shows a Windows desktop environment. In the foreground, a Notepad window is open with the title "Untitled - Notepad". It contains a single Java class definition: "class Test" with a main method that prints "Hello JAVA" to the console. The Notepad window has standard menu options like File, Edit, Format, View, and Help. The status bar at the bottom indicates "Ln 6, Col 2", "100%", "Windows (CRLF)", and "UTF-8". In the background, the Windows taskbar is visible with icons for Start, Search, Task View, File Explorer, Edge, and other system icons. The date and time "18-08-2020 08:02" are also shown on the taskbar.

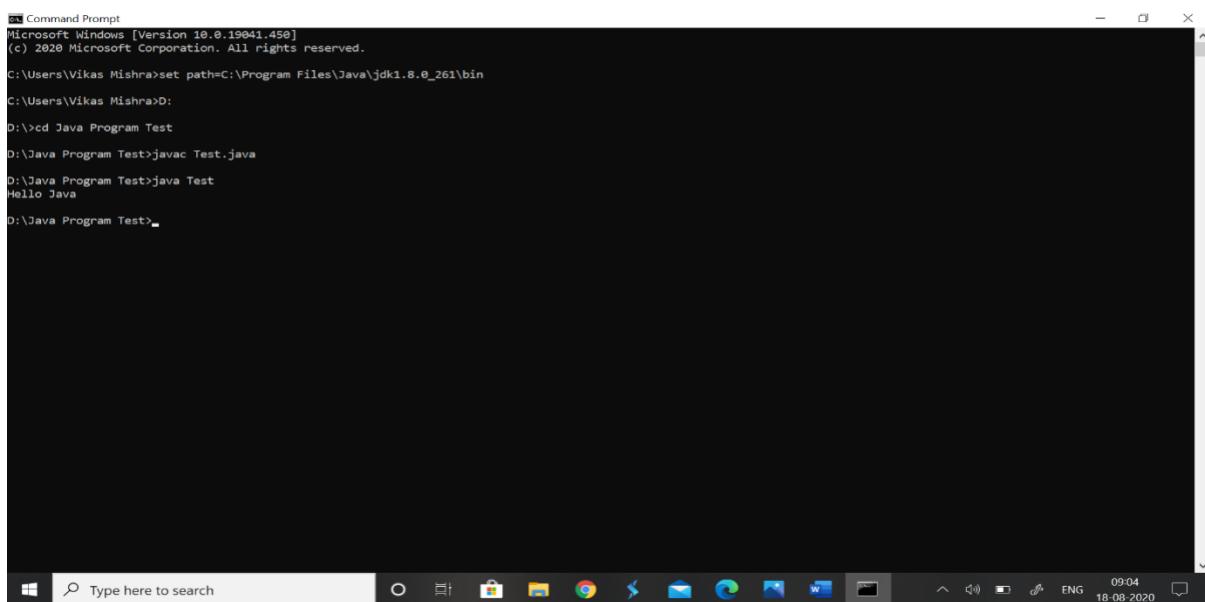
Now Click on **File->Save As->Local Disk(D:)->JAVA PROGRAMS-> Test.java**  
Now Open **COMMAND PROMPT**

### A)TEMPORARY PATH

**Write the commands:**

```
#java
#set path= C:\Program Files\Java\jdk1.8.0_261\bin
#D:
#cd JAVA PROGRAMS
#javac Test.java
#java Test
```

**RESULT:** Hello JAVA



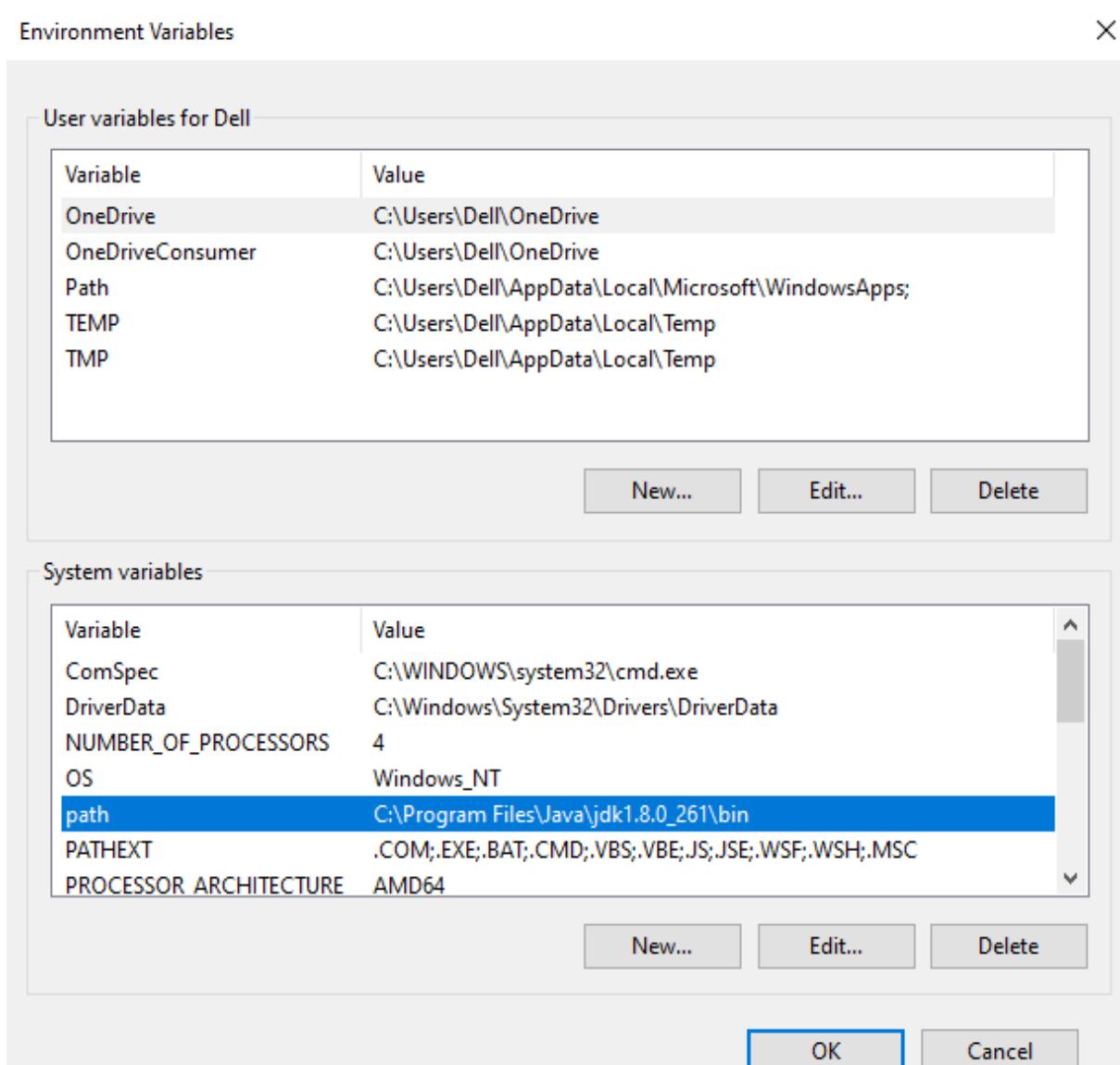
```
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Vikas Mishra>set path=C:\Program Files\Java\jdk1.8.0_261\bin
C:\Users\Vikas Mishra>D:
D:\>cd Java Program Test
D:\Java Program Test>javac Test.java
D:\Java Program Test>java Test
Hello Java
D:\Java Program Test>
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays a series of commands entered by the user: setting the path to the Java bin directory, switching to drive D:, changing to the "Java Program Test" directory, compiling the "Test.java" file with "javac", and finally running the compiled program with "java". The output of the program, "Hello Java", is shown. The command prompt window has a standard Windows title bar and status bar at the bottom indicating "09:04", "09-08-2020", and "ENG". The taskbar at the bottom of the screen is also visible.

### B)PERMANENT PATH

My PC-> Properties-> Advanced System Setting-> Environment Variables-> System Variables-> New...> Type **path**-> Variable value-> **C:\Program Files\Java\jdk1.8.0\_261\bin** -> OK-> OK-> OK  
And close the window



Now path is set Permanent and Now open **Command Prompt**

**Write the commands:**

```
#java  
#D:  
#cd JAVA PROGRAMS  
#Javac Test.java  
#java Test
```

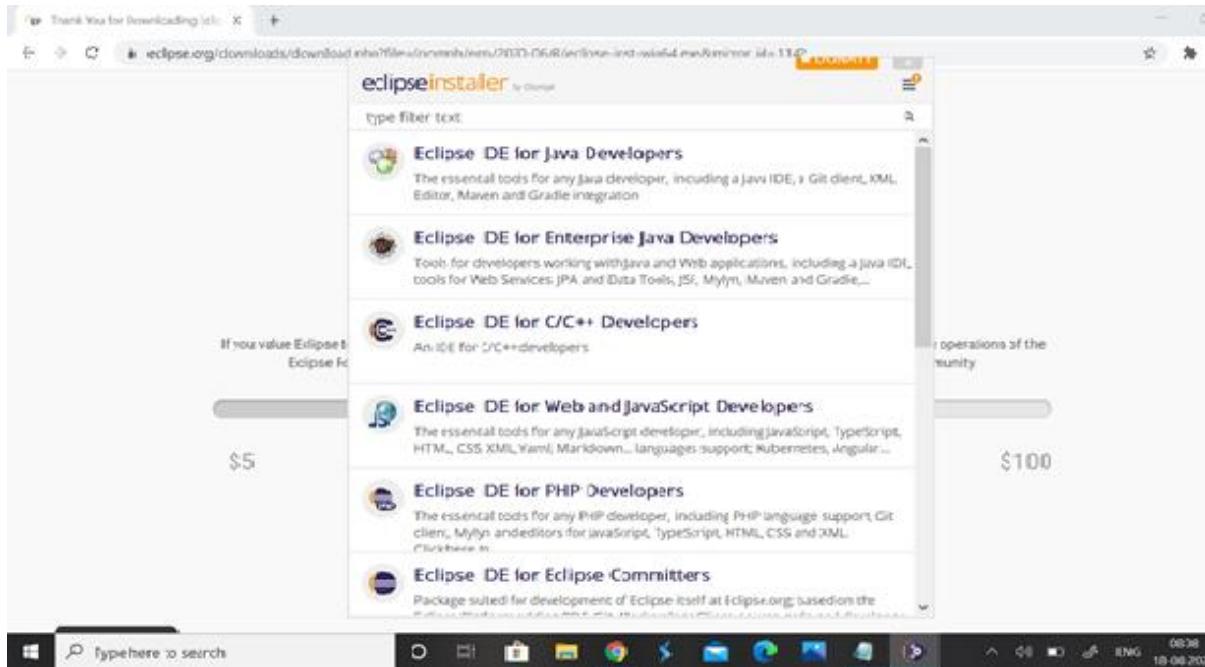
**RESULT:** Hello JAVA

```
Command Prompt
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

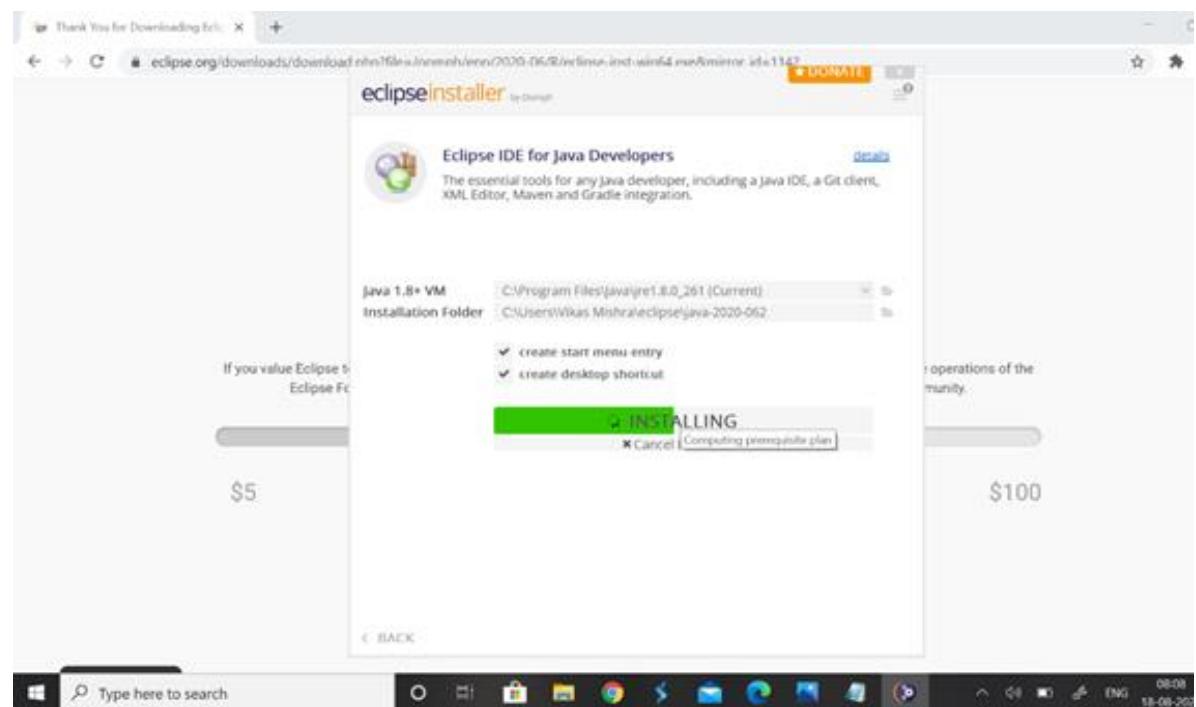
C:\Users\Vikas Mishra>D:
D:\>cd Java Program Test
D:\Java Program Test>javac Test.java
D:\Java Program Test>java Test
Hello Java
D:\Java Program Test>
```

### 3) Introduction to Eclipse

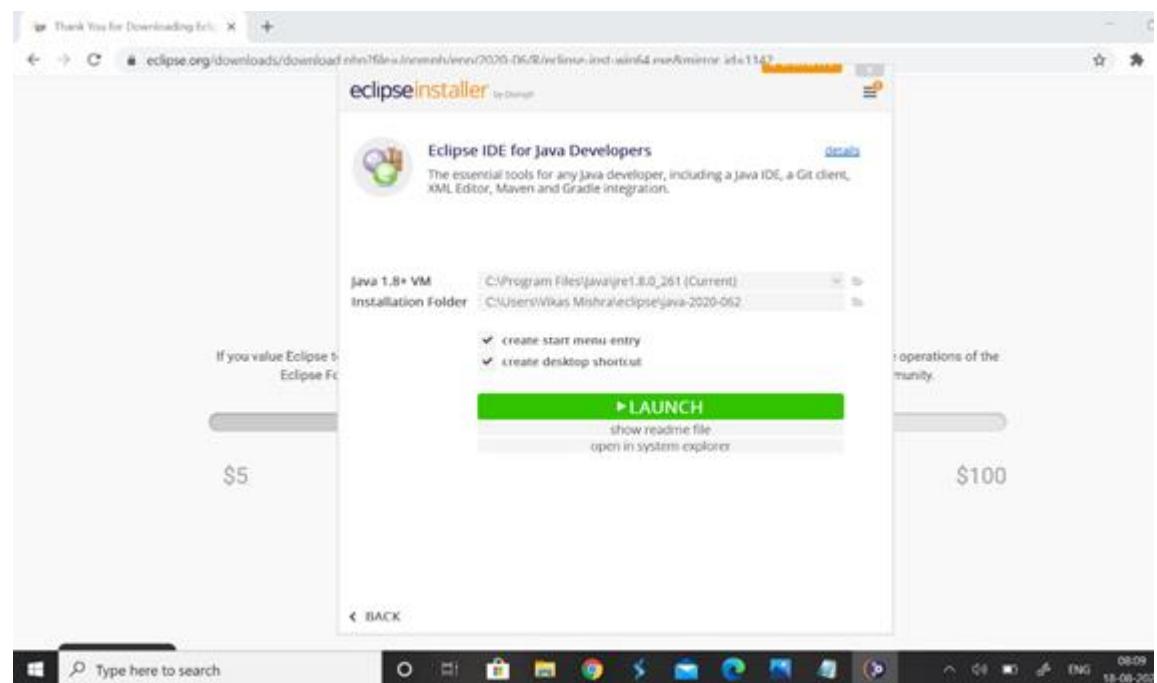
Search for [eclipse.org](http://eclipse.org) and click **Eclipse IDE for JAVA Developers**



## Now Install it

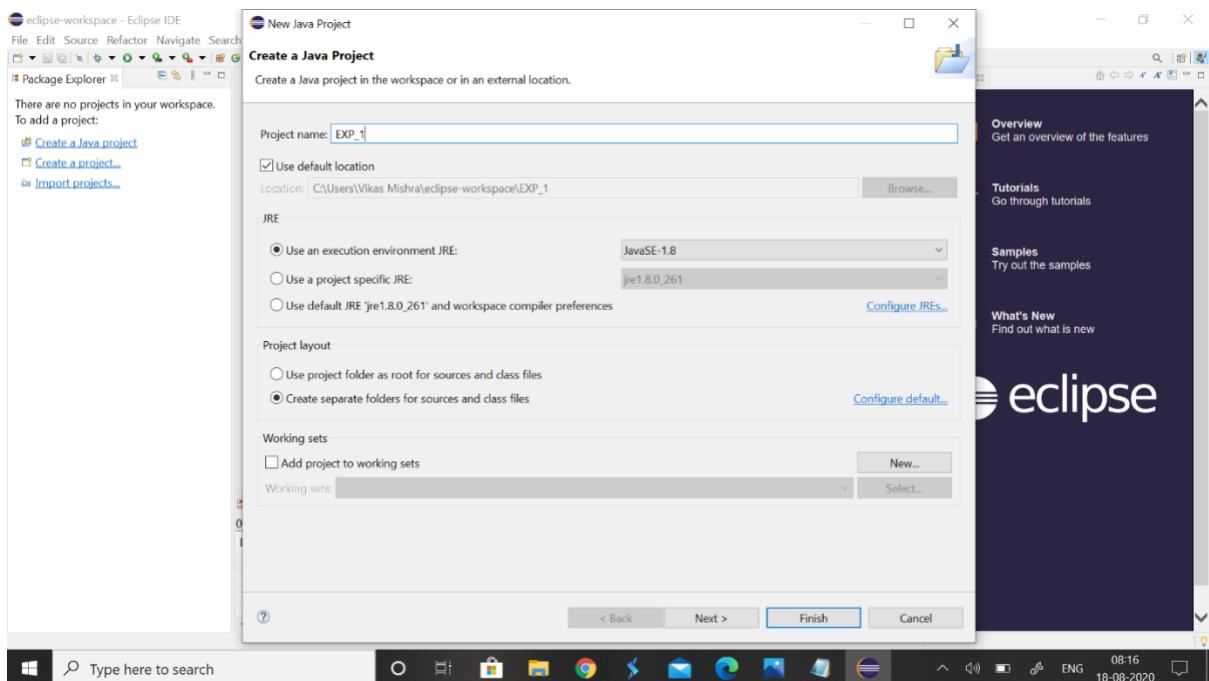


## LAUNCH ECLIPSE





Go to File-> New..->Java Project-> Create->Project Name->Name  
Modifiers as **public**  
AND START DOING CODING



## **EXPERIMENT NO -2,3**

**TITLE:** Basic Java Programming

**1. Write a program to find the largest of 3 numbers**

**CODE**

```
public class LargestNumber {  
  
    public static void main(String[] args) {  
        int i=5;  
        int j=6;  
        int k=10;  
        if(i>j && i>k) {  
            System.out.println("i is greater "+i);  
        }  
        else if(j>i && j>k) {  
            System.out.println("j is greater "+j);  
        }  
        else{  
            System.out.println("k is greater "+k);  
        }  
    }  
}
```

## OUTPUT

```
Java Program Test - Exp 1/src/LargestNumber.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer L LargestNumber.java PrimitiveValues.java FibonacciSeries.java random.java Task List Find All Activate...
1 public class LargestNumber {
2
3     public static void main(String[] args) {
4         int i=5;
5         int j=6;
6         int k=10;
7         if(i>j && i>k) {
8             System.out.println("i is greater "+i);
9         }
10        else if(j>i && j>k) {
11            System.out.println("j is greater "+j);
12        }
13        else{
14            System.out.println("k is greater "+k);
15        }
16    }
17 }
18
19 }
20

Problems Javadoc Declaration Coverage
<terminated> LargestNumber [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (17 Aug, 2020 9:57:21 PM - 9:57:24 PM)
k is greater 10

Windows Type here to search Writable Smart Insert 14 : 47 : 299 21:57 ENG 17-08-2020
```

## 2. Write a program to add two number using command line arguments

### CODE

```
import java.lang.*;
public class Addition
{
    public static void main(String args[])
    {
        int a=Integer.parseInt(args[0]);
        int b=Integer.parseInt(args[1]);

        int c=a+b;
        System.out.println("First number is:"+a);
        System.out.println("Second number is:"+b);
        System.out.println("Addition of two no is:"+c);
    }
}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java Program Test - Exp 1/src/Addition.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Package Explorer:** Shows a project named "Exp 1" containing several Java files: Addition.java, Calculator.java, FibonacciSeries.java, LargestNumber.java, PrimitiveValues.java, and random.java.
- Editor Area:** Displays the content of the "Addition.java" file:

```
1 import java.lang.*;
2 public class Addition
3 {
4     public static void main(String args[])
5     {
6         int a=Integer.parseInt(args[0]);
7         int b=Integer.parseInt(args[1]);
8
9         int c=a+b;
10        System.out.println("First number is:"+a);
11        System.out.println("Second number is:"+b);
12        System.out.println("Addition of two no is:"+c);
13    }
14 }
```
- Console Area:** Shows the terminal output of the application:

```
First number is:1
Second number is:5
Addition of two no is:6
```
- Bottom Status Bar:** Shows the date and time: 17-Aug-2020 11:02:02 PM.

### 3. Write a program to print Fibonacci series using loop.

#### CODE

```
class FibonacciSeries{
public static void main(String args[])
{
int n1=0,n2=1,n3,i,count=20;
System.out.print(n1+" "+n2);

for(i=2;i<count;++i)
{
n3=n1+n2;
System.out.print(" "+n3);
n1=n2;
n2=n3;
}

}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java Program Test - Exp 1/src/FibonacciSeries.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with icons for file operations.
- Package Explorer:** Shows a project named "Exp 1" with a package "src" containing files: Addition.java, Calculator.java, FibonacciSeries.java, LargestNumber.java, PrimitiveValues.java, and random.java.
- Editor:** Displays the code for "FibonacciSeries.java". The code prints the Fibonacci series up to 20 terms.
- Outline View:** Shows the class structure with "FibonacciSeries" and its main method.
- Task List:** Empty.
- Problems View:** Shows a terminated Java application with the output: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181.
- Console View:** Shows the command-line interface with the same output.
- Bottom Status Bar:** Shows the date (17-Aug-2020), time (11:03:46 PM), and system status (ENG).

## 4. Write a program to implement a command line calculator.

### CODE

```
public class Calculator {  
  
    public static void main(String[] args) {  
        int num1=Integer.parseInt(args[0]);  
        int num2=Integer.parseInt(args[1]);  
        int a=Integer.parseInt(args[2]);  
  
        System.out.println("First value : "+num1);  
        System.out.println("Second value : "+num2);  
        System.out.println("choice for calculation : "+a);  
  
        switch(a)  
        {  
        case 1:  
            System.out.println( "addition of two number: "+(num1+num2));  
            break;  
        case 2:  
            System.out.println( "subtraction of two number: "+(num1-num2));  
            break;  
        case 3:  
            System.out.println( "multiplication of two number: "+(num1*num2));  
            break;  
        case 4:  
            System.out.println( "division of two number: "+(num1/num2));  
            break;  
        }  
    }  
}
```

```

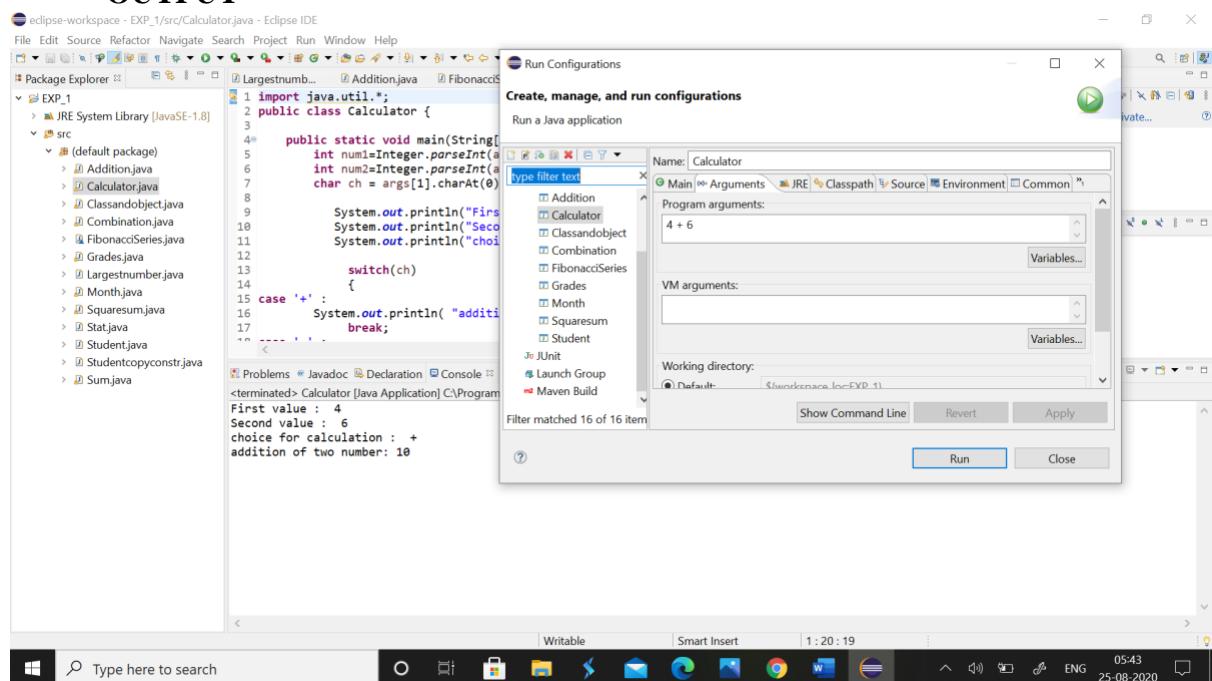
        System.out.println( "multiplication of two number:
" +(num1*num2));
    break;
case 4:System.out.println( "division of two number: "+(num1/num2));
    break;
default:System.out.println("wrong choice");
    break;

}

}

```

## OUTPUT



## 5. Write a program using classes and objects in java

### CODE

```

import java.util.*;
public class Classandobject
{
    int id;
    double marks;
    String name;

    void student1(int i,double m,String n){
        id=i;

```

```
marks=m;
name=n;

if(marks>60)
{
    System.out.println("name is: "+ name);
    System.out.println("Passed");
    System.out.println("Student id is "+id);
}
else
{
    System.out.println("name is: "+ name);
    System.out.println("not Passed");
    System.out.println("Student id is "+id);
}
}
```

```
public static void main(String[] args)
{
    Classandobject obj=new Classandobject();
    Scanner sc =new Scanner(System.in);
    System.out.println("Enter the student's name");
    String name=sc.nextLine();
    System.out.println("Enter the student's id");
    int id=sc.nextInt();
    System.out.println("Enter the student's marks");
    double marks=sc.nextDouble();
    System.out.println("Enter the student's name");
    obj.student1(id,marks,name);
}
```

## OUTPUT

```
6  double marks;
7  String name;
8
9  void student1(int i,double m,String n){
10    id=i;
11    marks=m;
12    name=n;
13
14    if(marks>60)
15    {
16      System.out.println("name is: "+ name);
17      System.out.println("Passed");
18      System.out.println("Student id is "+id);
19
20    }
21    else
22    {
23      System.out.println("name is: "+ name);
24      System.out.println("not Passed");
25      System.out.println("Student id is "+id);
26
27    }
28
29
30 }
```

Markers Properties Servers Data Source Explorer Snippets Console

Enter the student's name  
Aviral  
Enter the student's id  
500076136  
Enter the student's marks  
96  
Enter the student's name  
name is: Aviral  
Passed  
Student id is 500076136

6. Write a program to accept 10 student's marks in an array, arrange it into ascending order, convert into the following grades and print marks and grades in the tabular form.

**Between 40 and 50 : PASS**

**Between 51 and 75 : MERIT**

**and above : DISTINCTION**

## CODE

```
import java.util.*;
import java.io.*;
public class Grades {

  public static void main(String[] args) {

    int temp;
    int arr[] = new int[10];
```

```
Scanner sc=new Scanner(System.in);
System.out.println("Enter the marks greater than or equal to 40");
for(int i=0;i<10;i++) {

    arr[i]=sc.nextInt();
    for (int i = 0; i < 10; i++) {
        for (int j = i + 1; j < 10; j++) {
            if (arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    for(int i=0;i<10;i++) {
        if(arr[i]>=40&&arr[i]<=50) {
            System.out.println(arr[i]+":PASS");
        }
        else if(arr[i]>=51&&arr[i]<=75) {
            System.out.println(arr[i]+":MERIT");
        }
        else if(arr[i]>75) {
            System.out.println(arr[i]+":DISTINCTION");
        }
        else {
            System.out.println("INVALID INPUT FAIL");
        }
    }
}
```

## OUTPUT

```
21     arr[j] = temp;
22 }
23 }
24 }
```

Problems Javadoc Declaration Console  
<terminated> Grades [Java Application] C:\Program Files\Java\jre1.8.0\_261\bin\javaw.exe (24 Aug, 2020 10:29:16 PM – 10:29:47 PM)  
Enter the marks greater than or equal to 48  
68  
87  
88  
56  
45  
87  
54  
98  
54  
45  
45 :PASS  
45 :PASS  
54 :MERIT  
54 :MERIT  
56 :MERIT  
68 :MERIT  
87 :DISTINCTION  
87 :DISTINCTION  
88 :DISTINCTION  
98 :DISTINCTION

7. Write a program to accept three digits (i.e. 0 - 9) and print all its possible combinations. (For example if the three digits are 1, 2, 3 than all possible combinations are : 123, 132, 213, 231, 312, 321.)

## CODE

```
import java.util.*;
import java.io.*;
public class Combination {

    public static void main(String[] args) {

        int arr[]={new int[3];
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the 3 numbers of which possible
combinations we want to generate");
        for(int i=0;i<3;i++) {

            arr[i]=sc.nextInt();
        }
```

```

for(int i=0;i<3;i++) {
    for(int j=0;j<3;j++) {
        for(int k=0;k<3;k++) {
            if(i!=j&&j!=k&&k!=i) {

System.out.println(arr[i]+"'"+arr[j]+"'"+arr[k]);
        }
    }
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages EXP\_1 and src.
- Editor:** Displays the main.java file containing Java code to print combinations of three numbers.
- Output Window:** Shows the terminal output of the program execution, listing the generated combinations: 123, 132, 213, 231, 312, and 321.
- Task List:** Shows a search bar with "Find" and "All" options.
- Outline:** Shows the class Combination and its main method.

**8. Write a Java Program to accept 10 numbers in an array and compute the square of each number. Print the sum of these numbers.**

## CODE

```

import java.util.*;
import java.io.*;
public class Squaresum {

```

```

public static void main(String[] args) {

    int sum=0;
    int arr[]={};
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter ten numbers");
    for(int i=0;i<10;i++) {
        arr[i]=sc.nextInt();
    }
    for(int i=0;i<10;i++) {
        sum=sum+(arr[i]*arr[i]);
    }
    System.out.println("Sum of these squares of ten numbers is
"+sum);

}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under EXP\_1. The Squaresum.java file is selected.
- Code Editor:** Displays the Java code for Squaresum.java.
- Console:** Shows the command-line output of the Java application. It prompts for input ("Enter ten numbers") and then displays the result ("Sum of these squares of ten numbers is 385").
- Task List:** Shows a search bar and a list of tasks.
- Outline:** Shows the class definition and the main method.
- Bottom Bar:** Includes the Windows taskbar with various icons and system status.

```

eclipse-workspace - EXP_1/src/Squaresum.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer ▾ JRE System Library [JavaSE-1.8] Task List ▾
EXP_1 ▾ src ▾ Squaresum.java ▾ Find All Activate...
  ▾ (default package)
    ▾ Addition.java
    ▾ Calculator.java
    ▾ Classandobj.java
    ▾ Combination.java
    ▾ FibonacciSeries.java
    ▾ Grades.java
    ▾ Largestnumber.java
    ▾ Month.java
    ▾ Squaresum.java
    ▾ Stat.java
    ▾ Student.java
    ▾ Studentcopyconstr.java
    ▾ Sum.java
  ▾ Largestnumb... ▾ Addition.java ▾ FibonacciSe... ▾ Classandobj... ▾ Combination.... ▾ Squaresum.java ▾
  8   int arr[]={};
  9   Scanner sc=new Scanner(System.in);
 10  System.out.println("Enter ten numbers");
 11  for(int i=0;i<10;i++) {
 12      arr[i]=sc.nextInt();
 13  }
 14  for(int i=0;i<10;i++) {
 15      sum=sum+(arr[i]*arr[i]);
 16  }
 17  System.out.println("Sum of these squares of ten numbers is "+sum);
 18
 19
 20
 21 }
 22
 23 }
 24 }

Problems Declaration Console ▾
<terminated>:Squaresum [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (24 Aug, 2020 10:50:02 PM – 10:50:12 PM)
Enter ten numbers
1
2
3
4
5
6
7
8
9
10
Sum of these squares of ten numbers is 385

```

**9. Write a program to input a number of a month (1 - 12) and print its equivalent name of the month.( e.g 1 to Jan, 2 to Feb. 12 to Dec.)**

## CODE

```
import java.util.*;
import java.io.*;
public class Month {

    public static void main(String[] args) {

        Scanner sc =new Scanner(System.in);
        String month[]=
        {"January","February","March","April","May","June","July","August","September",
        "October","November","December"};
        System.out.println("Enter the month number:");
        int num=sc.nextInt();
        if(num>=1&&num<=12) {
            System.out.println(num+" is "+month[num-1]);
        }
        else {
            System.out.println("Entered a wrong number to print
the equivalent month");
        }
    }
}
```



}

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - EXP\_1/src/Sum.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Project Run Window Help
- Toolbars:** Standard toolbar with icons for New, Open, Save, Cut, Copy, Paste, Find, etc.
- Left Sidebar:** Package Explorer showing project structure. The EXP\_1 project contains a src folder with various Java files like Addition.java, Calculator.java, Classandobject.java, Combination.java, FibonacciSeries.java, Grades.java, Largestnumber.java, Month.java, Squaresum.java, Stat.java, Student.java, Studentcopyconstr.java, and Sum.java.
- Central Area:** Code editor showing the content of Sum.java. The code calculates the sum of integers greater than 40 and less than 250 that are divisible by 5.

```
1 import java.util.*;  
2 public class Sum {  
3     public static void main(String[] args) {  
4         // TODO Auto-generated method stub  
5         int sum=0;  
6         for(int i=41;i<250;i++)  
7         {  
8             if(i%5==0)  
9                 sum+=i;  
10        }  
11        System.out.println("Sum of all integers greater than 40 and less than 250 that are divisible  
12        by 5 is\n"+sum);  
13    }  
14 }  
15  
16  
17  
18 }
```

- Bottom Status Bar:** Writable, Smart Insert, 14 : 27 : 340, battery icon, ENG, 2254, 24-08-2020.
- Right Side Panels:** Task List, Outline, Problems, Javadoc, Declaration, Console, and a search bar.

## **EXPERIMENT NO- 4**

## **TITLE:** Inheritance

1. Write a program in Java to create a Player class. Inherit the classes Cricket \_Player, Football \_Player and Hockey\_ Player from Player class.

CODE

```
public class C {
```

```
public static void main(String args[])
{
    criket_player c=new criket_player("Ameer","cricket",25);
```

```
football_player f=new football_player("arun","foot ball",25);
hockey_player h=new hockey_player("Ram","hockey",25);
c.show();
f.show();
h.show();
}
}
class Player
{
String name;
int age;
Player(String n,int a)
{ name=n; age=a; }
void show()
{
System.out.println("\n");
System.out.println("Player name : "+name);
System.out.println("Age : "+age);
}
}
class criket_player extends Player
{
String type;
criket_player(String n,String t,int a)
{
super(n,a);
type=t;
}
public void show()
{
super.show();
System.out.println("Player type : "+type);
}
}
class football_player extends Player
{
String type;
football_player(String n,String t,int a)
{
super(n,a);
type=t;
}
public void show()
```

```

{
super.show();
System.out.println("Player type : "+type);
}
}

class hockey_player extends Player
{
String type;
hockey_player(String n,String t,int a)
{
super(n,a);
type=t;
}
public void show()
{
super.show();
System.out.println("Player type : "+type);
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the 'Sample Aviral Project' with a JRE System Library [JavaSE-10]. Inside 'src' is the 'aviral' package containing files: c.java, Hello\_World.java, Parent.java, Player.java, Room.java, Classandobject, and Classes.
- Code Editor:** Displays the content of 'c.java'.
- Console Tab:** Shows the output of three separate runs of the program. Each run prints the player's name, age, and type.
- Taskbar:** At the bottom, it shows the Windows taskbar with the search bar, system icons, and the date/time (9/17/2020, 12:15 PM).

```

eclipse-workspace - Sample Aviral Project/src/aviral/c.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer     Quick Access
Sample Aviral Project [JRE System Library [JavaSE-10]]
src
aviral
c.java
Hello_World.java
Parent.java
Player.java
Room.java
Classandobject
Classes

c.java
53 System.out.println("Player type : "+type);
54 }
55 }
56 class hockey_player extends Player
57 {
58     String type;
59     hockey_player(String n,String t,int a)
60     {
61         super(n,a);
62         type=t;
63     }
64     public void show()
65     {
66         super.show();
67         System.out.println("Player type : "+type);
68     }
69 }
70
71
72

Markers Properties Servers Data Source Explorer Snippets Console
<terminated> c [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Sep 17, 2020, 12:07:03 PM)
Player name : Amerer
Age : 25
Player type : cricket

Player name : arun
Age : 25
Player type : foot ball

Player name : Ram
Age : 25
Player type : hockey
|<

Type here to search     12:15 PM
ENG US 9/17/2020

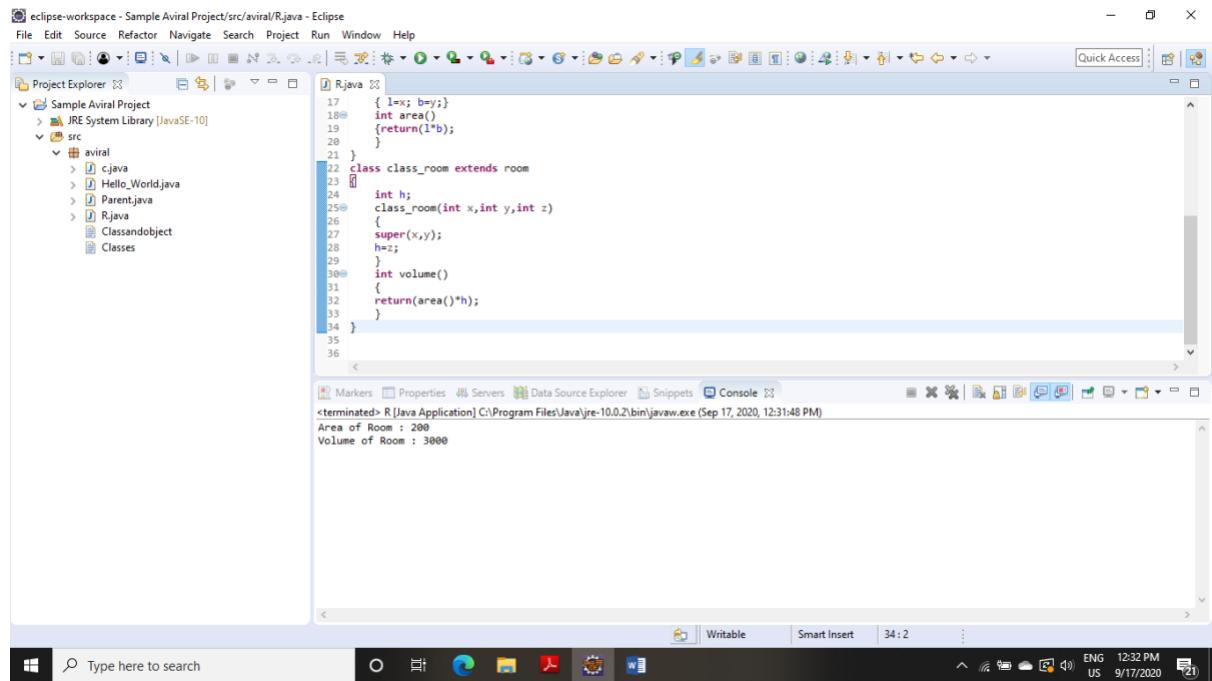
```

**2. Write a Java program to show that private member of a super class cannot be accessed from derived classes.**

**CODE**

```
public class R
{
    public static void main(String args[])
    {
        class_room cr=new class_room(10,20,15);
        int a1=cr.area();
        int v1=cr.volume();
        System.out.println("Area of Room : "+a1);
        System.out.println("Volume of Room : "+v1);
    }
}
class room
{
    private int l,b;
    room(int x,int y)
    {
        l=x; b=y;
    }
    int area()
    {
        return(l*b);
    }
}
class class_room extends room
{
    int h;
    class_room(int x,int y,int z)
    {
        super(x,y);
        h=z;
    }
    int volume()
    {
        return(area()*h);
    }
}
```

## OUTPUT



```
17    { l=x; b=y; }
18    int area()
19    {return(l*b);
20    }
21 }
22 class class_room extends room
23 {
24     int h;
25     class_room(int x,int y,int z)
26     {
27         super(x,y);
28         h=z;
29     }
30     int volume()
31     {
32         return(area()*h);
33     }
34 }
35
36
```

terminated> R [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Sep 17, 2020, 12:31:48 PM)  
Area of Room : 200  
Volume of Room : 3000

**3. Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days he/she works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.**

## CODE

```
class W
{
public static void main(String args[])
{
dailyworker d=new dailyworker(254,"Arjun",75);
salariedworker s=new salariedworker(666,"Unni",100);
d.compay(45);
s.compay();
}
}
class worker
{
String name;
```

```
int empno;
worker(int no,String n)
{ empno=no; name=n; }
void show()
{
System.out.println("\n-----");
System.out.println("Employee number : "+empno);
System.out.println("Employee name : "+name);
}
}

class dailyworker extends worker
{
int rate;
dailyworker(int no,String n,int r)
{
super(no,n);
rate=r;
}
void compay(int h)
{
show();
System.out.println("Salary : "+rate*h);
}
}

class salariedworker extends worker
{
int rate;
salariedworker(int no,String n,int r)
{
super(no,n);
rate=r;
}
int hour=40;
void compay()
{
show();
System.out.println("Salary : "+rate*hour);
}
}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface. The Project Explorer view displays a sample Aviral project with files like aviral, Hello\_World.java, Parent.java, R.java, and W.java. The W.java file is open in the editor, showing Java code for a worker class. The code defines a worker class with a salary calculation method. The code is as follows:

```
36 System.out.println("Salary : "+rate*h);
37 }
38 }
39 class salariedworker extends worker
40 {
41 int rate;
42 salariedworker(int no,String n,int r)
43 {
44 super(no,n);
45 rate=r;
46 }
47 int hour=40;
48 void compay()
49 {
50 show();
51 System.out.println("Salary : "+rate*hour);
52 }
53 }
54
55 }
```

The Console view shows the output of the program, which prints two sets of employee details. The first set is for employee number 254, name Arjun, and salary 3375. The second set is for employee number 666, name Unni, and salary 4000.

```
Employee number : 254
Employee name : Arjun
Salary : 3375

-----
Employee number : 666
Employee name : Unni
Salary : 4000
```

**4. Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges.**

## CODE

```
import java.util.Scanner;
class Telephone{
    int callnumber;
    String calltype;
    Telephone(int c,String s){
        callnumber = c;
        calltype = s;
    }
    void show() {
        System.out.println("call number"+ " "+callnumber);
        System.out.println("call type"+ " "+calltype);
    }
}
class Ordinary extends Telephone{
    float cost;
    Ordinary(int c,String s,float co){
        super(c,s);
        this.cost = co;
    }
}
```

```

void charge(double time) {
    super.show();//calling show() of parent class
    System.out.println("call charges"+ " "+cost*time);
}
}

class Urgent extends Telephone{
    float cost;
    Urgent(int c,String s,float co){
        super(c,s);
        this.cost = co;
    }
    void charge(double time) {
        super.show();
        System.out.println("call charges"+ " "+cost*time);
    }
}

class Lightening extends Telephone{
    float cost;
    Lightening(int c,String s,float co){
        super(c,s);
        this.cost = co;
    }
    void charge(double time) {
        super.show();
        System.out.println("call charges"+ " "+cost*time);
    }
}

public class Exp_4_4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        int a,no;
        double t;
        System.out.println("Enter 1 for ordinary call,2 for urgent call or 3
for lightening call");
        a=sc.nextInt();
        switch(a) {
            case 1:
                System.out.println("Enter the number to which call is to be
made");
                no=sc.nextInt();
        }
    }
}

```

```
Ordinary obj1 = new Ordinary(no,"ordinary call",3);
System.out.println("Enter the duration of the call");
t=sc.nextDouble();
obj1.charge(t);
break;

case 2:
System.out.println("Enter the number");

no=sc.nextInt();
Urgent obj2=new Urgent(no,"Urgent call",4);
System.out.println("Enter the duration of the call");
t=sc.nextDouble();
obj2.charge(t);
break;

case 3:
System.out.println("Enter the number");
no=sc.nextInt();
Lightening obj3 = new Lightening(no,"Lightening call",5);
System.out.println("Enter the duration of the call");
t=sc.nextDouble();
obj3.charge(t);

}

}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Exp6/src/Exp\_4.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Package Explorer:** Shows projects EXP\_1, Exp4, and Exp6. Exp6 contains a JRE System Library (JavaSE-1.8) and a src folder with Exp\_4.java.
- Code Editor:** Displays Java code for a call center application. The code includes methods for handling ordinary, urgent, and lightening calls, each calculating charges based on duration and number.
- Console:** Shows the execution of the program. It prompts for a call type (1 for ordinary, 2 for urgent, 3 for lightening), takes input (2), and then asks for a call number (345656) and duration (35.8). The output shows the call number, type, and total charges.
- Taskbar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and Task View.

**5. Design a class employee of an organization. An employee has a name, empid, and salary. Write the default constructor, a constructor with parameters (name, empid, and salary) and methods to return name and salary. Also write a method increaseSalary that raises the employee's salary by a certain user specified percentage. Derive a subclass Manager from employee. Add an instance variable named department to the manager class. Supply a test program that uses these classes and methods.**

## CODE

```
import java.util.Scanner;
class Employee{
    int empid;
    String name;
    double salary;
    Employee(){
        System.out.println("This is employee default constructor");
    }
    Employee(int emid, String name, double salary){
        this.empid = emid;
        this.name = name;
    }
}
```

```

        this.salary = salary;
    }
    String name()
    {
        return name;
    }
    double salary()
    {
        return salary;
    }
    void increase_salary(int i) {
        double x;
        x=((salary*i)/100)+salary;
        System.out.print("Increased salary ="+ " "+x);
    }
}

class Manager extends Employee{
    String department;
    Manager(int e,String n,double sal,String d){
        super(e,n,sal);
        department = d;
    }
    void display() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Name of the employee"+ " "+super.name());
        System.out.println("salary of a employee"+ " "+super.salary());

        System.out.print("department is"+ " "+ department+"\n");
        System.out.println("Enter the percentage by which a salary is to be
increased");
        int ps=sc.nextInt();
        super.increase_salary(ps);
    }
}
public class Exp_5_5 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the name of the employee");
        String name=sc.nextLine();

```

```

        System.out.println("Enter the id of the employee");
        int id=sc.nextInt();
        System.out.println("Enter the salary of the employee");
        double sal=sc.nextDouble();
        Manager M = new Manager(id,name,sal,"CS");
        M.display();
    }

}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "Sample Aviral Project".
- Code Editor:** Displays the Java code for `Exp_5_5.java`. The code prompts for employee name, ID, and salary, creates a `Manager` object, and prints the details.
- Console Output:** Shows the terminal output of the application. It asks for the name ("Aviral Mehra"), ID ("500076136"), and salary ("500000"). It then displays the employee details and calculates a 10% salary increase ("Increased salary = 550000.0").
- System Bar:** Shows the Windows taskbar with the application icon, search bar, and system status.

```

eclipse-workspace - Sample Aviral Project/src/aviral/Exp_5_5.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Sample Aviral Project
  JRE System Library [JavaSE-10]
  src
    aviral
      c.java
      Console.java
      DataString.java
      Div_by_zero_exception.java
      Division.java
      Exception2.java
      Exp_5_5.java
      GenThread.java
      Hello_World.java
      IExtend.java
      Inheritance1.java
      Inter.java
      main_class.java
      Multi1.java
      Multi2.java
      NonNumeric.java
      Nonum.java
      Numb.java
      Num.java
      Parent.java
      R.java
      SearchString.java
      StringUpper.java
      Students.java
      W.java
      Classandobject
      Classes
      aviralmehra
      Employee_Exception.java
Exp_5_5.java
46     super.increase_salary(ps);
47
48 }
49
50 }
51 public class Exp_5_5 {
52
53     public static void main(String[] args) {
54         Scanner sc=new Scanner(System.in);
55         System.out.println("Enter the name of the employee");
56         String name=sc.nextLine();
57         System.out.println("Enter the id of the employee");
58         int idsc.nextInt();
59         System.out.println("Enter the salary of the employee");
60         double sal=sc.nextDouble();
61         Manager M = new Manager(id,name,sal,"CS");
62         M.display();
63
64     }
65 }
66
Markers Properties Servers Data Source Explorer Snippets Console
<terminated> Exp_5_5[Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 26, 2020, 9:24:23 AM)
Enter the name of the employee
Aviral Mehra
Enter the id of the employee
500076136
Enter the salary of the employee
500000
Name of the employee Aviral Mehra
salary of a employee 500000.0
department is CS
Enter the percentage by which a salary is to be increased
10
Increased salary = 550000.0

```

## **EXPERIMENT NO– 8**

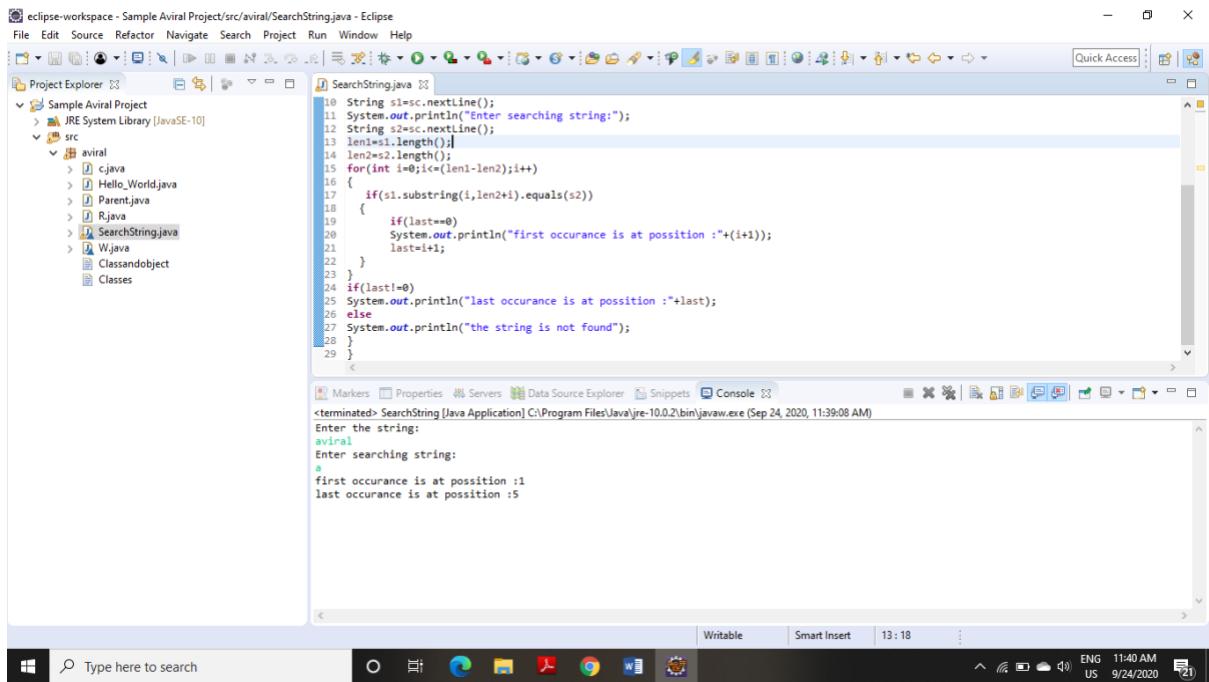
### **TITLE:** String handling

**1. Write a program for searching strings for the first occurrence of a character or substring and for the last occurrence of a character or substring.**

### **CODE**

```
import java.util.*;
class SearchString
{
    public static void main(String args[])
    {
        int len1,len2,last=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the string:");
        String s1=sc.nextLine();
        System.out.println("Enter searching string:");
        String s2=sc.nextLine();
        len1=s1.length();
        len2=s2.length();
        for(int i=0;i<=(len1-len2);i++)
        {
            if(s1.substring(i,len2+i).equals(s2))
            {
                if(last==0)
                    System.out.println("first occurrence is at position :"+(i+1));
                last=i+1;
            }
        }
        if(last!=0)
            System.out.println("last occurrence is at position :"+last);
        else
            System.out.println("the string is not found");
    }
}
```

### **OUTPUT**



**2. Write a program that converts all characters of a string in capital letters. (Use StringBuffer to store a string). Don't use inbuilt function.**

### CODE

```

import java.io.*;
public class StringUpper
{
    public static void main(String args[]) throws IOException
    {
        StringBuffer s=new StringBuffer("Aviral Mehra");
        String str = "";
        for(int x=0; x<s.length(); x++)
        {
            char ch = s.charAt(x);
            if(ch >= 'a' && ch <= 'z')
                str += "" + (char)(ch - 32);
            else
                str += "" + ch;
        }
    }
}

```

```

        System.out.println("Final String :" +str);

    }

}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left displays a project named "Sample Aviral Project" with a "src" folder containing several Java files: aviral (c.java, Hello\_World.java, Parent.java, R.java, SearchString.java, StringUpper.java), W.java, Classandobject, and Classes. The "StringUpper.java" file is open in the center editor, showing the following code:

```

import java.io.*;
public class Stringupper
{
    public static void main(String args[]) throws IOException
    {
        StringBuffer s=new StringBuffer("Aviral Mehra");
        String str = "";
        for(int x=0; x<s.length(); x++)
        {
            char ch = s.charAt(x);
            if(ch >= 'a' && ch <= 'z')
                str += "" + (char)(ch - 32);
            else
                str += "" + ch;
        }
        System.out.println("Final String :" +str);
    }
}

```

The Console view at the bottom shows the output of the program: "Final String :AVIRAL MEHRA". The taskbar at the bottom of the screen also displays the console output.

**3. Write a program in Java to read a statement from console, convert it into upper case and again print on console. (Don't use inbuilt function)**

## CODE

```

import java.io.*;
public class Console
{
    public static void main(String a[]) throws IOException
    {
        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter file Statement:");
        String s1=in.readLine();

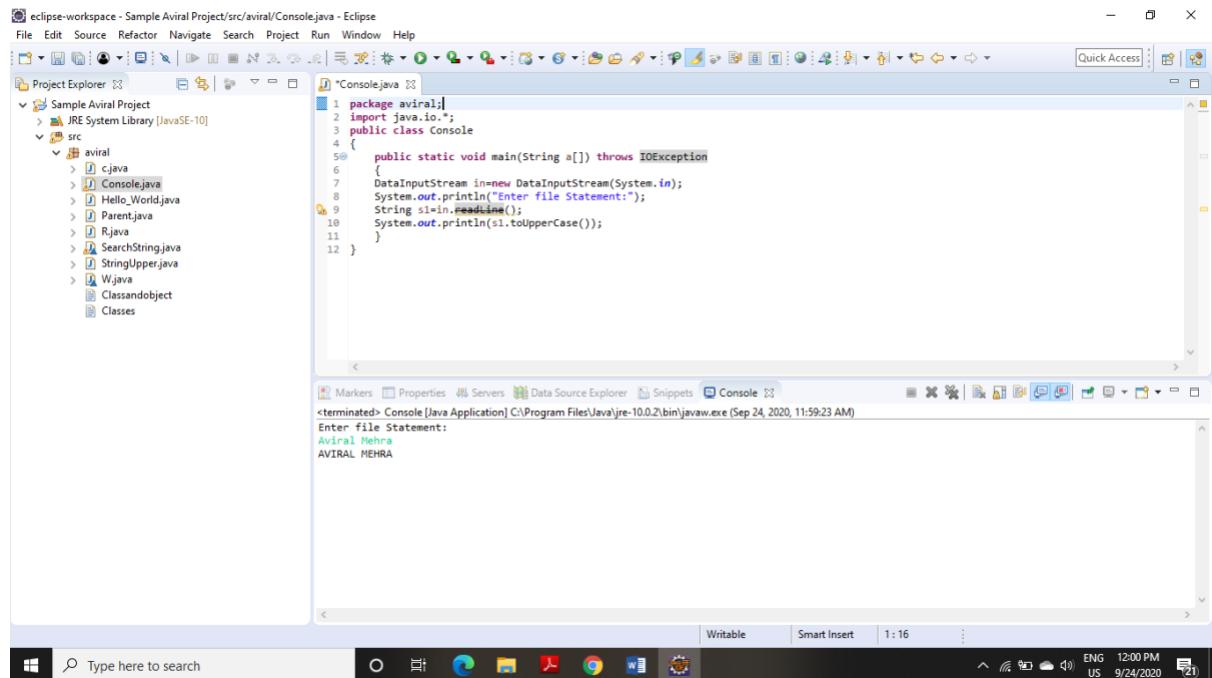
```

```

        System.out.println(s1.toUpperCase());
    }
}

```

## OUTPUT



**4. Write a program in Java to create a String object. Initialize this object with your name. Find the length of your name using the appropriate String method. Find whether the character ‘a’ is in your name or not; if yes find the number of times ‘a’ appears in your name. Print locations of occurrences of ‘a’ .Try the same for different String object**

## CODE

```

public class DataString
{
public static void main(String args[])
{
Data d1=new Data("aviral mehra");
d1.disp();
Data d2=new Data("aman mehra");
d2.disp();
}

```

```

}
class Data
{
String name;
Data(String n)
{
    name=n;
}
void disp()
{
System.out.println("Name :" +name);
int c=0;
int len=name.length();
for(int i=0;i<len;i++)
if(name.charAt(i)=='A'||name.charAt(i)=='a')
{
    c++;
    System.out.println("number of occurrence :" +c);
    System.out.println("Position :" +(i+1));
}
if(c==0)
System.out.println("there is no 'A' available in the string");
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the "Sample Aviral Project" with files: aviral, c.java, Console.java, DataString.java, Hello\_World.java, Parent.java, R.java, SearchString.java, StringUpper.java, W.java, and Classandobject.
- DataString.java Content:**

```

1 package aviral;
2 public class DataString
3 {
4     public static void main(String args[])
5     {
6         Data d1=new Data("aviral mehra");
7         d1.disp();
8         Data d2=new Data("aman mehra");
9         d2.disp();
10    }
11    class Data
12    {
13        String name;
14        Data(String n)
15        {
16            name=n;
17        }
18        void disp()
19        {
20        }
21    }
22 }
```
- Output Console:**

```

Name :aviral mehra
number of occurrence :1
Position :1
Name :aman mehra
number of occurrence :1
Position :1
number of occurrence :2
Position :3
number of occurrence :3
Position :12
Name :aman mehra
number of occurrence :1
Position :1
number of occurrence :2
Position :3
number of occurrence :3
Position :10
```

## EXPERIMENT NO-6

**TITLE:** Package

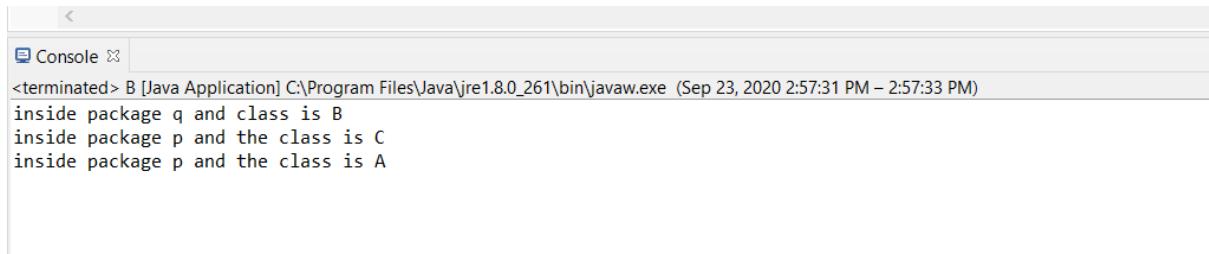
**1. Write a Java program to implement the concept of importing classes from user defined package and created packages.**

```
package p;
public class A {
    public void display()
    {
        System.out.println("inside package p and the class is A");
    }
}
```

```
package p;
public class C{
    public void display()
    {
        System.out.println("inside package p and the class is C");
    }
}
```

```
package q;
import p.*;
public class B{
    private void disp()
    {
        System.out.println("inside package q and class is B");
    }
    public static void main(String[] args) {
        B o=new B();
        o.disp();
        C ob=new C();
        ob.display();
        A obj=new A();
        obj.display();
    }
}
```

**OUTPUT**



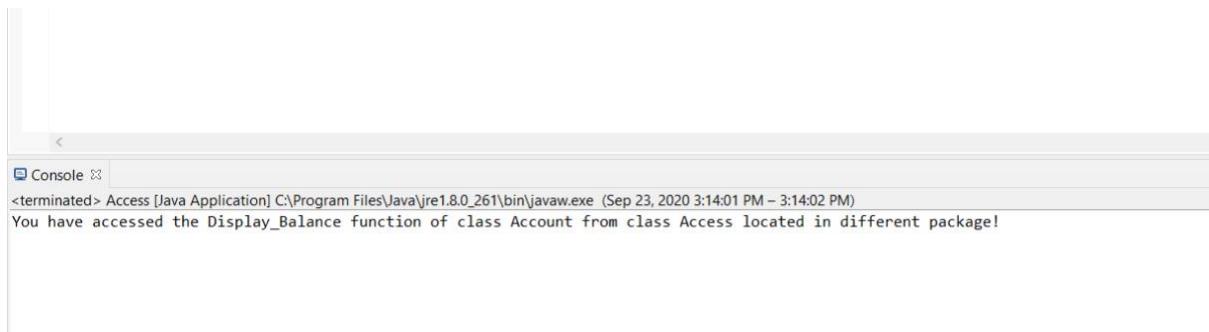
```
Console <terminated> B [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Sep 23, 2020 2:57:31 PM – 2:57:33 PM)
inside package q and class is B
inside package p and the class is C
inside package p and the class is A
```

**2. Write a program to make a package Balance. This has an Account class with Display\_Balance method. Import Balance package in another program to access Display\_Balance method of Account class.**

```
package Balance;
public class Account {
    public void Display_Balance()
    {
        System.out.print("You have accessed the Display_Balance
function of class Account");
    }
}
package q;
import Balance.Account;
public class Access {

    public static void main(String[] args) {
        Account obj=new Account();
        obj.Display_Balance();
        System.out.println(" from class Access located in different
package!");
    }
}
```

## OUTPUT



```
<terminated> Access [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Sep 23, 2020 3:14:01 PM – 3:14:02 PM)
You have accessed the Display_Balance function of class Account from class Access located in different package!
```

## **EXPERIMENT NO – 7**

### **TITLE:** Exceptions

**1. Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle `ArrayIndexOutOfBoundsException`, so that any such problem doesn't cause illegal termination of program.**

### **CODE**

```
import java.io.*;
class student
{
    String name,grade;
    int reg,m1,m2,m3;
    void read()throws Exception
    {
        DataInputStream in=new DataInputStream(System.in);
        System.out.println("Enter the register no:");
        reg=Integer.parseInt(in.readLine());
        System.out.println("Enter name :");
        name=in.readLine();
        System.out.println("Enter marks 1:");
        m1=Integer.parseInt(in.readLine());
        System.out.println("Enter marks 2:");
        m2=Integer.parseInt(in.readLine());
        System.out.println("Enter marks 3:");
        m3=Integer.parseInt(in.readLine());
    }
    void disp_grade()
    {
        int total=m1+m2+m3;
        if(total>=250) grade="A";
    }
}
```

```

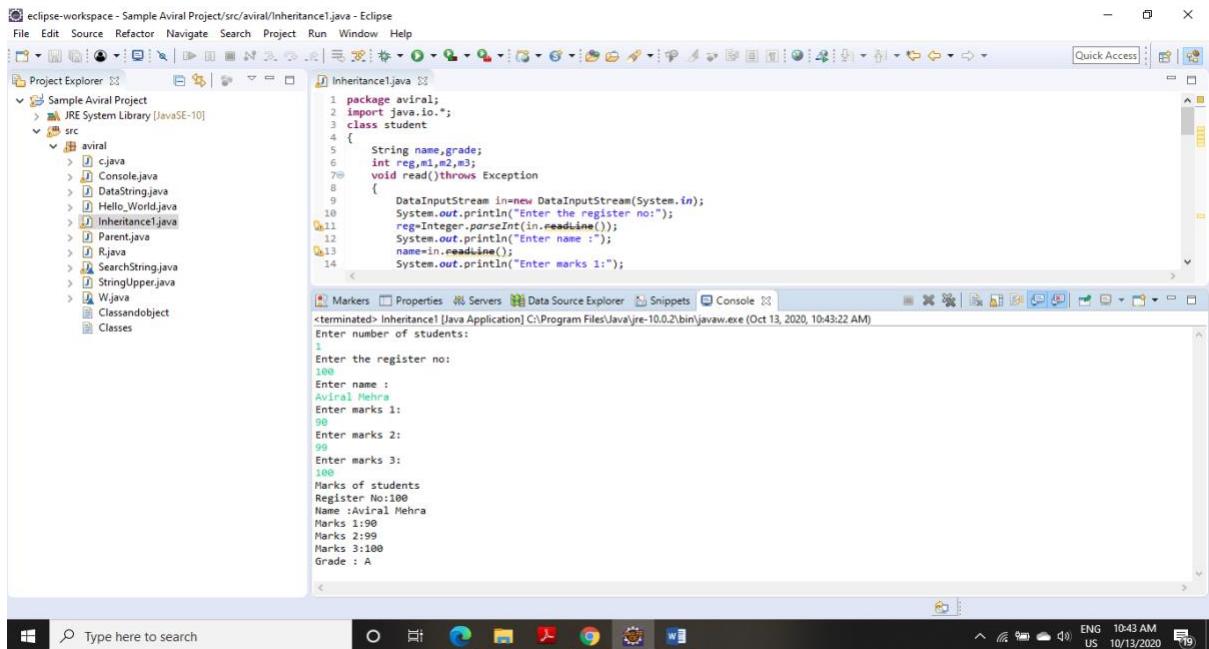
        else if(total>=200) grade="B";
        else if(total>=150) grade="C";
        else if(total>=100) grade="D";
        else grade="E";
        System.out.println("Grade : "+grade);
    }
    void disp()
    {
        System.out.println("Marks of students");
        System.out.println("Register No:" +reg);
        System.out.println("Name :" +name);
        System.out.println("Marks 1:" +m1);
        System.out.println("Marks 2:" +m2);
        System.out.println("Marks 3:" +m3);
        disp_grade();
    }
}

public class Inheritance1
{
    public static void main(String args[])
    {

        int n=0;
        student s=new student();
        try
        {
            DataInputStream in=new DataInputStream(System.in);
            System.out.println("Enter number of students:");
            n=Integer.parseInt(in.readLine());
            for(int x=0;x<n;x++);
                s.read();
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Maximum no of students is 10\n");
            n=10;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        for(int x=0;x<n;x++);
        s.disp();
    }
}

```

## OUTPUT



## 2. Write a Java program to enable the user to handle any chance of divide by zero exception.

### CODE

```

public class Exception2
{
    public static void main(String ar[])
    {
        int no=0,m=10,result=0;
        try
        {
            result=m/no;
        }
        catch(ArithmaticException e)
        {
            System.out.println(" division by zero ");
            System.out.println(" value of result has been set as one");
            result=1;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        System.out.println(" Result :" +result);
    }
}

```

```
}
```

## OUTPUT

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer [ ] Quick Access [ ]
Sample Aviral Project [ ] JRE System Library [JavaSE-10]
src [ ] aviral [ ] cjava [ ] Console.java [ ] DataString.java [ ] Exception2.java [ ] Hello_World.java [ ] Inheritance1.java [ ] Parent.java [ ] R.java [ ] SearchString.java [ ] StringUpper.java [ ] W.java [ ] Classandobject [ ] Classes [ ]
Exception2.java [ ]
1 package aviral;
2
3 public class Exception2 {
4     public static void main(String ar[]) {
5         int no=0,m=10,result=0;
6         try {
7             result=m/no;
8         }
9         catch(ArithmeticException e) {
10             System.out.println(" division by zero ");
11             System.out.println(" value of result has been set as one");
12             result=1;
13         }
14         catch(Exception e) {
15             System.out.println(e);
16         }
17     }
18 }
19
20
21
22 System.out.println(" Result :" +result);
```

Markers Properties Servers Data Source Explorer Snippets Console

```
<terminated> Exception2 [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 13, 2020, 10:48:08 AM)
division by zero
value of result has been set as one
Result :1
```

Type here to search

**3. Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).**

## CODE

```
import java.io.*;
import java.util.*;
class Numb
{
    public static void main(String args[])
    {
        int i,j;
        float add,sub,mul,div;
```

```
System.out.println("CALCULATOR:");
System.out.println("Enter two Operands:");
Scanner in=new Scanner(System.in);

try
{
    i=in.nextInt();
    j=in.nextInt();
    add=i+j;
    sub=i-j;
    mul=i*j;
    div=i/j;

    System.out.println("Addition =" +add);
    System.out.println("Subtraction =" +sub);
    System.out.println("Multiplication =" +mul);
    System.out.println("Division =" +div);
}

catch(InputMismatchException e)
{
    System.out.println("Program Is Terminated Exception Caught");
}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the "Sample Aviral Project" with the "src" folder expanded, displaying files like c.java, Console.java, DataString.java, Exception2.java, Hello\_World.java, Inheritance1.java, Nonum.java, Numb.java, Parent.java, R.java, SearchString.java, StringUpper.java, W.java, Classandobject, and Classes.
- Code Editor:** Displays the Numb.java code, which is a simple calculator program using Scanner to read two integers and perform addition, subtraction, multiplication, or division based on user input (+, -, \*, /). It handles InputMismatchException.
- Console:** Shows the execution output of the Java application. The user enters "50" and "40". The program outputs:
 

```

CALCULATOR:
Enter two Operands:
50
40
Addition =90.0
Subtraction =10.0
Multiplication =2000.0
Division =-1.0
      
```
- System Tray:** Shows the date and time as "10/13/2020 11:00 AM".

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the "Sample Aviral Project" with the "src" folder expanded, displaying files like c.java, Console.java, DataString.java, Exception2.java, Hello\_World.java, Inheritance1.java, Nonum.java, Numb.java, Parent.java, R.java, SearchString.java, StringUpper.java, W.java, Classandobject, and Classes.
- Code Editor:** Displays the Numb.java code, identical to the one in the first screenshot.
- Console:** Shows the execution output of the Java application. The user enters "+" and the program outputs:
 

```

CALCULATOR:
Enter two Operands:
+
Program Is Terminated Exception Caught
      
```
- System Tray:** Shows the date and time as "10/13/2020 11:00:57 AM".

**4. On a single track two vehicles are running. As vehicles are going in same direction there is no problem. If the vehicles are running in different direction there is a chance of collision. To avoid collisions write a Java program using exception handling. You are free to make necessary assumptions.**

## CODE

```
import java.io.*;
```

```

class collision extends Exception
{
collision(String s)
{
    super(s);
}
}
class coll
{
public static void main(String args[])
{
    String t1=null,t2=null;
try
{
    DataInputStream in= new DataInputStream(System.in);
    System.out.println("enter the direction of vehicle1:(left/right):");
    t1=in.readLine();
    System.out.println("enter the direction of vehicle2:(left/right):");
    t2=in.readLine();
    if(!t1.equals(t2))
throw new collision("truck2 has to go on "+ t1 +" direction");
}
catch(collision e)
{
    System.out.println(e);
    t2=t1;
    System.out.println("the collision has been avoided by redirection truck2");
}
catch(Exception e)
{
    System.out.println(e);
}
System.out.println("direction of truck1 :" +t1);
System.out.println("direction of truck2 :" +t2);
}
}
OUTPUT

```

```

eclipse-workspace - Sample Aviral Project/src/aviral/coll.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Quick Access
Sample Aviral Project
  JRE System Library [JavaSE-10]
  src
    aviral
      c.java
      coll.java
      Console.java
      DataString.java
      Exception2.java
      Hello_World.java
      Inheritance1.java
      Nonum.java
      Numb.java
      Parent.java
      R.java
      SearchString.java
      StringUpper.java
      W.java
      Classandoject
      Classes
coll.java
19   t1=in.readline();
20   System.out.println("enter the direction of vehicle2:(left/right):");
21   t2=in.readline();
22   if(!t1.equals(t2))
23     throw new collision("truck2 has to go on "+ t1 +" direction");
24   }
25   catch(collision e)
26   {
27     System.out.println(e);
28     t2=t1;
29     System.out.println("the collision has been avoided by redirection truck2");
30   }
31   catch(Exception e)
32   {
33     System.out.println(e);
34   }
35   System.out.println("direction of truck1 :" +t1);
36   System.out.println("direction of truck2 :" +t2);
37   }
38   }
39   |
<terminated> coll [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 13, 2020, 11:20:09 AM)
enter the direction of vehicle1:(left/right):
left
enter the direction of vehicle2:(left/right):
right
aviral.collision: truck2 has to go on left direction
the collision has been avoided by redirection truck2
direction of truck1 :left
direction of truck2 :left

```

**5. Write a java program to throw an exception for an employee details.** • If an employee name is a number, a name exception must be thrown. • If an employee age is greater than 50, an age exception must be thrown. • Or else an object must be created for the entered employee details

## CODE

```

import java.io.*;
import java.util.*;
class Numx
{
public static void main(String args[])
{
String name;
int age;
System.out.println("-----ENTER EMPLOYEE DETAILS-----");
System.out.println("Enter Name and Age:");
Scanner sc=new Scanner(System.in);
try
{
if(!(sc.nextLine().matches("[a-zA-Z]+")))
{
throw new IOException();
}
age=sc.nextInt();
}

```

```

if(age>50)
{
    System.out.println("Age greater than 50 Exception");
    throw new Exception();
}
Numx x=new Numx();
    System.out.println("-----Object Created-----");
}
catch(Exception e)
{
    System.out.println("Exception");
}
}
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the Sample Aviral Project with several Java files under the aviral package.
- Editor View:** Displays the Numx.java code. The line `throw new IOException();` is highlighted in blue.
- Console View:** Shows the output of the application. It prints "-----ENTER EMPLOYEE DETAILS-----", "Enter Name and Age:", and then "Exception".
- Taskbar:** At the bottom, it shows the Windows taskbar with various icons and the system status bar indicating ENG US 11:34 AM 10/13/2020.

## EXPERIMENT NO – 9

### TITLE: Threads

**1. Write a program to implement the concept of threading by extending Thread Class and Runnable interface.**

**Java Thread Example by extending Thread class  
CODE**

```

class Multi1 extends Thread
{
public void run()
{
    System.out.println("thread is running...");}
}
public static void main(String args[])
{
    Multi1 t1=new Multi1();
    t1.start();
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the "Sample Aviral Project" with the "src" folder expanded, displaying various Java files like aviral, c.java, coll.java, Console.java, DataString.java, Exception2.java, Hello\_World.java, Inheritance1.java, Multi1.java, Nonum.java, Numb.java, Num.java, Parent.java, R.java, SearchString.java, StringUpper.java, W.java, Classandobject, and Classes.
- Multi1.java:** The code editor window displays the Java code for Multi1.java.
- Console View:** The bottom window shows the output of the application. It starts with "<terminated> Multi1 [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 13, 2020, 11:42:37 AM)" followed by the printed message "thread is running...".

## Java Thread Example by implementing Runnable interface CODE

```

class Multi2 implements Runnable
{
public void run()
{
    System.out.println("thread is running...");}
}

```

```
}
```

```
public static void main(String args[])
{
    Multi2 m1=new Multi2();
    Thread t1 =new Thread(m1);
    t1.start();
}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface. The left pane displays the 'Project Explorer' with a tree view of files under 'src'. The right pane shows the 'Multi2.java' editor with the following code:

```
1 package aviral;
2 class Multi2 implements Runnable
3 {
4     public void run()
5     {
6         System.out.println("thread is running...");
7     }
8
9     public static void main(String args[])
10 {
11     Multi2 m1=new Multi2();
12     Thread t1 =new Thread(m1);
13     t1.start();
14 }
15
16
```

Below the editor is the 'Console' tab, which shows the output of the program: '<terminated> Multi2 [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 13, 2020, 11:45:11 AM) thread is running...'. The bottom status bar indicates the date and time as '10/13/2020 11:45 AM'.

**2) Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.**

## CODE

```
class even extends Thread
{
Thread t=null;
even()
{
    t=new Thread(this);
    start();
}
public void run()
{
try
{
    for(int i=2;i<50;i+=2)
        System.out.print(i+" ");
    Thread.sleep(100);
}
catch(Exception e)
{
    System.out.println("thread interepted");
}
}
}

class odd extends Thread
{
Thread t=null;
odd()
{
    t=new Thread(this);
    start();
}
public void run()
{
try
{
    for(int i=1;i<50;i+=2)
        System.out.print(i+" ");
    Thread.sleep(100);
}
catch(Exception e)
{
    System.out.println("thread interepted");
}
}
```

```

}
}

class GenThreads
{
    public static void main(String arg[])
    {
        even e=new even();
        odd o=new odd();
    }
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "Sample Aviral Project". The "src" folder contains several Java files: aviral, c.java, coll.java, Console.java, DataString.java, Exception2.java, GenThreads.java, Hello\_World.java, Inheritance1.java, Multi.java, Multi2.java, Nonum.java, Numb.java, Num.java, Parent.java, Rjava, SearchString.java, StringUpper.java, W.java, Classandobject, and Classes.
- Code Editor:** Displays the content of GenThreads.java. The code defines a class GenThreads with a main method that creates two threads, even and odd, which print even and odd numbers respectively, with a 100ms sleep between prints. A catch block handles any exceptions.
- Console View:** Shows the output of the Java application. It displays a sequence of numbers from 2 to 49, with each number followed by a space and a carriage return. The output is as follows:
 

```

2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
      
```

3. Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state.

**Check the aliveness of the threads and mark which thread is long lasting.**

## CODE

```
package aviralmehra;
public class fivethreads implements Runnable
{
    public static void main(String args[]) throws InterruptedException
    {
        Thread T1=new Thread();
        Thread T2=new Thread();
        Thread T3=new Thread();
        Thread T4=new Thread();
        Thread T5=new Thread();
        T1.setPriority(7);
        T2.setPriority(2);
        T3.setPriority(10);
        T4.setPriority(5);
        T5.setPriority(8);
        T1.sleep(1500);
        if (T1.isAlive())
            System.out.println("Thread 1 is alive");
        else
            System.out.println("Thread 1 is not alive");
        T2.start();
        if (T2.isAlive())
            System.out.println("Thread 2 is alive");
        else
            System.out.println("Thread 2 is not alive");
        T3.sleep(1000);
        if (T3.isAlive())
            System.out.println("Thread 3 is alive");
        else
            System.out.println("Thread 3 is not alive");
        T4.start();
        if (T4.isAlive())
            System.out.println("Thread 4 is alive");
        else
            System.out.println("Thread 4 is not alive");
        T5.start();
        if (T5.isAlive())
            System.out.println("Thread 5 is alive");
```

```

else
System.out.println("Thread 5 is not alive");
}
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left lists several Java files under the 'src' folder of the 'Sample Aviral Project'. The Editor view in the center displays the code for 'fivethreads.java'. The code creates five threads and prints their status. The Console view at the bottom shows the execution results, indicating that all five threads are alive.

```

package aviralmehra;
public class fivethreads implements Runnable
{
    public static void main(String args[]) throws InterruptedException
    {
        Thread T1=new Thread();
        Thread T2=new Thread();
        Thread T3=new Thread();
        Thread T4=new Thread();
        Thread T5=new Thread();
        T1.setPriority(7);
        T2.setPriority(2);
        T3.setPriority(10);
        T4.setPriority(5);
        T5.setPriority(8);
        T1.sleep(1500);
        if (T1.isAlive())
            System.out.println("Thread 1 is alive");
        else
            System.out.println("Thread 1 is not alive");
        T2.start();
    }
}


```

```

terminated: fivethreads [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 14, 2020, 10:00:57 AM)
Thread 1 is not alive
Thread 2 is alive
Thread 3 is not alive
Thread 4 is alive
Thread 5 is alive

```

**4. Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization.**

### CODE

```

package aviralmehra;
class ThreadSynch
{
    public static void main(String arg[])throws Exception
    {
        data d1=new data();
        data d2=new data();
        data d3=new data();
        data d4=new data();
        data d5=new data();

```

```

data d6=new data();
data d7=new data();
data d8=new data();
data d9=new data();
data d10=new data();
System.out.println(d10.count);
}
}
class item
{
    static int count=0;
}
class data extends item implements Runnable
{
    item d=this;
    Thread t;
data()
{
    t=new Thread(this);
    t.start();
}
public void run()
{
    d=syn.increment(d);
}
}
class syn
{
    synchronized static item increment(item i)
    {
        i.count++;
        return(i);
    }
}

```

## OUTPUT

```

eclipse-workspace - Sample Aviral Project/src/aviralmehra/ThreadSynch.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Quick Access
Sample Aviral Project JRE System Library [JavaSE-10]
src
aviral
c.java
Console.java
DataString.java
Exception2.java
GenThreads.java
Hello_World.java
Inheritance1.java
Multi.java
Multi2.java
Nonum.java
Numb.java
Numx.java
Parent.java
Rjava
SearchString.java
StringUpper.java
W.java
Classandobject
Classes
aviralmehra
fivethreads.java
ThreadSynch.java

```

```

1 package aviralmehra;
2 class ThreadSynch
3 {
4 public static void main(String arg[])throws Exception
5 {
6     data d1=new data();
7     data d2=new data();
8     data d3=new data();
9     data d4=new data();
10    data d5=new data();
11    data d6=new data();
12    data d7=new data();
13    data d8=new data();
14    data d9=new data();
15    data d10=new data();
16    System.out.println(d10.count);
17 }
18 }
19 class item
20 {
21     static int count=0;
}

```

```

Markers Properties Servers Data Source Explorer Snippets Console
<terminated> ThreadSynch [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 15, 2020, 9:40:41 AM)
10

```

Type here to search    Writable    Smart Insert    1:1

ENG US 9:43 AM 10/15/2020

## **TITLE:** Collections

### **1. Write a program for the following:**

**(i) Read all elements from ArrayList by using Iterator.**

#### **CODE**

```

import java.util.ArrayList;
import java.util.Iterator;
public class ArrayListIterator
{
    public static void main(String args[])
    {
        ArrayList<String> arrl = new ArrayList<String>();
        arrl.add("First");
        arrl.add("Second");
        arrl.add("Third");
        arrl.add("Random");
        Iterator<String> itr = arrl.iterator();
        while(itr.hasNext())
        {
            System.out.println(itr.next());
        }
    }
}

```

## }

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Sample Aviral Project/src/aviral/ArrayListIterator.java - Eclipse
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with various icons.
- Project Explorer:** Shows a project named "Sample Aviral Project" containing a "src" folder with multiple Java files, including "ArrayListIterator.java".
- Code Editor:** Displays the code for "ArrayListIterator.java":

```
1 package aviral;
2 import java.util.ArrayList;
3 import java.util.Iterator;
4 public class ArrayListIterator
5 {
6     public static void main(String args[])
7     {
8         ArrayList<String> arrl = new ArrayList<String>();
9         arrl.add("First");
10        arrl.add("Second");
11        arrl.add("Third");
12        arrl.add("Random");
13        Iterator<String> itr = arrl.iterator();
14        while(itr.hasNext())
15        {
16            System.out.println(itr.next());
17        }
18    }
19 }
```
- Console View:** Shows the output of the program:

```
First
Second
Third
Random
```
- Bottom Status Bar:** Shows the system tray, battery level (ENG US), and current time (11:39 AM 10/29/2020).

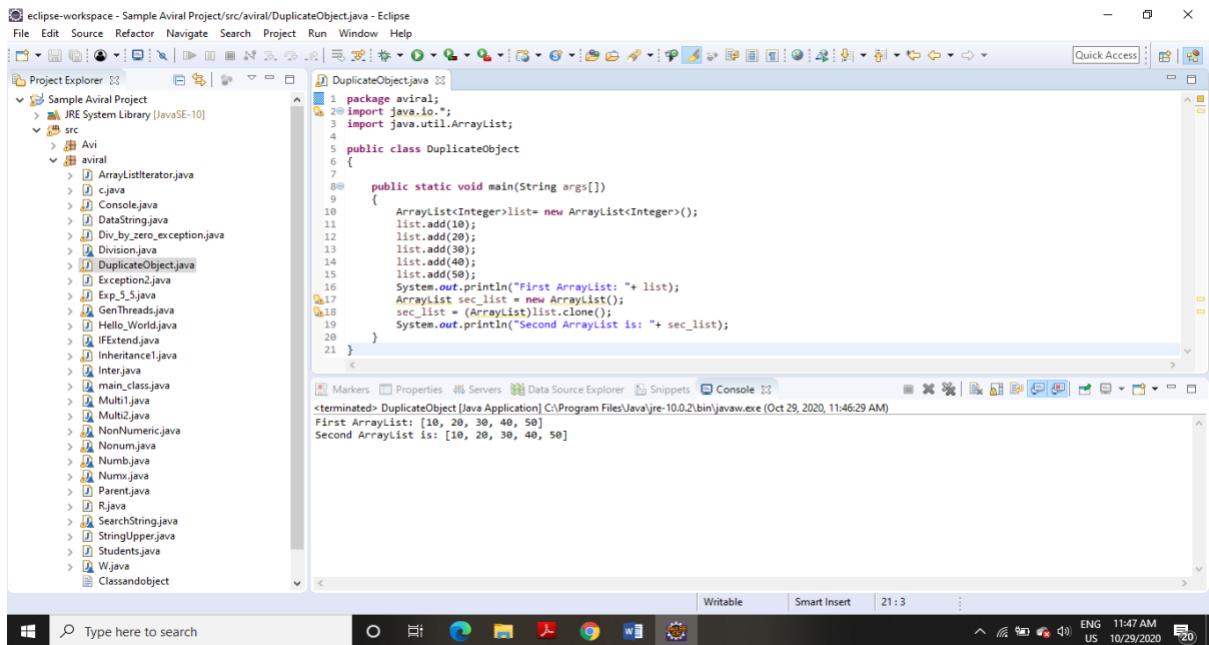
## (ii) Create duplicate object of an ArrayList instance.

## CODE

```
import java.io.*;
import java.util.ArrayList;

public class DuplicateObject
{
    public static void main(String args[])
    {
        ArrayList<Integer>list= new ArrayList<Integer>();
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        System.out.println("First ArrayList: "+ list);
        ArrayList sec_list = new ArrayList();
        sec_list = (ArrayList)list.clone();
        System.out.println("Second ArrayList is: "+ sec_list);
    }
}
```

## OUTPUT



### (iii) Reverse ArrayList content. CODE

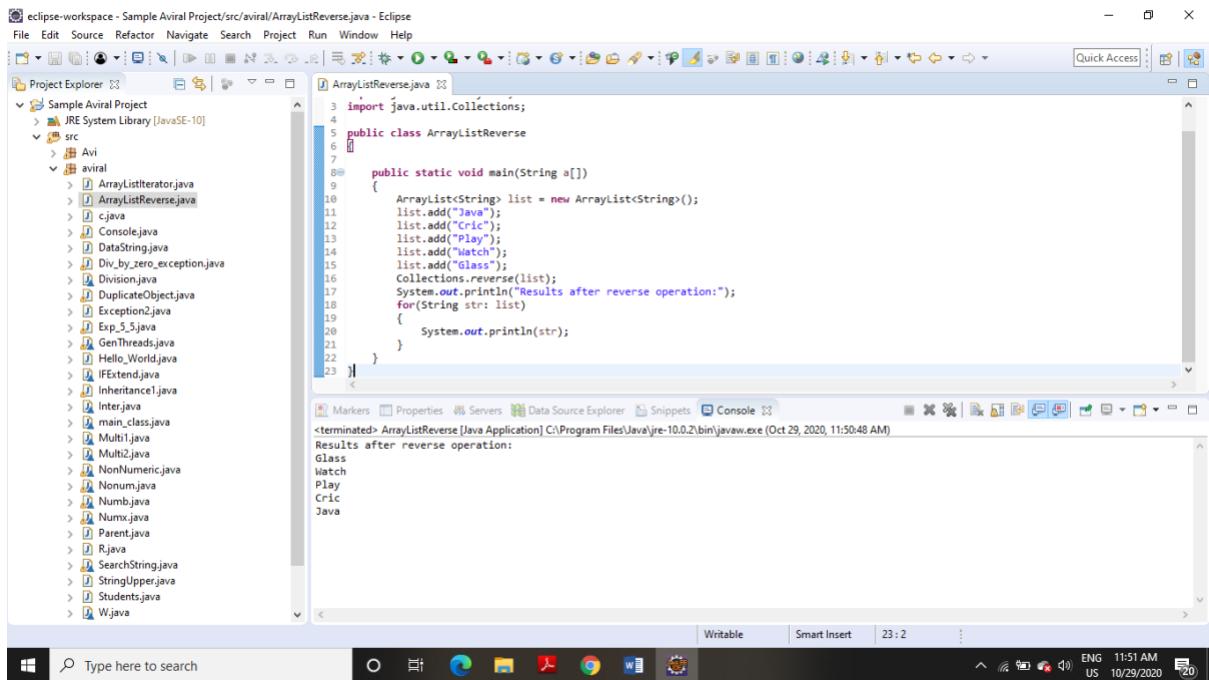
```

import java.util.ArrayList;
import java.util.Collections;

public class ArrayListReverse
{
    public static void main(String a[])
    {
        ArrayList<String> list = new ArrayList<String>();
        list.add("Java");
        list.add("Cric");
        list.add("Play");
        list.add("Watch");
        list.add("Glass");
        Collections.reverse(list);
        System.out.println("Results after reverse operation:");
        for(String str: list)
        {
            System.out.println(str);
        }
    }
}

```

### OUTPUT



## 2. Write a program for the following HashMap:

(i) find whether specified key exists or not.

**CODE**

```

import java.util.*;
public class GFG
{
    public static void main(String[] args)
    {
        HashMap<Integer, String> map = new HashMap<>();
        map.put(1, "Hi");
        map.put(2, "Hello");
        map.put(3, "How are you");
        int keyToBeChecked = 2;
        System.out.println("HashMap: "+ map);
        Iterator<Map.Entry<Integer, String>> iterator = map.entrySet().iterator();
        boolean isKeyPresent = false;
        while (iterator.hasNext())
        {
            Map.Entry<Integer, String> entry=iterator.next();
            if (keyToBeChecked == entry.getKey())
            {
                isKeyPresent = true;
            }
        }
    }
}

```

```

        System.out.println("Does key " + keyToBeChecked + " exists: " +
isKeyPresent);
    }
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface. The left pane displays the Project Explorer with a tree structure of Java files under a project named 'Sample Aviral Project'. The right pane shows the code editor with GFG.java open, containing the provided Java code. Below the code editor is the Console view, which shows the output of running the program. The output reads: <terminated> GFG Java Application C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 29, 2020, 2:33:25 PM) Hashmap: {1=H1, 2=Hello, 3=How are you} Does key 2 exists: true.

```

public static void main(String[] args)
{
    HashMap<Integer, String> map = new HashMap<Integer, String>();
    map.put(1, "H1");
    map.put(2, "Hello");
    map.put(3, "How are you");
    int keyToBeChecked = 2;
    System.out.println("Hashmap: " + map);
    Iterator<Map.Entry<Integer, String>> iterator = map.entrySet().iterator();
    boolean isKeyPresent = false;
    while (iterator.hasNext())
    {
        Map.Entry<Integer, String> entry = iterator.next();
        if (keyToBeChecked == entry.getKey())
        {
            isKeyPresent = true;
        }
    }
    System.out.println("Does key " + keyToBeChecked + " exists: " + isKeyPresent);
}

```

## (ii) find whether specified value exists or not

### CODE

```

import java.util.HashMap;
public class CheckValue
{
    public static void main(String[] args)
    {
        HashMap<Integer, String> hashmap = new HashMap<Integer, String>();
        hashmap.put(11, "Aviral");
        hashmap.put(22, "Mehra");
        hashmap.put(33, "Singh");
        hashmap.put(44, "Rajesh");
        hashmap.put(55, "Kate");
        boolean flag = hashmap.containsValue("Avi");
        System.out.println("String Singh exists in HashMap? : " + flag);
    }
}

```

## OUTPUT

The screenshot shows two instances of the Eclipse IDE interface. Both instances have the same project structure and code editor content, but differ in their console outputs.

**Project Explorer:**

- Sample Aviral Project
- JRE System Library [JavaSE-10]
- src
- Avi
- aviral
- ArrayListIterator.java
- ArrayListReverse.java
- c.java
- CheckValue.java
- Console.java
- DataString.java
- Div\_by\_zero\_exception.java
- Division.java
- DuplicateObject.java
- Exception2.java
- Exp\_5.java
- GenThreads.java
- GFG.java
- Hello\_World.java
- IFExtend.java
- Inheritance1.java
- Inter.java
- main\_class.java
- Multi.java
- Multi2.java
- NonNumeric.java
- Nonum.java
- Numb.java
- Num.java
- Parent.java
- R.java
- SearchString.java
- StringUpper.java

**Code Editor (CheckValue.java):**

```

1 package aviral;
2 import java.util.HashMap;
3 public class CheckValue
4 {
5
6    public static void main(String[] args)
7    {
8        HashMap<Integer, String> hashmap = new HashMap<Integer, String>();
9        hashmap.put(11, "Aviral");
10       hashmap.put(22, "Mehra");
11       hashmap.put(33, "Singh");
12       hashmap.put(44, "Rajesh");
13       hashmap.put(55, "Kate");
14       boolean flag = hashmap.containsKey("Avi");
15       System.out.println("String Singh exists in HashMap? : " + flag);
16   }
17 }
18

```

**Console View Output:**

```
<terminated> CheckValue[Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 29, 2020, 2:39:19 PM)
String Singh exists in HashMap? : false
```

**Console View Output (Second Instance):**

```
<terminated> CheckValue[Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 29, 2020, 2:40:18 PM)
String Singh exists in HashMap? : true
```

**Updates Available Dialog:**

Updates are available for your software. Click to review and install updates. Set up Reminder options.

### (iii) get all keys from the given HashMap

```

import java.util.HashMap;
import java.util.Set;
public class HashMapKeys
{

```

```

public static void main(String a[])
{
    HashMap<String, String> hm = new HashMap<String, String>();

```

```

hm.put("first", "FIRST INSERTED");
hm.put("second", "SECOND INSERTED");
hm.put("third", "THIRD INSERTED");
System.out.println(hm);
Set<String> keys = hm.keySet();
for(String key: keys)
{
    System.out.println(key);
}
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the "Sample Aviral Project" with the "src" folder expanded, displaying numerous Java source files.
- Code Editor:** Displays the content of `HashMapKeys.java`.
- Console View:** Shows the output of the Java application, which includes the insertion of three key-value pairs into a HashMap and the subsequent printing of all keys.
- Status Bar:** Shows system information like battery level, network status, and the date/time (10/29/2020, 2:44 PM).
- Updates Available:** A notification in the bottom right corner indicates updates are available for the software.

```

package aviral;
import java.util.HashMap;
import java.util.Set;
public class HashMapKeys
{
    public static void main(String a[])
    {
        HashMap<String, String> hm = new HashMap<String, String>();
        hm.put("first", "FIRST INSERTED");
        hm.put("second", "SECOND INSERTED");
        hm.put("third", "THIRD INSERTED");
        System.out.println(hm);
        Set<String> keys = hm.keySet();
        for(String key: keys)
        {
            System.out.println(key);
        }
    }
}

```

```

terminated: HashMapKeys [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 29, 2020, 2:43:35 PM)
[third=THIRD INSERTED, first=FIRST INSERTED, second=SECOND INSERTED]
third
first
second

```

### (iv) get all key-value pair as Entry objects CODE

```

import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Set;
public class HashEntrySet
{

```

```

public static void main(String a[])
{
    HashMap<String, String> hm = new HashMap<String, String>();
    hm.put("first", "FIRST INSERTED");
    hm.put("second", "SECOND INSERTED");
    hm.put("third", "THIRD INSERTED");
    System.out.println(hm);
    Set<Entry<String, String>> entires = hm.entrySet();
    for(Entry<String, String> ent:entires)
    {
        System.out.println(ent.getKey()+" ==> "+ent.getValue());
    }
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "Sample Aviral Project" containing several Java files under the "src" folder.
- Editor View:** Displays the code for HashEntrySet.java.
- Console View:** Shows the output of the program execution. The output text is:

```

terminated: HashEntrySet [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 29, 2020, 2:47:18 PM)
[third==>THIRD INSERTED, first==>FIRST INSERTED, second==>SECOND INSERTED]
first ==> FIRST INSERTED
second ==> SECOND INSERTED

```
- Status Bar:** Shows system information like "ENG US 2:47 PM 10/29/2020".

### 3. Write a program for the following:

**(i) HashSet copy another collection object to HashSet object.**  
**CODE**

```

import java.util.HashSet;
public class HashSetCopy
{
    public static void main(String a[])
    {
}

```

```

{
    HashSet<String> hs = new HashSet<String>();
    hs.add("first");
    hs.add("second");
    hs.add("third");
    System.out.println(hs);
    HashSet<String> subSet = new HashSet<String>();
    subSet.add("s1");
    subSet.add("s2");
    hs.addAll(subSet);
    System.out.println("HashSet content after adding another collection:");
    System.out.println(hs);
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "Sample Aviral Project".
- Code Editor:** Displays the content of `HashSetCopy.java`.
- Console View:** Shows the terminal output of the application's execution.
- Taskbar:** Includes the Windows taskbar at the bottom with various icons and system status.

```

eclipse-workspace - Sample Aviral Project/src/aviral/HashSetCopy.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Properties Servers Data Source Explorer Snippets Console
HashSetCopy.java
1 package aviral;
2 import java.util.HashSet;
3 public class HashSetCopy
4 {
5     public static void main(String a[])
6     {
7         HashSet<String> hs = new HashSet<String>();
8         hs.add("first");
9         hs.add("second");
10        hs.add("third");
11        System.out.println(hs);
12        HashSet<String> subSet = new HashSet<String>();
13        subSet.add("s1");
14        subSet.add("s2");
15        hs.addAll(subSet);
16        System.out.println("HashSet content after adding another collection:");
17        System.out.println(hs);
18    }
19 }

terminated: HashSetCopy [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Nov 2, 2020, 10:41:00 AM)
[third, first, second]
HashSet content after adding another collection:
[third, first, s1, second, s2]

```

## (ii) delete all entries at one call from HashSet CODE

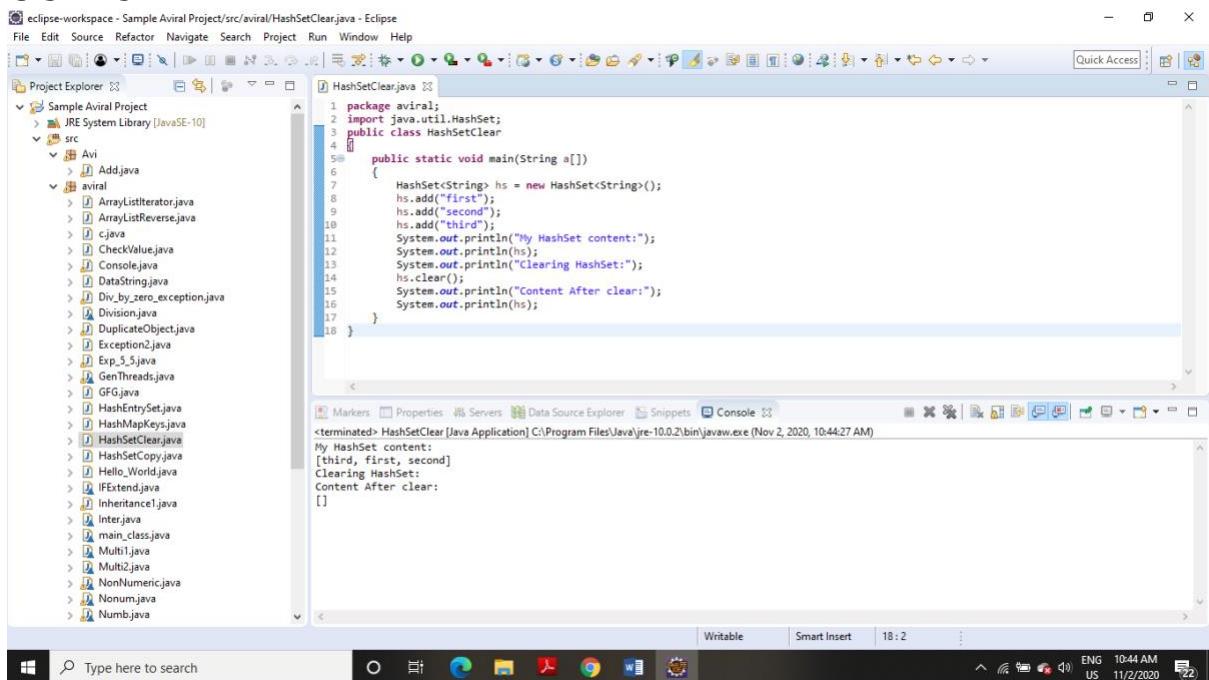
```

import java.util.HashSet;
public class HashSetClear
{
    public static void main(String a[])
    {

```

```
    HashSet<String> hs = new HashSet<String>();
    hs.add("first");
    hs.add("second");
    hs.add("third");
    System.out.println("My HashSet content:");
    System.out.println(hs);
    System.out.println("Clearing HashSet:");
    hs.clear();
    System.out.println("Content After clear:");
    System.out.println(hs);
}
```

## OUTPUT



**(iii) search user defined objects from HashSet  
CODE**

```
import java.util.HashSet;
public class HashSetSearchObject
{
    public static void main(String a[])
    {
        HashSet<Price> lhs = new HashSet<Price>();
```

```

lhs.add(new Price("Banana", 20));
lhs.add(new Price("Apple", 40));
lhs.add(new Price("Orange", 30));
for(Price pr:lhs)
{
    System.out.println(pr);
}
Price key = new Price("Banana", 20);
System.out.println("Does set contains key? "+lhs.contains(key));
}

class Price
{

private String item;
private int price;

public Price(String itm, int pr)
{
    this.item = itm;
    this.price = pr;
}

public int hashCode()
{
    System.out.println("In hashCode");
    int hashcode = 0;
    hashcode = price*20;
    hashcode += item.hashCode();
    return hashcode;
}

public boolean equals(Object obj)
{
    System.out.println("In equals");
    if (obj instanceof Price)
    {
        Price pp = (Price) obj;
        return (pp.item.equals(this.item) && pp.price == this.price);
    } else
    {
        return false;
    }
}

```

```

        }
    }

public String getItem()
{
    return item;
}

public void setItem(String item)
{
    this.item = item;
}

public int getPrice()
{
    return price;
}

public void setPrice(int price)
{
    this.price = price;
}

public String toString()
{
    return "item: "+item+" price: "+price;
}
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the Sample Aviral Project with various Java files listed under the src and aviral packages.
- Code Editor:** Displays the HashSetSearchObject.java file with the provided code.
- Console View:** Shows the output of the application. It includes several 'In hashCode' messages, three 'item: price:' pairs (Apple, Orange, Banana), and a final message 'Does set contains key? true'.
- Bottom Status Bar:** Shows system information like battery level, signal strength, and date/time (11/2/2020, 10:50 AM).

```

eclipse-workspace - Sample Aviral Project/src/aviral/HashSetSearchObject.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer   HashSetSearchObject.java
Sample Aviral Project
  JRE System Library [JavaSE-10]
  src
    Avi
      Add.java
  aviral
    ArrayListIterator.java
    ArrayListReverse.java
    c.java
    CheckValue.java
    Console.java
    DataString.java
    Div_by_zero_exception.java
    Division.java
    DuplicateObject.java
    Exception.java
    Exp_5.java
    GenThreads.java
    GFG.java
    HashEntrySet.java
    HashMapKeys.java
    HashSetClear.java
    HashSetCopy.java
    HashSetSearchObject.java
    Hello_World.java
    IExtend.java
    Inheritance1.java
    Inter.java
    main_class.java
    Multi1.java
    Multi2.java
    NonNumeric.java
    Nonum.java
Markers Properties Servers Data Source Explorer Snippets Console
<terminated> HashSetSearchObject [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Nov 2, 2020, 10:50:04 AM)
In hashCode
In hashCode
In hashCode
item: Apple price: 40
item: Orange price: 30
item: Banana price: 20
In hashCode
In hashCode
Does set contains key? true

```

## **EXPERIMENT NO-5**

**Title:** Interface

**1. Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt. In this class use the object of arithmetic class.**

**CODE:**

```
interface test
{
    int square();
}

class arithmetic implements test
{
    int b;

    arithmetic(int x)
    {
        b = x;
    }

    public int square()
    {
        return (b*b);
    }
}

class ToTestInt
{
    public int return_ans(int x)
```

```

    {
        arithmetic a=new arithmetic(x);
        return a.square();
    }
}

class main_class
{
    public static void main(String []args)
    {
        ToTestInt x= new ToTestInt();
        System.out.println("\nThe square of 64 is "+x.return_ans(64));
    }
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like arithmetic.java, ToTestInt.java, and main\_class.java.
- Code Editor:** Displays the Java code for main\_class.java.
- Console:** Shows the output of the program execution: "The square of 64 is 4096".
- Taskbar:** Shows the Windows taskbar with various application icons and system status.

```

17     {
18         return (b*b);
19     }
20
21 }
22 class ToTestInt
23 {
24     public int return_ans(int x)
25     {
26         arithmetic a=new arithmetic(x);
27         return a.square();
28     }
29 }
30 class main_class
31 {
32     public static void main(String []args)
33     {
34         ToTestInt x= new ToTestInt();
35         System.out.println("\nThe square of 64 is "+x.return_ans(64));
36     }
37 }

```

**2. Write a program to create interface A, in this interface we have two method meth1 and meth2. Implements this interface in another class named MyClass.**

## CODE

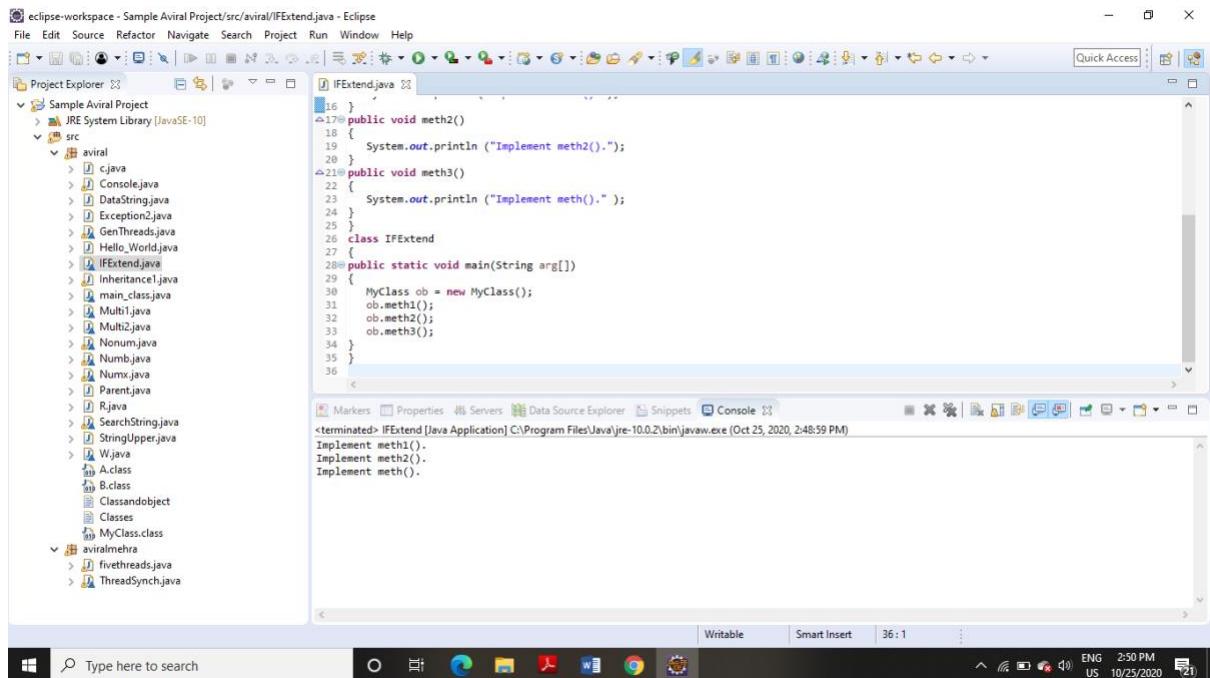
### interface A

```

{
    void meth1();
    void meth2();
}
interface B extends A
{
    void meth3();
}
class MyClass implements B
{
    public void meth1 ()
    {
        System.out.println("Implement meth1().");
    }
    public void meth2()
    {
        System.out.println ("Implement meth2().");
    }
    public void meth3()
    {
        System.out.println ("Implement meth().");
    }
}
class IFEExtend
{
    public static void main(String arg[])
    {
        MyClass ob = new MyClass();
        ob.meth1();
        ob.meth2();
        ob.meth3();
    }
}

```

## OUTPUT



### 3. Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program

## CODE

**interface** area

```
{
    static final float pi=3.142f;
    float compute(float x,float y);
}
class rectangle implements area
{
    public float compute(float x,float y)
    {
        return(x*y);
    }
}
```

```

}
}

class circle implements area
{
    public float compute(float x,float y)
    {
        return(pi*x*x);
    }
}

class Inter
{
    public static void main(String args[])
    {
        rectangle rect=new rectangle();
        circle cr=new circle();
        area ar;
        ar=rect;
        System.out.println("Area of the rectangle= "+ar.compute(10,20));
        ar=cr;
        System.out.println("Area of the circle= "+ar.compute(10,0));
    }
}

```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like Sample Aviral Project, JRE System Library [JavaSE-10], aviral, and various Java source files.
- Editor:** Displays the content of the Inter.java file.
- Console:** Shows the terminal output of the application's execution.
- Bottom Status Bar:** Provides system information such as battery level, network status, and system date/time.

```

eclipse-workspace - Sample Aviral Project/src/aviral/Inter.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer ▾
  Sample Aviral Project
    > JRE System Library [JavaSE-10]
    > src
      > aviral
        > cjava
        > Console.java
        > DataString.java
        > Exception2.java
        > GenThreads.java
        > Hello_World.java
        > IFExtend.java
        > Inheritance1.java
        > Inter.java
        > main_class.java
        > Multi1.java
        > Multi2.java
        > Nonum.java
        > Numb.java
        > Num.java
        > Parent.java
        > R.java
        > SearchString.java
        > StringUpper.java
        > W.java
        > area.class
        > circle.class
        > Classandobject
        > Classes
        > rectangle.class
        > aviralmehra
        > fivethreads.java
        > ThreadSync.java

```

```

Inter.java
14   class circle implements area
15   {
16       public float compute(float x,float y)
17   {
18       return(pi*x*x);
19   }
20   }
21   class Inter
22   {
23       public static void main(String args[])
24   {
25       rectangle rect=new rectangle();
26       circle cr=new circle();
27       area ar;
28       ar=rect;
29       System.out.println("Area of the rectangle= "+ar.compute(10,20));
30       ar=cr;
31       System.out.println("Area of the circle= "+ar.compute(10,0));
32   }
33   }
34

Markers Properties Servers Data Source Explorer Snippets Console
<terminated> Inter [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 25, 2020, 2:52:25 PM)
Area of the rectangle= 200.0
Area of the circle= 314.2

```

**4. Write a program to create an Interface having two methods division and modules. Create a class, which overrides these methods.**

## CODE

```
interface course
{
    void division(int a);
    void modules(int b);
}
class stud implements course
{
    String name;
    int div,mod;
    void name(String n)
    {
        name=n;
    }
    public void division(int a)
    {
        div=a;
    }
    public void modules(int b)
    {
        mod=b;
    }
    void disp()
    {
        System.out.println("Name :" +name);
        System.out.println("Division :" +div);
        System.out.println("Modules :" +mod);
    }
}
class Division
{
    public static void main(String args[])
    {
        stud s=new stud();
        s.name("Arun");
        s.division(5);
        s.modules(15);
        s.disp();
    }
}
```

}

## OUTPUT

The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left lists a project named 'Sample Aviral Project' containing several Java files under the 'src' folder, including 'Division.java'. The 'Division.java' file is open in the editor, showing code that creates a 'Division' object and prints its name, division, and modules. The 'Console' view at the bottom shows the output of running the program, which includes the printed values: Name : Arun, Division : 5, and Modules : 15.

```
20 {
21     mod=b;
22 }
23 void disp()
24 {
25     System.out.println("Name :" +name);
26     System.out.println("Division :" +div);
27     System.out.println("Modules :" +mod);
28 }
29 }
30 class Division
31 {
32     public static void main(String args[])
33     {
34         stud = new stud();
35         s.name("Arun");
36         s.division(5);
37         s.modules(15);
38         s.disp();
39     }
40 }
```

Markers Properties Servers Data Source Explorer Snippets Console

<terminated> Division [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Oct 25, 2020, 2:55:51 PM)

Name :Arun  
Division :5  
Modules :15

## EXPERIMENT NO-10

### TITLE: JDBC

**1. Create a database table to store the records of employee in a company. Use getConnection function to connect the database. The statement object uses executeUpdate function to create a table.**

### CODE

```
import java.sql.*;
class JDBC
{
    public static void main(String args[])
    {
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306","root","avir
al08");
```

```

System.out.println("Connection Successful");
Statement stmt = conn.createStatement();
stmt.execute("use student");
stmt.executeUpdate("create table Employees(empid varchar(10) primary
key,empname varchar(45),city varchar(45),salary varchar(45))");
System.out.println("table is created");
stmt.executeUpdate("insert into Employees values('R110219031','Aviral
Mehra','Dehradun','25000')");
stmt.executeUpdate("insert into Employees
values('R110217685','Rajesh','Delhi','35000')");
System.out.println("value is Update");
ResultSet rs=stmt.executeQuery("select * from Employees");
while(rs.next()) {
    String mempid=rs.getString(1);
    String mempname=rs.getString(2);
    String mcity=rs.getString(3);
    String msalary=rs.getString(4);
    System.out.println(mempid+ " "+mempname+ " "+mcity+
"+msalary);

}
conn.close();
}
catch(SQLException e){
    System.out.println(e);
}
catch(ClassNotFoundException e) {
    System.out.println(e);
}
}
}

OUTPUT

```

```

eclipse-workspace - Sample Aviral Project/src/aviral/JDBC.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer JDBC.java
src
  > Avi
  > aviral
  > aviralmehra
  > JRE System Library [JavaSE-10]
  > Referenced Libraries
13     stat.executeUpdate("Create table Employees(empid varchar(10) primary key,empname varchar(45),city varchar(45),salary varchar(45))");
14     System.out.println("Table is created");
15     stat.executeUpdate("Insert into Employees values('R110219031','Aviral Mehra','Dehradun','25000')");
16     stat.executeUpdate("Insert into Employees values('R110217685','Rajesh','Delhi','35000')");
17     System.out.println("Value is Update");
18     ResultSet rs=stmt.executeQuery("select * from Employees");
19     while(rs.next()) {
20         String empid=rs.getString(1);
21         String empname=rs.getString(2);
22         String mcity=rs.getString(3);
23         String msalary=rs.getString(4);
24         System.out.println(empid+ " "+empname+" "+mcity+" "+msalary);
25     }
26 }
27 }
28 catch(SQLException e){
29     System.out.println(e);
30 }
31 catch(ClassNotFoundException e) {
32     System.out.println(e);
33 }
34 }
35 }
36 }
37 }
38 }

```

Console output:

```

<terminated> JDBC [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Nov 19, 2020, 9:04:28 PM)
Connection Successful
table is created
value is Update
R110217685 Rajesh Delhi 35000
R110219031 Aviral Mehra Dehradun 25000

```

**2. Create a database of employee of company in mysql and then use java program to access the database for inserting information of employees in database. The SQL statement can be used to view the details of the data of employees in the database.**

```

import java.sql.*;
class Employee2
{
    public static void main(String s[])
    {
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306","root","avir
al08");
            System.out.println("Connection Successful");
            Statement stmt = conn.createStatement();
            stmt.execute("use student");
            stmt.executeUpdate("insert into employees1
values('435325','Mohan','Allahabad','25000')");
            stmt.executeUpdate("insert into employees1
values('435326','sohan','Allahabad','35000')");
            System.out.println("value is Update");
            conn.close();
        }
    }
}

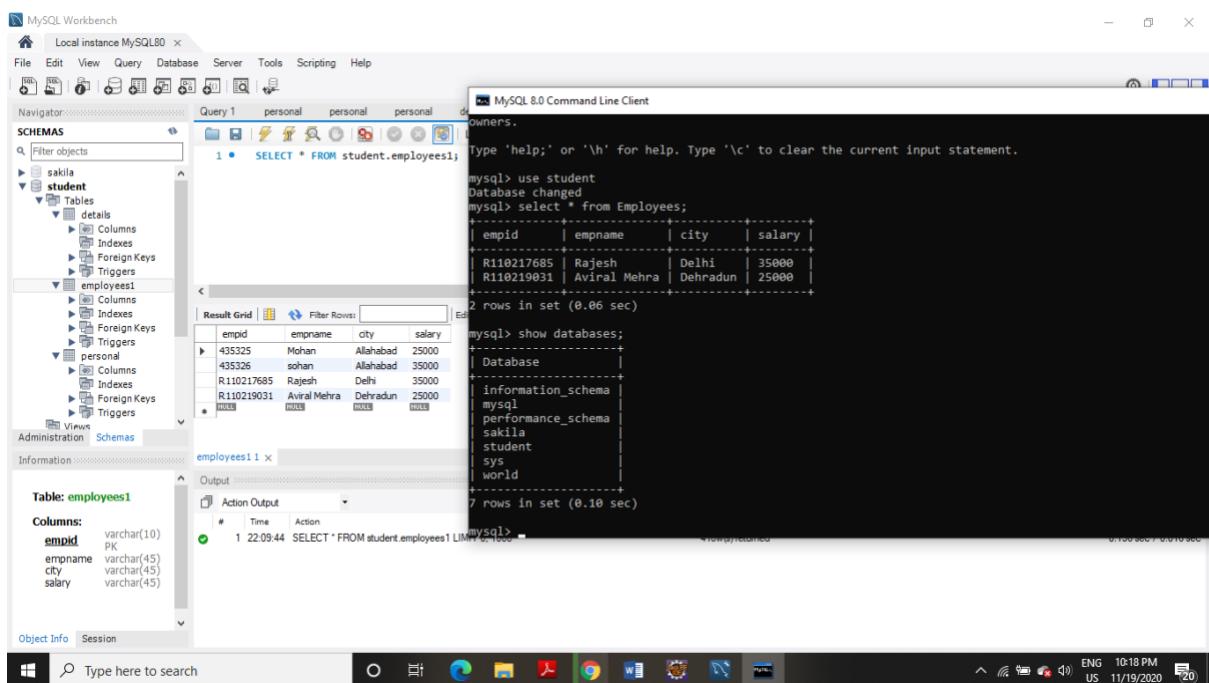
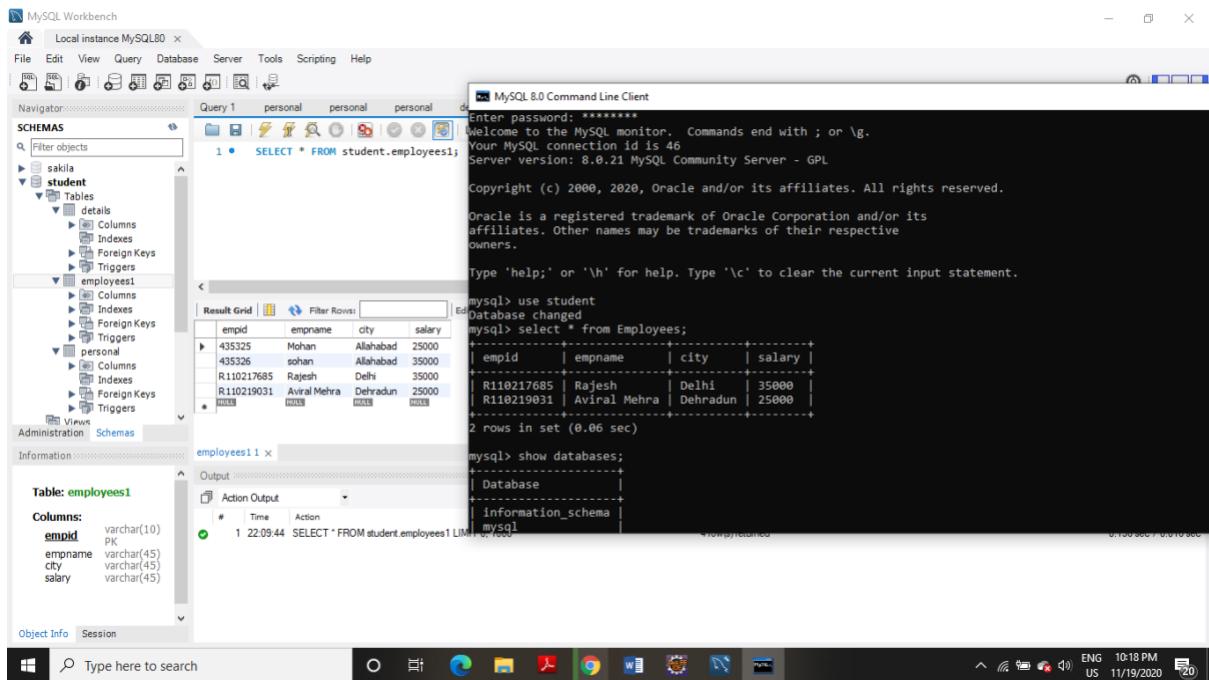
```

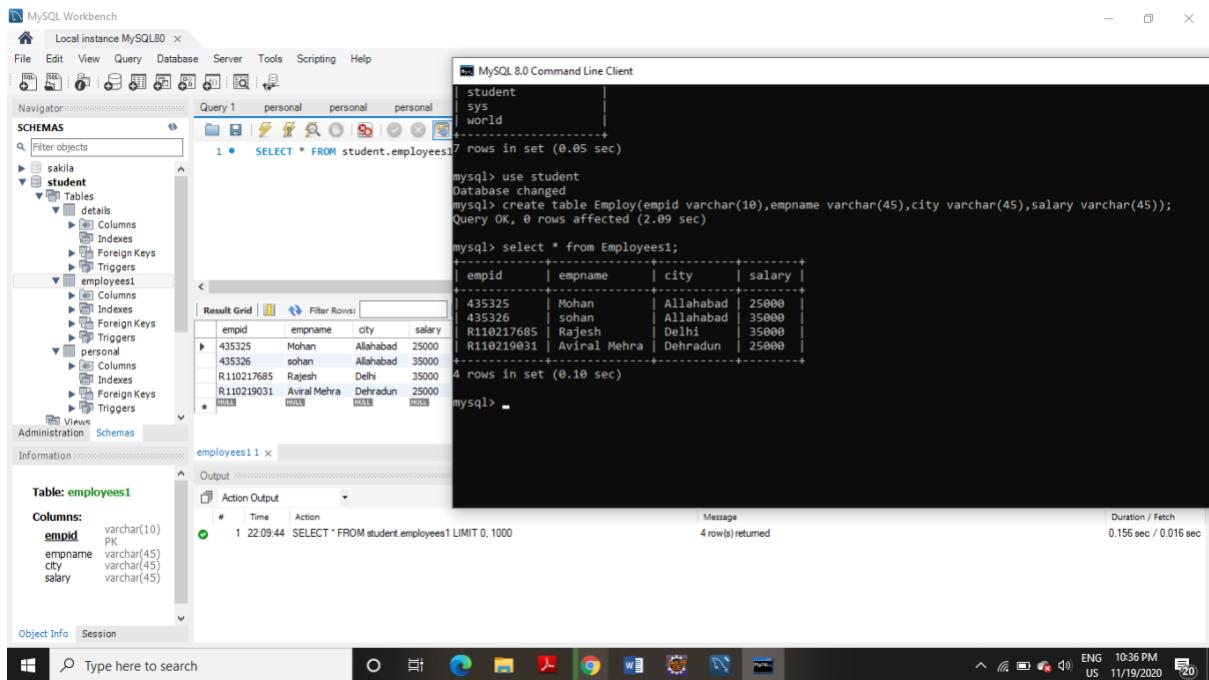
```
        }  
    }  
    catch(SQLException e){  
        System.out.println(e);  
    }  
    catch(ClassNotFoundException e) {  
        System.out.println(e);  
    }  
}
```

## OUTPUT

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Sample Aviral Project/src/aviral/Employee2.java - Eclipse
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, etc.
- Project Explorer:** Shows the project structure with files like aviral, ArrayListIterator.java, ArrayListReverse.java, c.java, CheckValue.java, Console.java, DataString.java, Dir\_by\_zero\_exception.java, Division.java, DuplicateObject.java, Employee2.java, Exception2.java, Exp\_5\_5.java, GenThreads.java, GFJava, HashEntrySet.java, HashMapKeys.java, HashSetClear.java, HashSetCopy.java, HashSetSearchObject.java, Hello\_World.java, IFFExtend.java, Inheritance1.java, Inter.java, JDBC.java, main\_class.java, Multi1.java, Multi2.java, NonNumeric.java, Noun.java, Numb.java, Num.java, Parent.java.
- Code Editor:** Displays the Java code for Employee2.java, which uses JDBC to connect to a MySQL database and execute SQL statements to insert and update data in the employees table.
- Output View:** Shows the terminal output with the message "Connection Successful" and "value is Update".
- Bottom Status Bar:** Writable, Smart Insert, 15:88, 9:37 PM, ENG US, 11/19/2020.





## EXPERIMENT NO-11

### TITLE: Servlet

**1. Servlet: a) ServletContext interface b)getParameterValues( ) of Servlet Request.**

(a)

**CODE:**

**SERVLET**

```

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

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

        //creating ServletContext object
        ServletContext context=getServletContext();
    }
}

```

```

//Getting the value of the initialization parameter and printing it
String driverName=context.getInitParameter("dname");
pw.println("Name of student is: "+driverName);

pw.close();

}}

```

## 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_3_0.xsd" id="WebApp_ID"
  version="3.0">
  <display-name>avi</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>Scontext</servlet-name>
    <servlet-class>Scontext</servlet-class>
  </servlet>
  <context-param>
    <param-name>dname</param-name>
    <param-value>Aviral Mehra</param-value>
  </context-param>
  <servlet-mapping>
    <servlet-name>Scontext</servlet-name>
    <url-pattern>/Scontext</url-pattern>
  </servlet-mapping>
</web-app>

```

eclipse-workspace - avi/src/Scontext.java - Eclipse IDE

```
1 import java.io.*;
2 import javax.servlet.*;
3 import javax.servlet.http.*;
4 
5 public class Scontext extends HttpServlet{
6     @Override
7     protected void doGet(HttpServletRequest req,HttpServletResponse res)
8     {
9         res.setContentType("text/html");
10        PrintWriter pw=res.getWriter();
11 
12        //creating ServletContext object
13        ServletContext context=getServletContext();
14 
15        //Getting the value of the initialization parameter and printing it
16        String driverName=context.getInitParameter("dname");
17        pw.println("Name of student is: "+driverName);
18 
19        pw.close();
20    }
21 }
```

Project Explorer    Scontext.java    web.xml    http://localhost:8080/avi/Scontext

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Type here to search    Writable Smart Insert 16 : 53 : 526

ENG US 7:20 PM 12/3/2020

eclipse-workspace - avi/WebContent/WEB-INF/web.xml - Eclipse IDE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <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_3_0.xsd">
3     <display-name>avi</display-name>
4     <welcome-file-list>
5         <welcome-file>index.html</welcome-file>
6         <welcome-file>index.htm</welcome-file>
7         <welcome-file>index.jsp</welcome-file>
8         <welcome-file>default.html</welcome-file>
9         <welcome-file>default.htm</welcome-file>
10        <welcome-file>default.jsp</welcome-file>
11        <welcome-file>list</welcome-file>
12    <!-->
13    <!-->
14    <!-->
15    <!-->
16    <!-->
17    <!-->
18    <!-->
19    <!-->
20    <!-->
21    <!-->
22    <!-->
23    <!-->
24 </web-app>
```

Project Explorer    Scontext.java    web.xml    http://localhost:8080/avi/Scontext

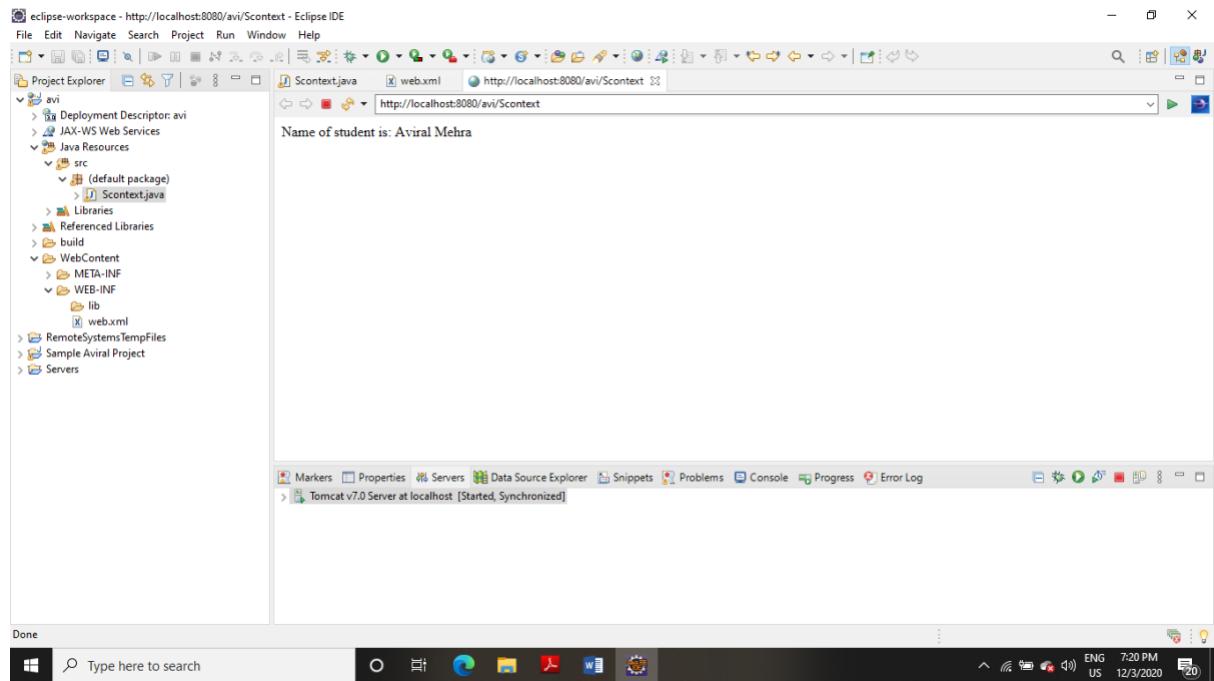
Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Type here to search    Writable Smart Insert 15 : 13 : 750

ENG US 7:20 PM 12/3/2020

## OUTPUT



(b)

## CODE:

### SERVLET

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
public class Scontext extends HttpServlet
{
    public void doGet(HttpServletRequest req,HttpServletResponse res)
    throws ServletException,IOException
    {
        res.setContentType("text/html");
        PrintWriter pwrtwr=req.getWriter();
        String name = req.getParameter("uname");
        String age = req.getParameter("uage");
        pwrtwr.println("Name: "+name);
        pwrtwr.println("Age: "+age);
        pwrtwr.close();
    }
}
```

## 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_3_0.xsd" id="WebApp_ID"
  version="3.0">
  <display-name>avi</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>BeginnersBook</servlet-name>
    <servlet-class>Scontext</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>BeginnersBook</servlet-name>
    <url-pattern>/details</url-pattern>
  </servlet-mapping>
</web-app>
```

## Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="details" method="get">
  User Name: <input type="text" name="uname"><br>
  User Age: <input type="text" name="uage"><br>
  <input type="submit" value="submit">
</form>
</body>
</html>
```

eclipse-workspace - avi/src/Scontext.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer Scontext.java web.xml index.html http://localhost:8080/avi/details?uname=Aviral+Mehra&age=20

```
1 import javax.servlet.http.*;
2 import javax.servlet.*;
3 import java.io.*;
4 
5 public class Scontext extends HttpServlet
6 {
7     public void doGet(HttpServletRequest req,HttpServletResponse res)
8         throws ServletException,IOException
9     {
10        res.setContentType("text/html");
11        PrintWriter pwrtwr=res.getWriter();
12        String name = req.getParameter("uname");
13        String age = req.getParameter("age");
14        pwrtwr.println("Name: "+name);
15        pwrtwr.println("Age: "+age);
16        pwrtwr.close();
17    }
18 }
19
```

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Type here to search Writable Smart Insert 19 : 1 : 530 ENG US 11:42 AM 12/4/2020 20

eclipse-workspace - avi/WebContent/WEB-INF/web.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help

Project Explorer Scontext.java web.xml index.html http://localhost:8080/avi/details?uname=Aviral+Mehra&age=20

```
<?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_3_0.xsd">
<display-name>avi</display-name>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
< servlets>
< servlet-name>BeginnersBook</servlet-name>
< servlet-class>Scontext</servlet-class>
</ servlet>
< servlet-mapping>
< servlet-name>BeginnersBook</servlet-name>
< url-pattern>/details</url-pattern>
</ servlet-mapping>
</ web-app>
```

Design Source

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Type here to search Writable Smart Insert 21 : 1 : 888 ENG US 11:42 AM 12/4/2020 20

The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left displays a Java project named 'avi'. The 'src' folder contains 'AddServlet.java', 'Scontext.java', and 'Scontext'. The 'WebContent' folder contains 'index.html', 'lib', and 'web.xml'. The code editor window shows the content of Scontext.java:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <form action="details" method="get">
9 User Name: <input type="text" name="uname"><br>
10 User Age: <input type="text" name="uage"><br>
11 <input type="submit" value="submit">
12 </form>
13
14 </body>
15 </html>
```

The browser window at the top right shows the URL <http://localhost:8080/avi/details?uname=Aviral+Mehra&uage=20>. The status bar at the bottom indicates the time is 11:42 AM on 12/4/2020.

## OUTPUT

The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left displays a Java project named 'avi'. The 'src' folder contains 'AddServlet.java', 'Scontext.java', and 'Scontext'. The 'WebContent' folder contains 'index.html', 'lib', and 'web.xml'. The code editor window shows the content of Scontext.java:

```
Name: Aviral Mehra Age: 20
```

The browser window at the top right shows the URL <http://localhost:8080/avi/details?uname=Aviral+Mehra&uage=20>. The status bar at the bottom indicates the time is 11:42 AM on 12/4/2020.

## 2. Write a Servlet page to display current date of the server.

### CODE:

#### SERVLET

```
import java.io.*;
import javax.servlet.*;
public class Date1 extends GenericServlet{
public void service(ServletRequest req, ServletResponse res)
throws IOException, ServletException
{
    res.setContentType("text/html");
    PrintWriter pw = res.getWriter();
    java.util.Date date = new java.util.Date();
    pw.println("Current Date " +date.toString());

    pw.close();
}
}
```

#### 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_3_0.xsd" id="WebApp_ID"
version="3.0">
<display-name>avi</display-name>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<servlet>
<servlet-name>Date1</servlet-name>
<servlet-class>Date1</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Date1</servlet-name>
```

```

<url-pattern>/Date1</url-pattern>
</servlet-mapping>
</web-app>

```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "avi". The "src" folder contains Java files: Date1.java, AddServlet.java, and Scontext.java.
- Date1.java Content:**

```

1 import java.io.*;
2 import javax.servlet.*;
3 public class Date1 extends GenericServlet{
4     public void service(ServletRequest req, ServletResponse res)
5         throws IOException, ServletException
6     {
7         res.setContentType("text/html");
8         PrintWriter pw = res.getWriter();
9         java.util.Date date = new java.util.Date();
10        pw.println("Current Date " +date.toString());
11        pw.close();
12    }
13}
14
15
16

```
- Browser Output:** The browser window displays the output of the servlet: "Current Date Sun Apr 12 11:53:00 US 2020".
- System Tray:** Shows the date and time as "11:53 AM 12/4/2020".

The screenshot shows the Eclipse IDE interface with the following details:

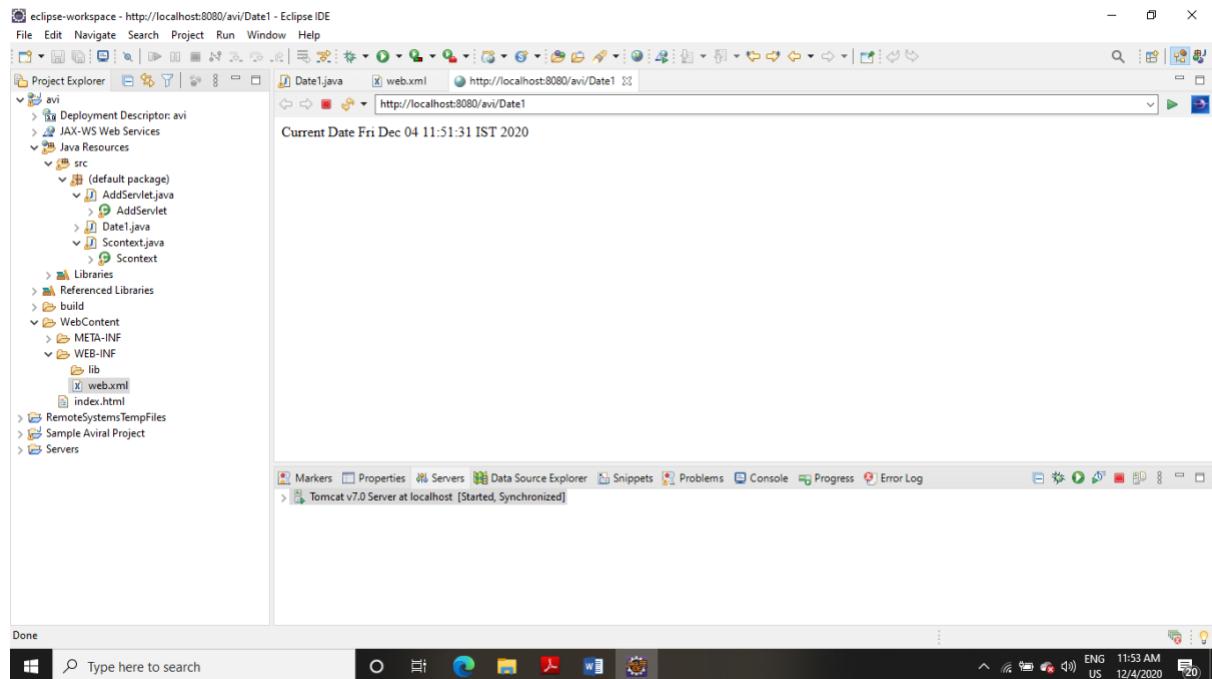
- Project Explorer:** Shows the project structure for "avi". The "WEB-INF" folder contains "web.xml".
- web.xml Content:**

```

<?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_3_0.xsd" version="3.0">
    <display-name>avi</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>Date1</servlet-name>
        <servlet-class>Date1</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Date1</servlet-name>
        <url-pattern>/Date1</url-pattern>
    </servlet-mapping>
</web-app>

```
- System Tray:** Shows the date and time as "11:53 AM 12/4/2020".

## OUTPUT



**3. Write a Servlet page to which include the two other Servlet page through of include directives feature provided in Servlet.**

### CODE:

#### SERVLET

##### first.jsp

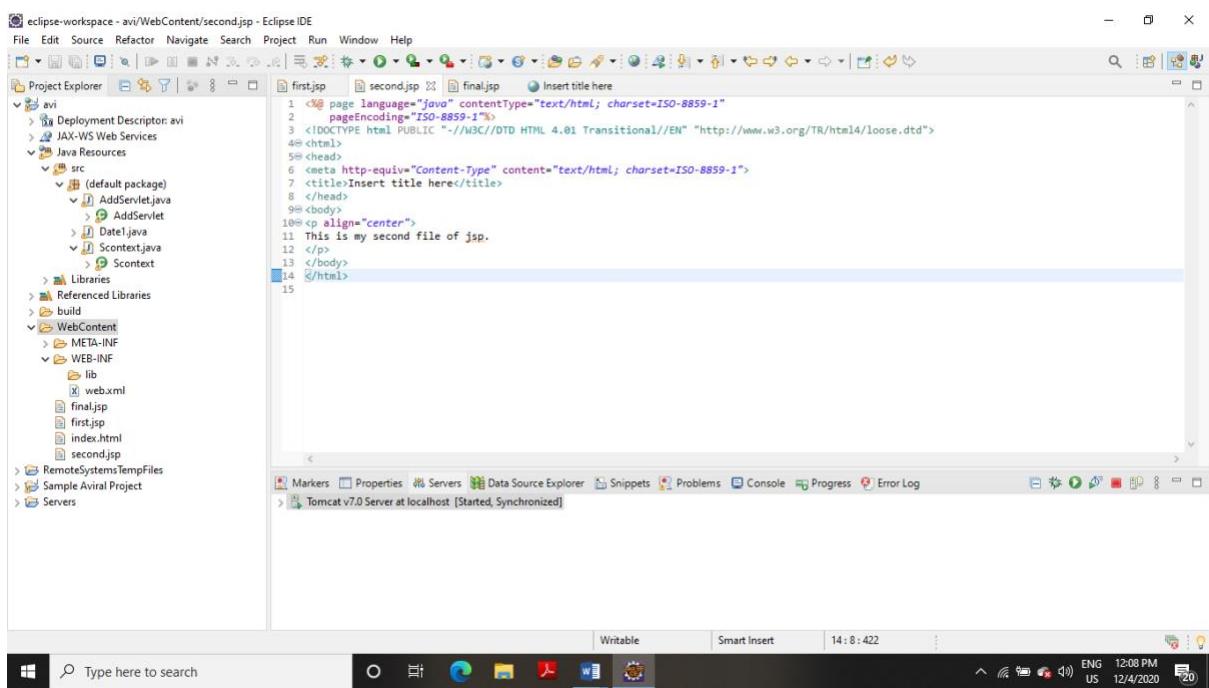
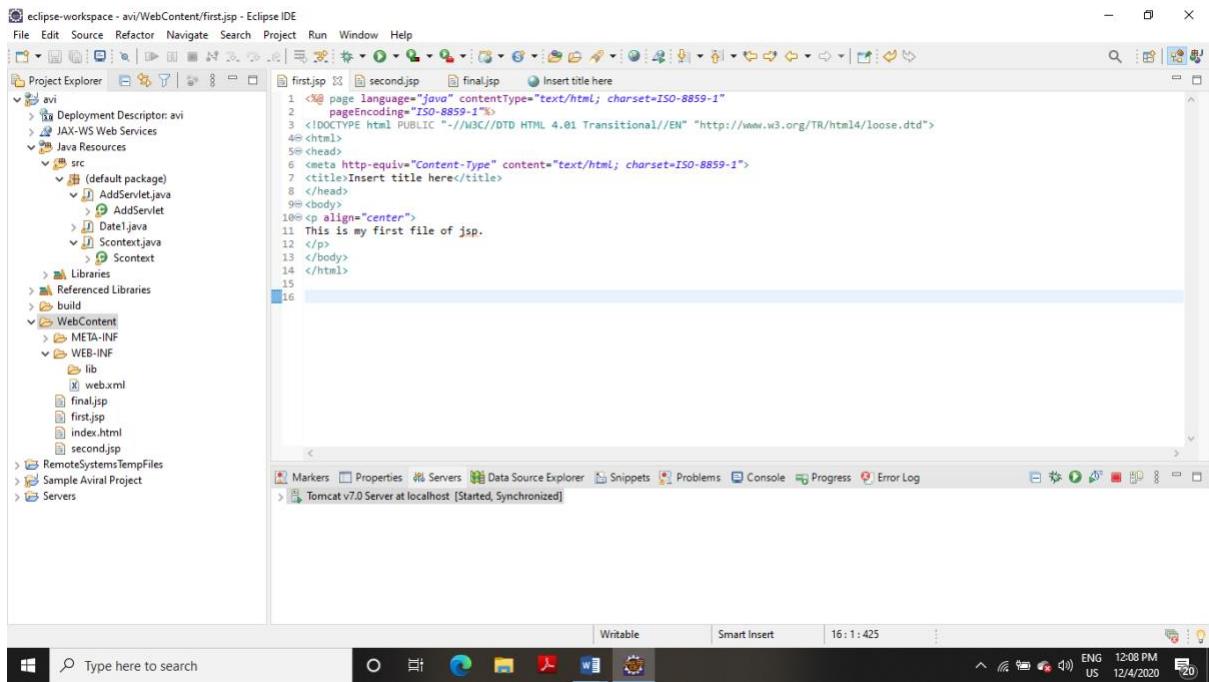
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<p align="center">
This is my first file of jsp.
</p>
</body>
</html>
```

## **Second.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!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>Insert title here</title>
</head>
<body>
<p align="center">
This is my second file of jsp.
</p>
</body>
</html>
```

## **Final.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!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>Insert title here</title>
</head>
<body>
<%@ include file="first.jsp" %>
<h3>joining two jsp file </h3>
<%@ include file="second.jsp" %>
</body>
</html>
```



**eclipse-workspace - avi/WebContent/final.jsp - Eclipse IDE**

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer    first.jsp second.jsp final.jsp Insert title here

```

1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4<html>
5<head>
6 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7 <title>Insert title here</title>
8</head>
9<body>
10<%@ include file="first.jsp" %>
11<h3>joining two jsp file </h3>
12<%@ include file="second.jsp" %>
13</body>
14</html>
15

```

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Type here to search    Writable Smart Insert 15 : 1 : 465

ENG US 12:08 PM 12/4/2020

## OUTPUT

**eclipse-workspace - http://localhost:8080/avi/final.jsp - Eclipse IDE**

File Edit Navigate Search Project Run Window Help

Project Explorer    first.jsp second.jsp final.jsp Insert title here

This is my first file of jsp.

joining two jsp file

This is my second file of jsp.

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Type here to search

Done

ENG US 12:08 PM 12/4/2020

#### 4. Write a Servlet page to create a simple calculator.

##### CODE

###### SERVLET

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

public class Calculator extends HttpServlet
{
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
    }
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException
    {
        PrintWriter out = response.getWriter();
        String n1 = request.getParameter("txt1");
        String n2 = request.getParameter("txt2");
        String op = request.getParameter("op");
        switch (op)
        {
            case "Addition":
                out.println("Answer = "+(Integer.parseInt(n1) +
Integer.parseInt(n2)));
                break;
            case "Subtraction":
                out.println("Answer = "+(Integer.parseInt(n1) -
Integer.parseInt(n2)));
                break;
            case "multiplication":
                out.println("Answer = "+(Integer.parseInt(n1) *
Integer.parseInt(n2)));
                break;
            default:
                out.println("Answer = "+(Integer.parseInt(n1) /
Integer.parseInt(n2)));
        }
    }
}
```

```
        break;  
    }  
}  
}
```

### 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_3_0.xsd" id="WebApp_ID"  
         version="3.0">  
    <display-name>avi</display-name>  
    <welcome-file-list>  
        <welcome-file>index.html</welcome-file>  
        <welcome-file>index.htm</welcome-file>  
        <welcome-file>index.jsp</welcome-file>  
        <welcome-file>default.html</welcome-file>  
        <welcome-file>default.htm</welcome-file>  
        <welcome-file>default.jsp</welcome-file>  
    </welcome-file-list>  
    <servlet>  
        <servlet-name>Calculator</servlet-name>  
        <servlet-class>Calculator</servlet-class>  
    </servlet>  
    <servlet-mapping>  
        <servlet-name>Calculator</servlet-name>  
        <url-pattern>/Calculator</url-pattern>  
    </servlet-mapping>  
    <welcome-file-list>  
        <welcome-file>index.html</welcome-file>  
    </welcome-file-list>  
</web-app>
```

### Index.html

```
<!DOCTYPE html>  
<html>  
    <head>  
        <meta charset="ISO-8859-1">  
        <title>Insert title here</title>  
    </head>  
    <body>  
        <form action="Calculator" method="get" name="frm">
```

Enter num1:

```
<input name="txt1" type="text" /><br><br>
```

Enter num2:

```
<input name="txt2" type="text" /><br><br>
```

Operator

```
<select name="op">
```

```
    <option value="Addition">Addition</option>
    <option value="Subtraction">Subtraction</option>
    <option value="multiplication">multiplication</option>
    <option value="division">division</option>
</select>
```

```
<input type="submit" value="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "avi". It includes a "src" folder containing Java files like AddServlet.java, Calculator.java, Date1.java, Scontext.java, and Scontext.java. It also includes "Libraries", "build", "WebContent" (with "META-INF" and "WEB-INF" subfolders), and HTML files like index.html, final.jsp, first.jsp, and second.jsp.
- Code Editor:** Displays the content of the "Calculator.java" file. The code defines a class "Calculator" that extends "HttpServlet". It overrides the "processRequest" and "doGet" methods. The "processRequest" method reads parameters "txt1" and "txt2" from the request, sets the response content type to "text/html; charset=UTF-8", and prints the result based on the operator "op". The "doGet" method is similar but uses "PrintWriter" instead of "response.getWriter()".
- Bottom Status Bar:** Shows the current time as 27:23:1169, system status icons, and a search bar.

eclipse-workspace - avi/WebContent/WEB-INF/web.xml - Eclipse IDE

```
<?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" display-name="avi">
<!-- welcome-file-list -->
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<servlet>
<!-- servlet-name:Calculator /servlet-name -->
<servlet-name>Calculator</servlet-name>
<servlet-class>calculator</servlet-class>
</servlet>
<servlet-mapping>
<!-- servlet-name:Calculator /url-pattern -->
<servlet-name>Calculator</servlet-name>
<url-pattern>/Calculator</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>
```

Project Explorer    web.xml    index.html    Insert title here

Calculator.java    web.xml    index.html

File Edit Source Navigate Search Project Run Window Help

src  
Java Resources  
Libraries  
Referenced Libraries  
build  
WebContent  
META-INF  
WEB-INF  
lib  
web.xml  
final.jsp  
first.jsp  
index.html  
second.jsp

RemoteSystemsTempFiles  
Sample Aviral Project  
Servers

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

Tomcat v7.0 Server at localhost [Started, Synchronized]

Design Source

Writable Smart Insert 13 : 35 : 683

Type here to search ENG US 12:18 PM 12/4/2020

eclipse-workspace - avi/WebContent/index.html - Eclipse IDE

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="Calculator" method="get" name="frm">
<input type="text" value="Enter num1:<br><br>Enter num2:<br><br>Operator:<br><select name="op"><option value="Addition">Addition</option><option value="Subtraction">Subtraction</option><option value="multiplication">multiplication</option><option value="division">division</option></select>">
</form>

```

Project Explorer    web.xml    index.html    Insert title here

Calculator.java    web.xml    index.html

File Edit Source Navigate Search Project Run Window Help

src  
Java Resources  
Libraries  
Referenced Libraries  
build  
WebContent  
META-INF  
WEB-INF  
lib  
web.xml  
final.jsp  
first.jsp  
index.html  
second.jsp

RemoteSystemsTempFiles  
Sample Aviral Project  
Servers

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Error Log

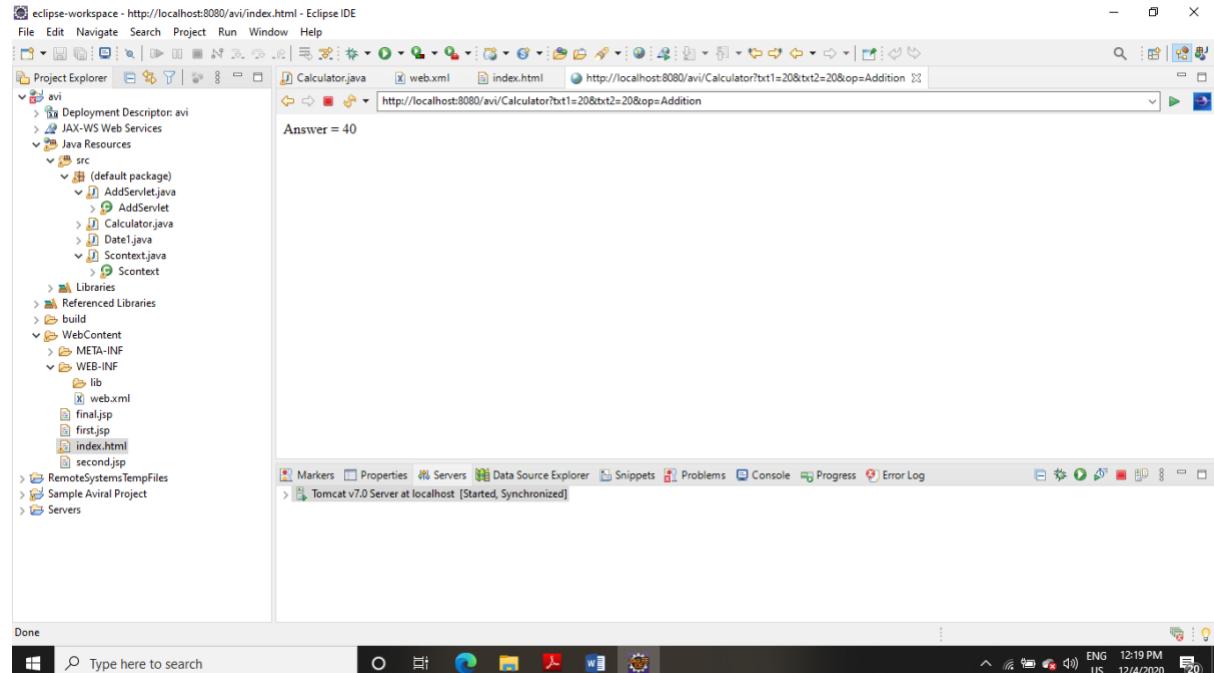
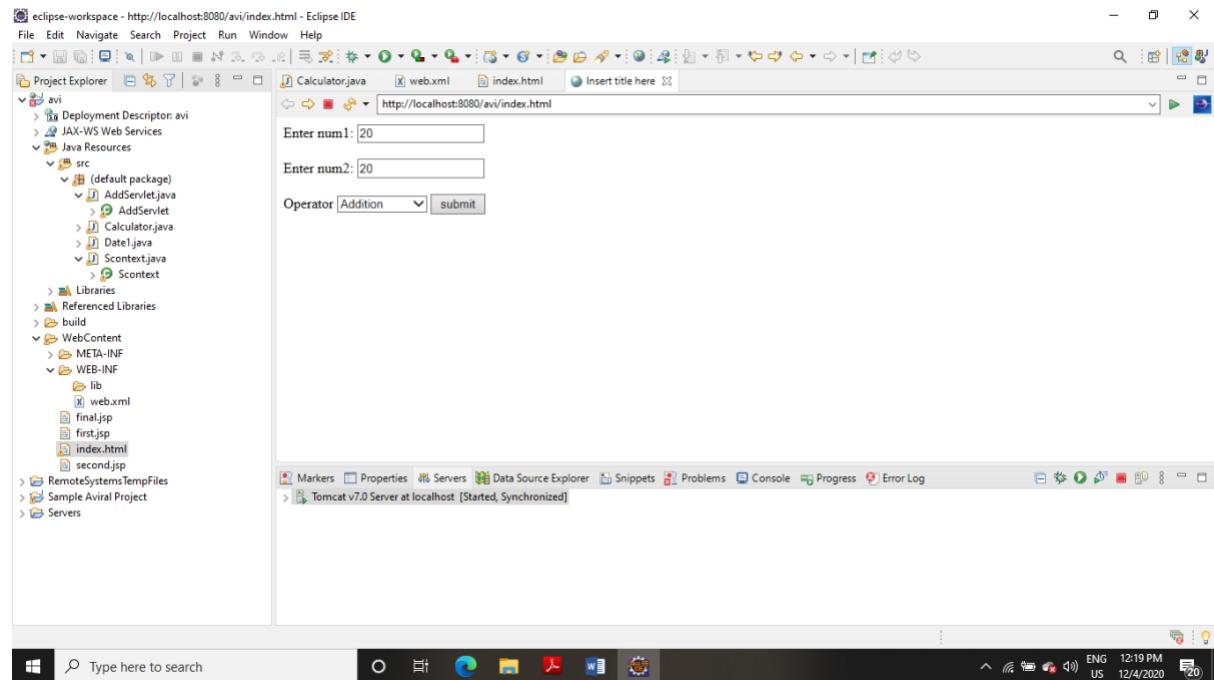
Tomcat v7.0 Server at localhost [Started, Synchronized]

Writable Smart Insert 36 : 1 : 929

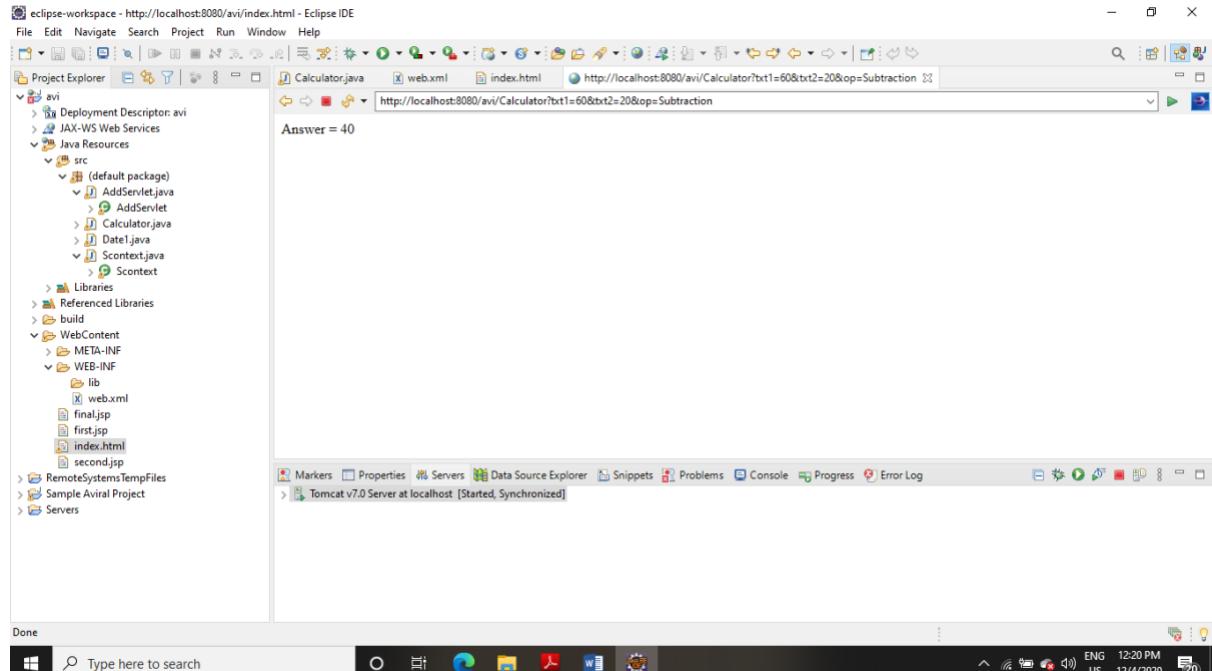
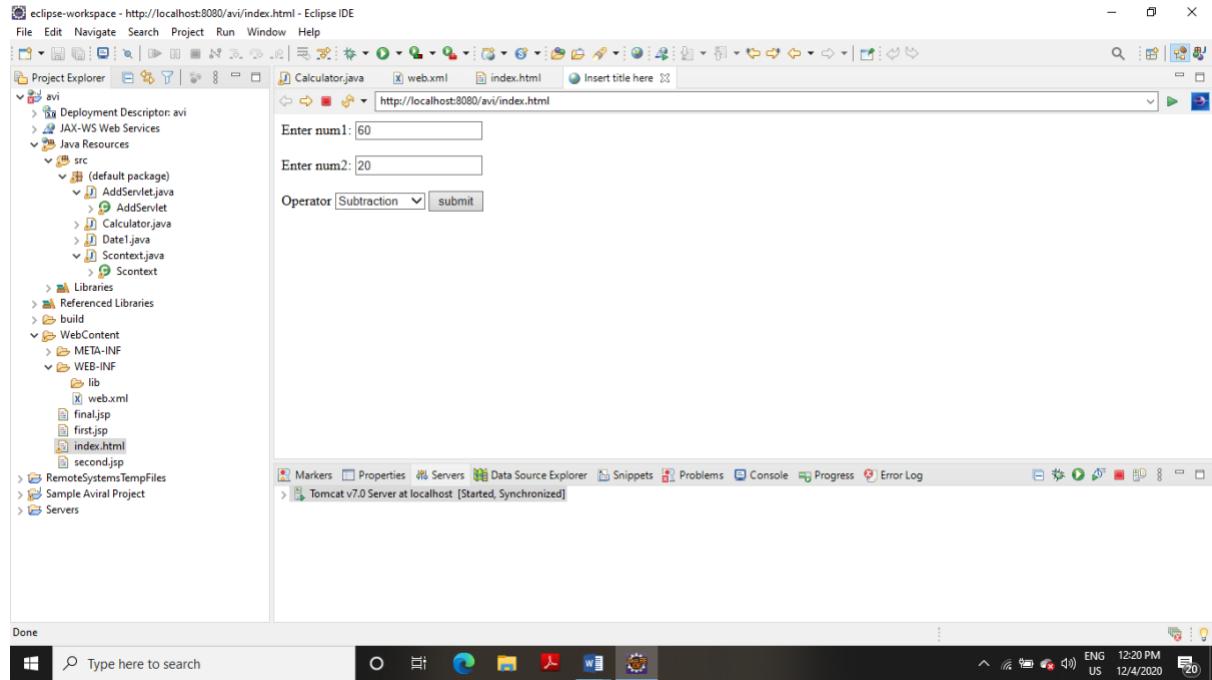
Type here to search ENG US 12:18 PM 12/4/2020

## OUTPUT:

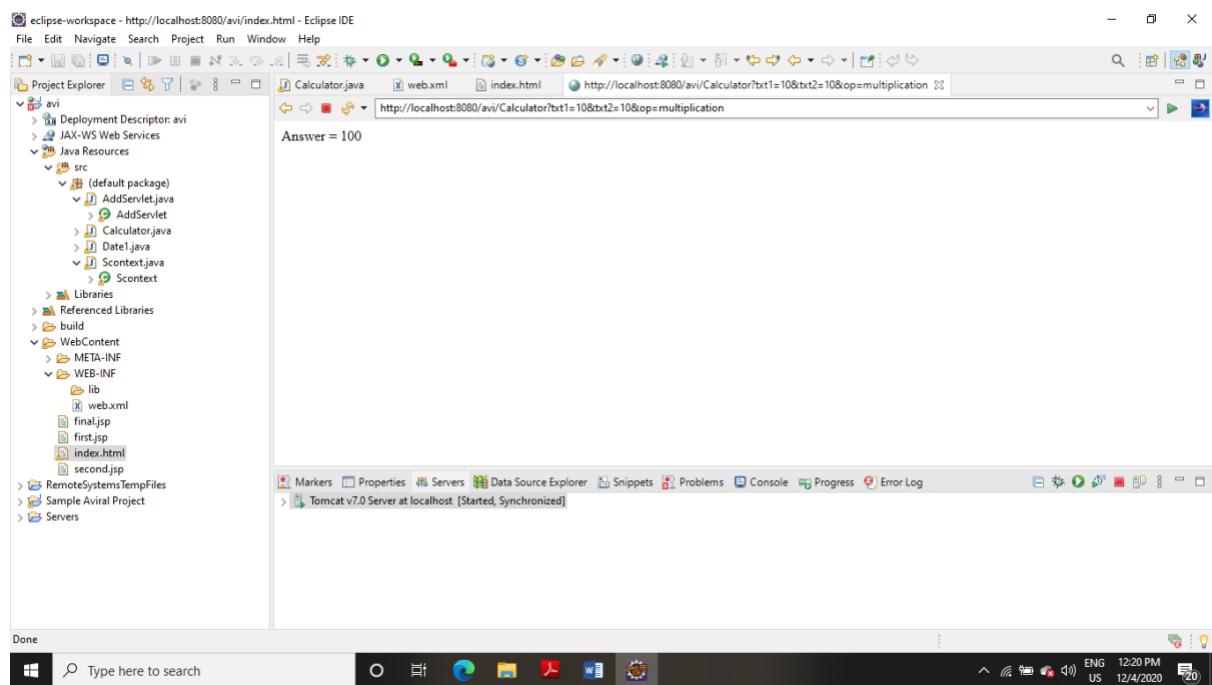
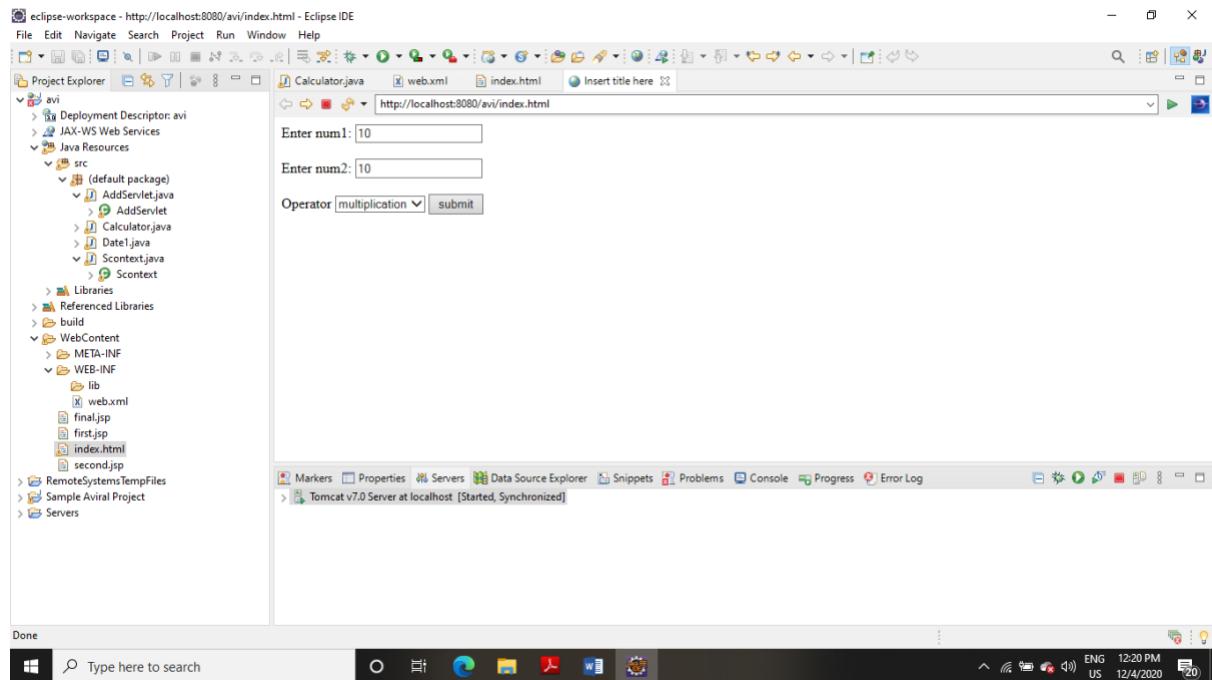
### (a) Addition



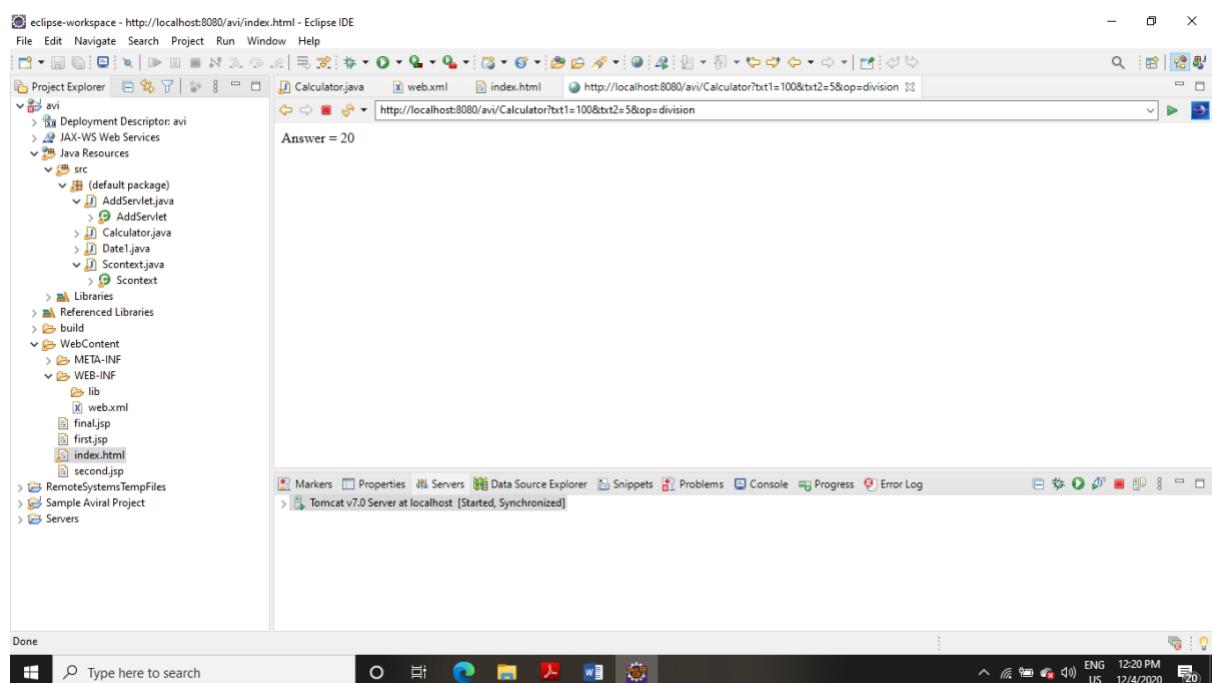
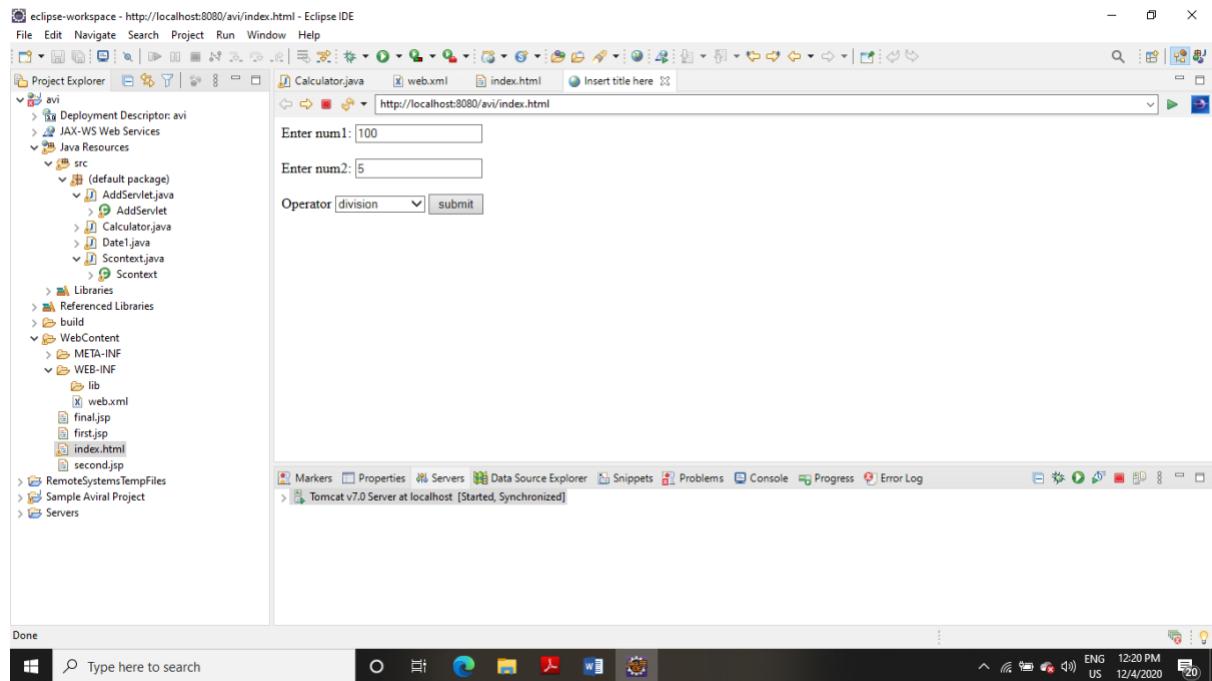
## (b) Subtraction



### (c) Multiplication



## (d) Division



## **EXPERIMENT NO-12**

### **TITLE:** JSP

**1. Write a JSP page to access the data of a student from the student table.**

### **CODE**

```
<%@ page import = "java.io.* , java.util.* , java.sql.*" %>
<%@ page import = "javax.servlet.http.* , javax.servlet.*" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix = "sql" %>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%
String id = request.getParameter("userid");
String driver = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://127.0.0.1:3306/student";
String database = "student";
String userid = "root";
String password = "aviral08";
try
{
Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException e)
{
e.printStackTrace();
}
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<!DOCTYPE html>
<html>
<body>

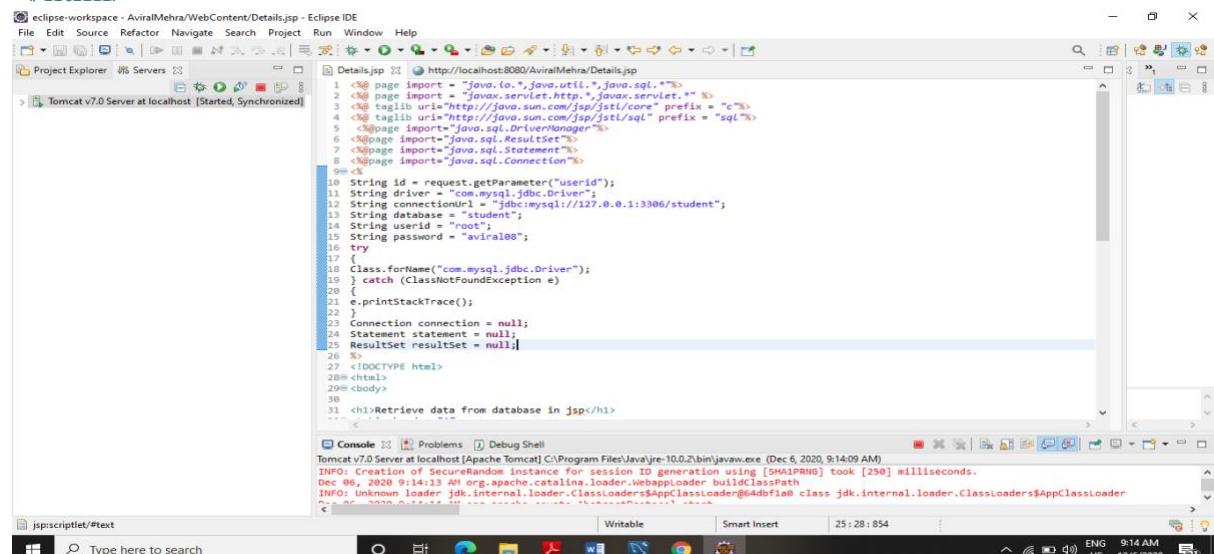
<h1>Retrieve data from database in jsp</h1>
<table border="1">
<tr>
<td>ID</td>
<td>AGE</td>
<td>First Name</td>
```

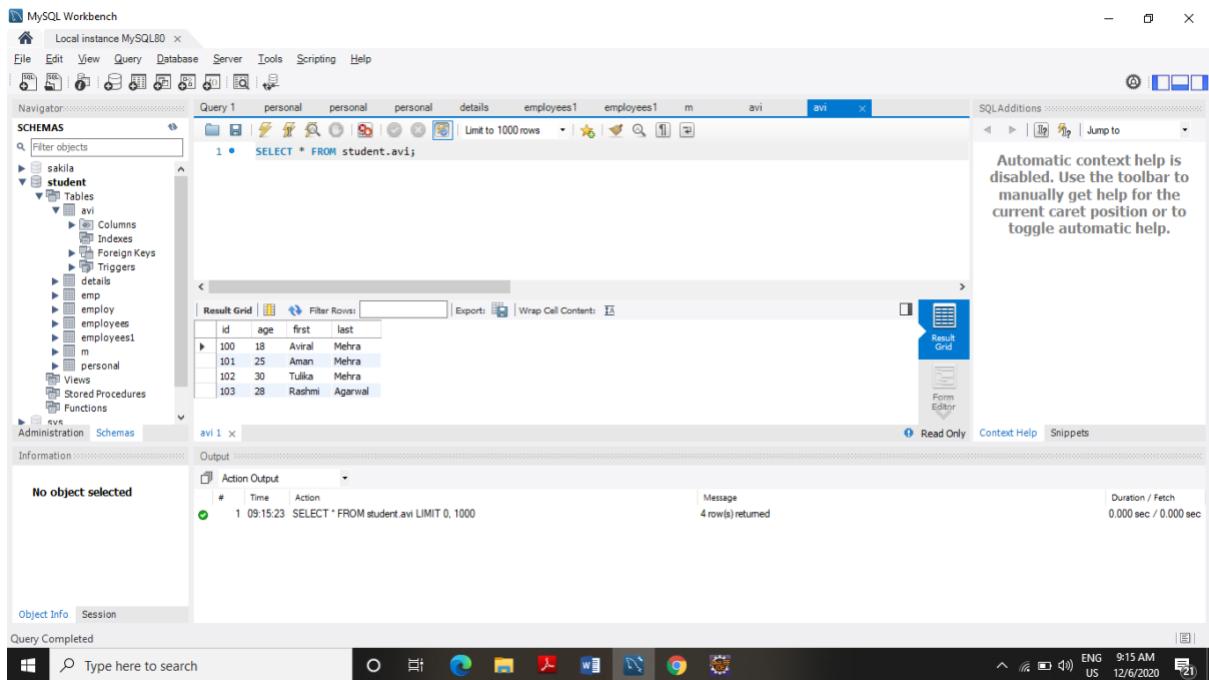
```

<td>Last Name</td>

</tr>
<%
try
{
connection = DriverManager.getConnection(connectionUrl,userid,password);
statement=connection.createStatement();
String sql ="SELECT * FROM avi";
resultSet = statement.executeQuery(sql);
while(resultSet.next())
{
%
<tr>
<td><%=resultSet.getString("id") %></td>
<td><%=resultSet.getString("age") %></td>
<td><%=resultSet.getString("first") %></td>
<td><%=resultSet.getString("last") %></td>
</tr>
<%
}
connection.close();
} catch (Exception e)
{
e.printStackTrace();
}
%
</table>
</body>
</html>

```





## OUTPUT

The screenshot shows the Eclipse IDE interface with a project named 'eclipse-workspace'. A browser window in the center displays the URL <http://localhost:8080/AviralMehra/Details.jsp>. The page content is titled 'Retrieve data from database in jsp' and contains the following table:

ID	AGE	First Name	Last Name
100	18	Aviral	Mehra
101	25	Aman	Mehra
102	30	Tulika	Mehra
103	28	Rashmi	Agarwal

In the bottom right corner of the browser window, there is a status bar showing 'ENG US 9:15 AM 12/6/2020'.

**2. Write a JSP Login page to enter the username and password entered by user and display the welcome page on successful login otherwise display wrong authentication page.**

## CODE

### Register\_1.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```

pageEncoding="ISO-8859-1">%>
<!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>Registration Form</title>
</head>
<body>
<h1>Registration Form</h1>
<form action="register" method="post">
    <table style="width: 50%">
        <tr>
            <td>First Name</td>
            <td><input type="text" name="first_name" /></td>
        </tr>
        <tr>
            <td>Last Name</td>
            <td><input type="text" name="last_name" /></td>
        </tr>
        <tr>
            <td>UserName</td>
            <td><input type="text" name="username" /></td>
        </tr>
        <tr>
            <td>Password</td>
            <td><input type="password" name="password" /></td>
        </tr>
        <tr>
            <td>Address</td>
            <td><input type="text" name="address" /></td>
        </tr>
        <tr>
            <td>Contact No</td>
            <td><input type="text" name="contact" /></td>
        </tr>
    </table>
    <input type="submit" value="Submit" /></form>

```

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

## Register.java

```
import java.io.IOException;
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 register
 */
public class register extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        String first_name = request.getParameter("first_name");
        String last_name = request.getParameter("last_name");
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String address = request.getParameter("address");
        String contact = request.getParameter("contact");

        if(first_name.isEmpty() || last_name.isEmpty() ||
username.isEmpty() ||
                password.isEmpty() || address.isEmpty() ||
contact.isEmpty())
        {
            RequestDispatcher req =
request.getRequestDispatcher("Register_1.jsp");
            req.include(request, response);
        }
        else
        {
            RequestDispatcher req =
request.getRequestDispatcher("Register_2.jsp");
        }
    }
}
```

```
        req.forward(request, response);
    }

}
```

## Register\_2.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!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>Success Page</title>
</head>
<body>
    <a><b>WELCOME AVIRAL MEHRA!!!!</b></a>
</body>
</html>
```

## Registration\_3.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!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>LOGIN PAGE</title>
</head>
<body>
<form action="Login" method="post">
    <table style="width: 50%">

        <tr>
            <td>UserName</td>
            <td><input type="text" name="username" /></td>
        </tr>
        <tr>
            <td>Password</td>
            <td><input type="password" name="password" /></td>
        </tr>
    </table>
</form>

```

```

</table>
<input type="submit" value="Login" /></form>
</body>
</html>
Login.java
import java.io.IOException;
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 Login
 */
public class Login extends HttpServlet {

    public Login() {
        super();
        // TODO Auto-generated constructor stub
    }

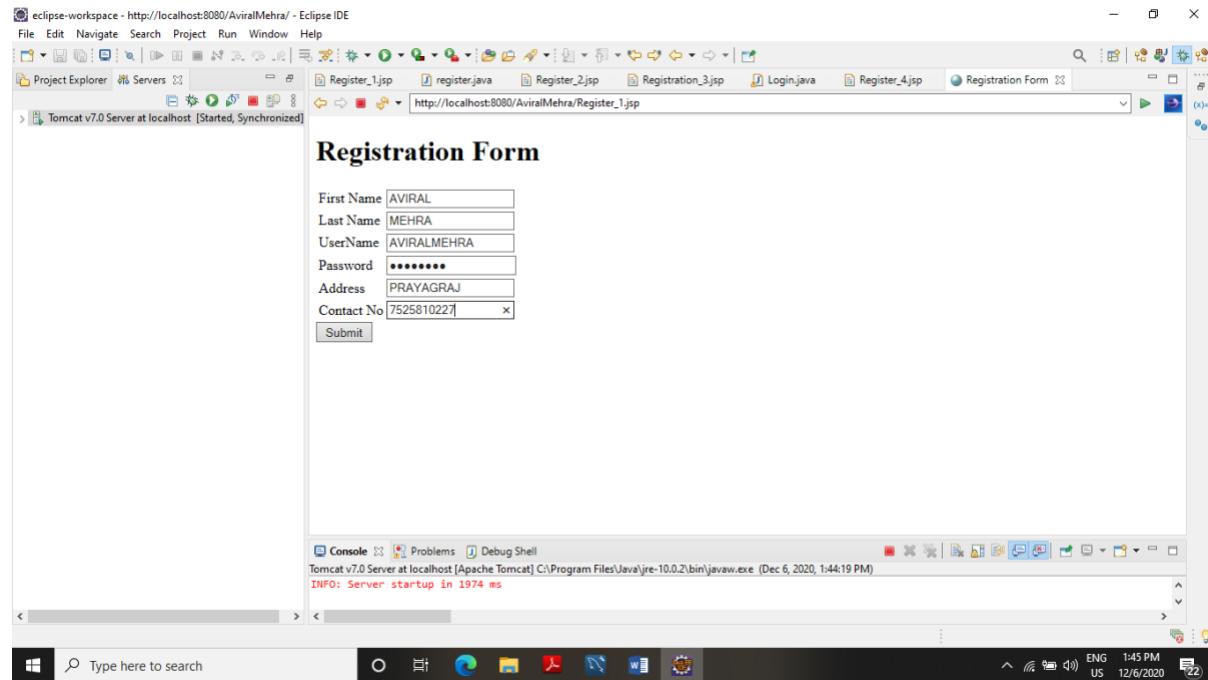
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        // TODO Auto-generated method stub
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        if(username.isEmpty() || password.isEmpty() )
        {
            RequestDispatcher req =
request.getRequestDispatcher("Registration_3.jsp");
            req.include(request, response);
        }
        else
        {
            RequestDispatcher req =
request.getRequestDispatcher("Register_4.jsp");
            req.forward(request, response);
        }
    }
}

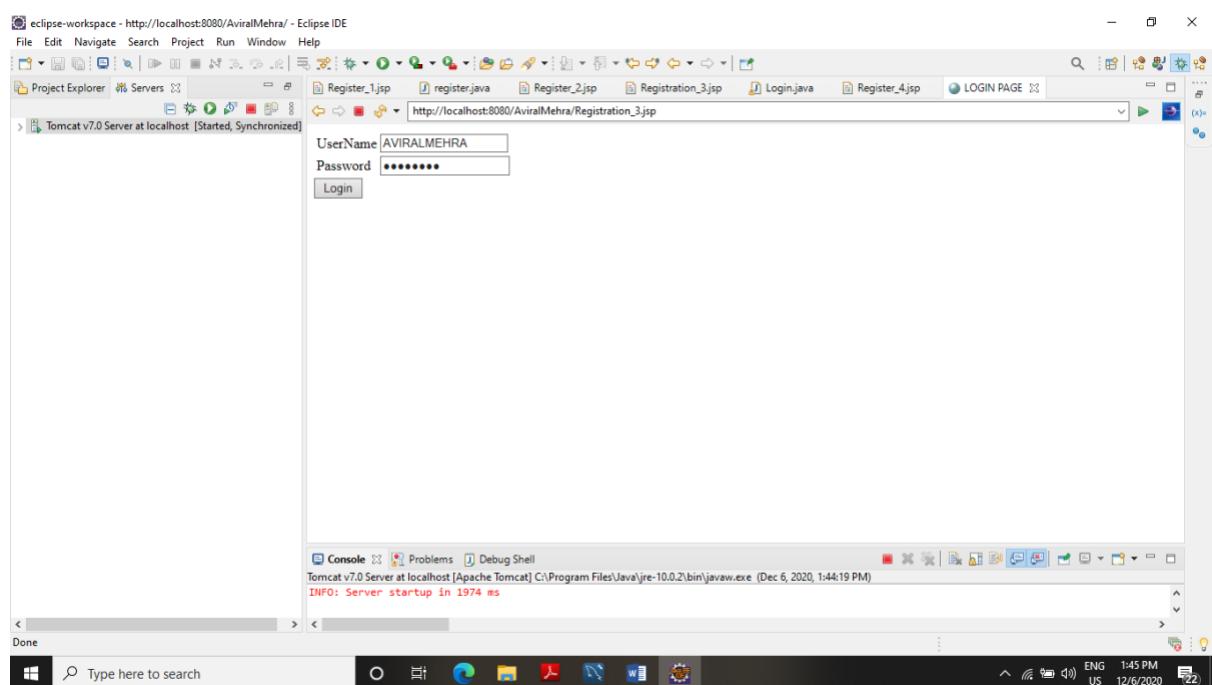
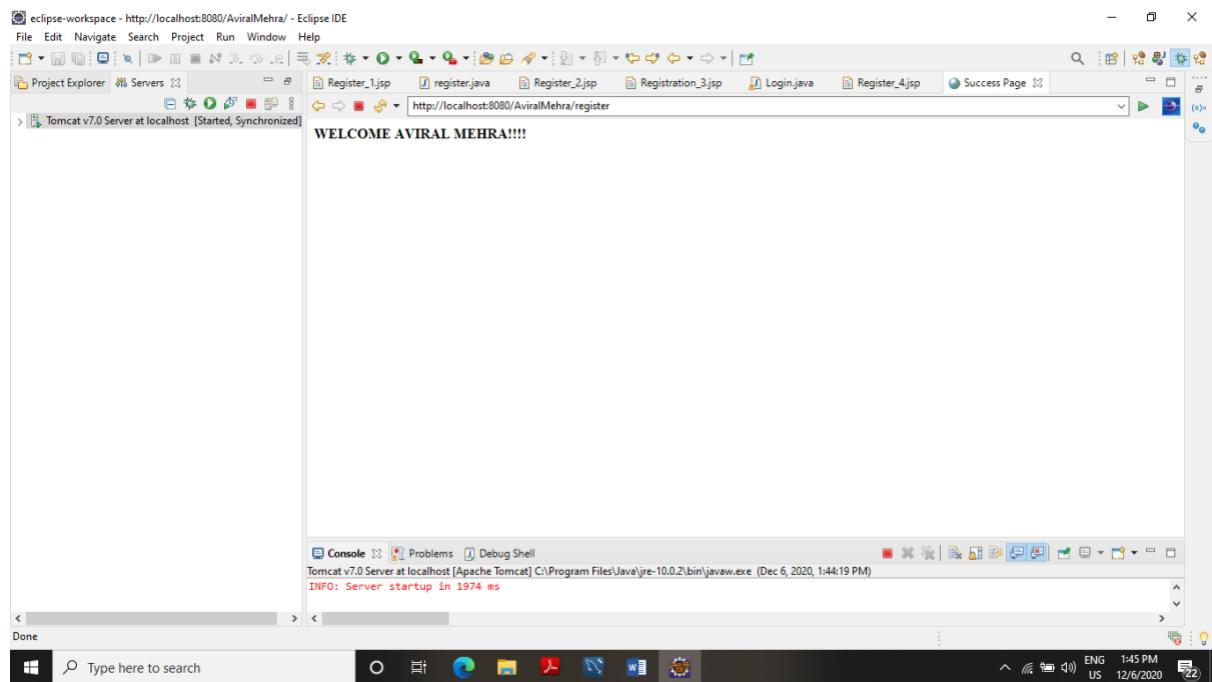
```

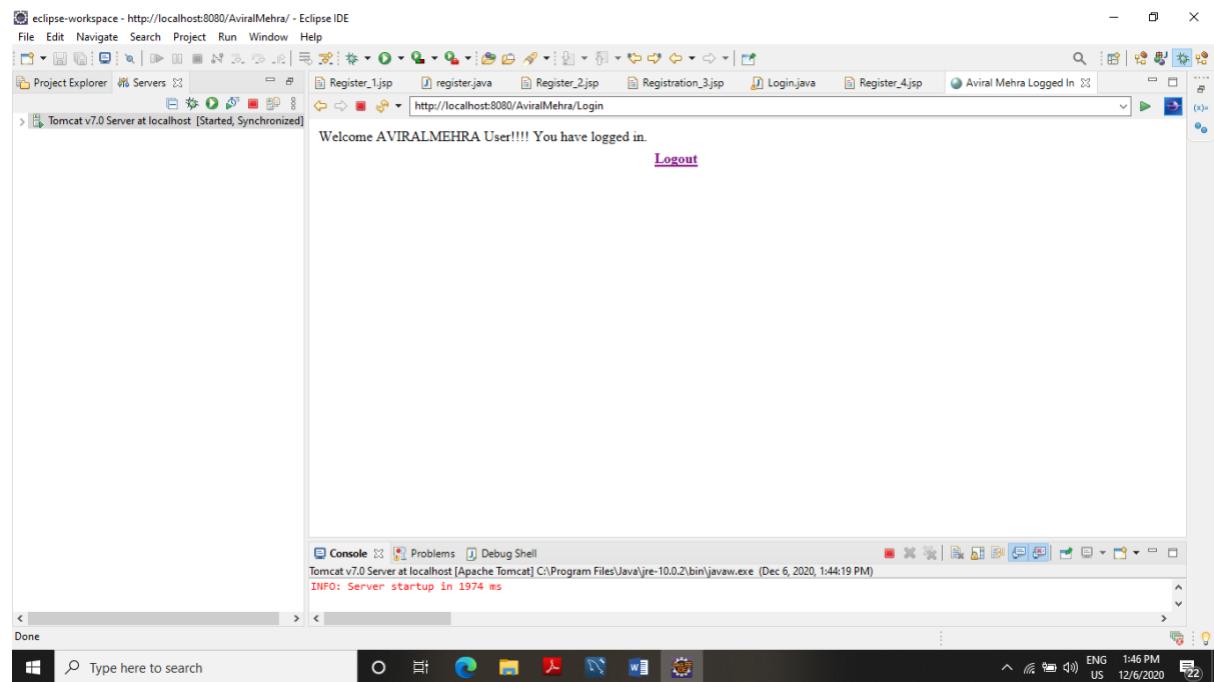
## Register\_4.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!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>Aviral Mehra Logged In</title>
</head>
<body>
    <table style="width: 50%">
        <tr><td>
            <% String username = request.getParameter("username"); %>
<a>Welcome <% out.println(username); %> User!!!! You have logged
in.</a></td></tr>
        <tr><td></td><td></td><td></td><td></td><td><a
href="Registration_3.jsp"><b>Logout</b></a></td></tr>
    </table>
</body>
</html>
```

## OUTPUT







**END**