

***LAB PRACTICAL FILE***

**NAME: ABHISHEK SHARMA**

**CLASS: CSE DEVOPS 4th SEM B-1**

**ROLL NO.: R171218005**

**SAP ID: 500067644**

**Subject: Object Oriented Programming Language (Java)**

**EXPERIMENT – 2,3**

**TITLE:** Basic Java Programming

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

import java.util.\*;

public class LargestofthreeNumbers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a,b,c;

a = sc.nextInt();

b = sc.nextInt();

c = sc.nextInt();

if(a>b && a>c)

{

System.out.print("Largest no" + a);

}

else if(b>c && a>c)

{

System.out.print("Largest no" + b);

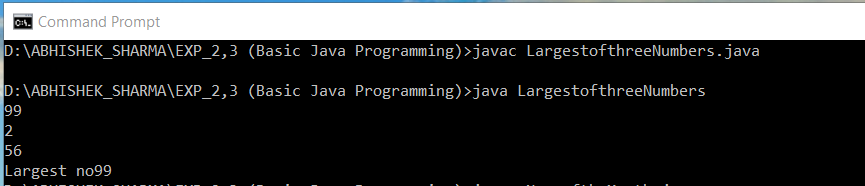
}

else

{

System.out.print("Largest no" + c);

}}}



1. **Write a program to add two number using command line arguments.**

import java.util.\*;

public class add two number {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a,b;

System.out.println("Enter first number");

a = sc.nextInt();

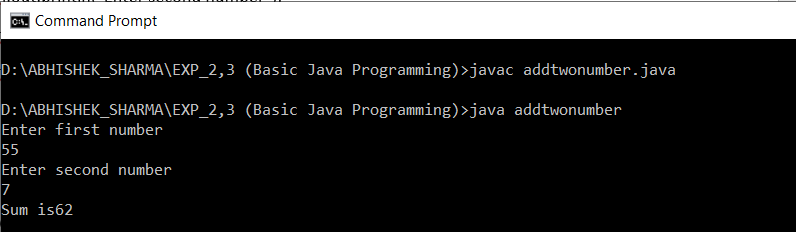
System.out.println("Enter second number");

b = sc.nextInt();

System.out.println("Sum is" + (a+b));

}

}



1. **Write a program to print Fibonacci series using loop.**

public class Fibonacci series

{

public static void main(String[] args)

{

int n1=0,n2=1,n3,i,count=10;

System.out.print(n1+" "+n2);//printing 0 and 1

for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed

{

n3=n1+n2;

System.out.print(" "+n3);

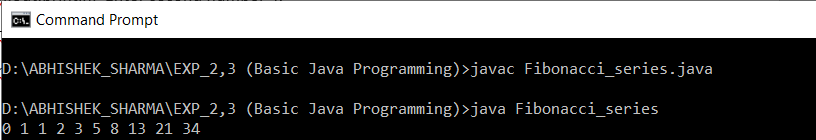
n1=n2;

n2=n3;

}

}

}



1. **Write a program to implement a command line calculator.**

import java.util.\*;

public class calculator

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int a,b,c = 0;

System.out.println("Enter first number");

a = sc.nextInt();

System.out.println("Enter second number");

b = sc.nextInt();

System.out.print("Enter an operator (+, -, \*, /): ");

char operator = sc.next().charAt(0);

switch(operator)

{

case '+':

c = a + b;

break;

case '-':

c = a - b;

break;

case '\*':

c = a \* b;

break;

case '/':

c = a / b;

break;

default:

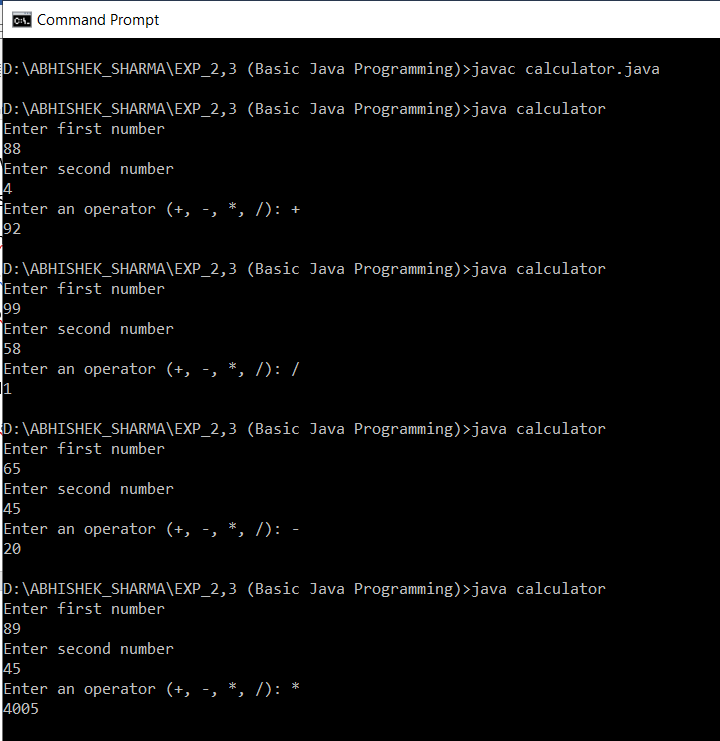
System.out.println("Wrong value");

}

System.out.println(c);

}

}



1. **Write a program using classes and object in java.**

public class Student {

int student\_age;

public Student(String name) {

System.out.println("Nameis :" + name );

}

public void setAge( int age ) {

student\_age = age;

}

public int getAge( ) {

System.out.println("Student's age is :" + student\_age );

return student\_age;

}

public static void main(String[] args) {

Student s = new Student( "ABHISHEK" );

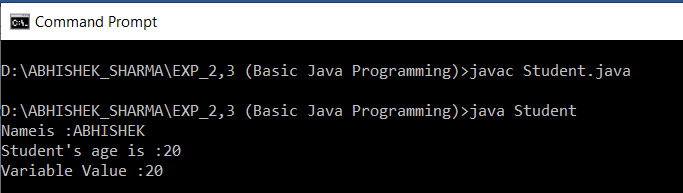
s.setAge( 20 );

s.getAge( );

System.out.println("Variable Value :" + s.student\_age );

}

}



**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**

import java.util.\*;

public class Student{

public static void main(String[] args) {

int n, total = 0, percentage;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of subject:");

n = s.nextInt();

int marks[] = new int[n];

System.out.println("Enter marks out of 100:");

for(int i = 0; i < n; i++)

{

marks[i] = s.nextInt();

total = total + marks[i];

}

percentage = total / n;

System.out.println("Sum:"+total);

System.out.println("Percentage:"+percentage);

if(40<percentage && percentage< 50)

{

System.out.println("PASS");

}

else if(51<percentage && percentage<75)

{

System.out.println("MERIT");

}

else

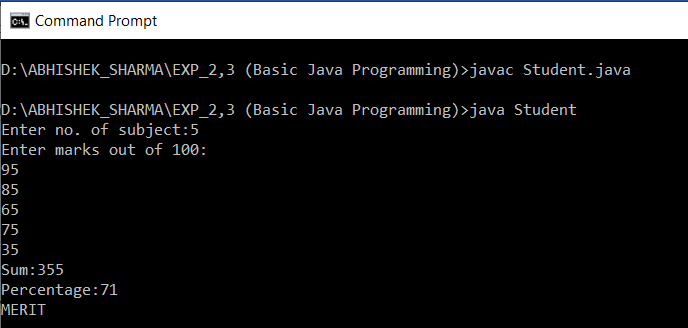
{

System.out.println("DESTINCTION");

}

}

}



**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.)**

public class Combinations

{

public static void main(String[] args)

{

int[] input = { 1, 2, 3 };

for (int x = 0; x < 3; x++) {

for (int y = 0; y < 3; y++) {

for (int z = 0; z < 3; z++) {

if (x != y && y != z && z != x) {

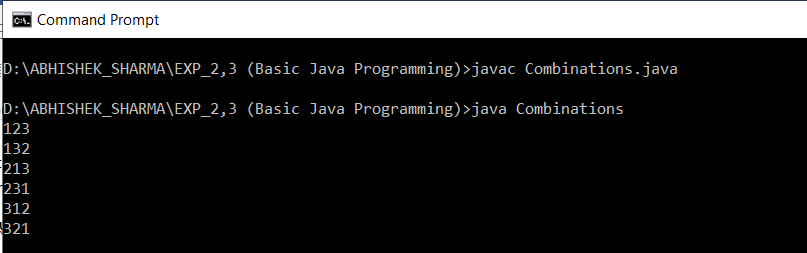
System.out.println(input[x] + "" + input[y] + ""

+ input[z]);

}

}

}



**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.**

import java.util.\*;

public class SquareofNumbers {

public static void main(String[] args) {

int n, sum = 0;

Scanner s = new Scanner(System.in);

n=10;

int a[] = new int[n];

System.out.println("Enter all the elements:");

for(int i = 0; i < n; i++)

{

a[i] = s.nextInt();

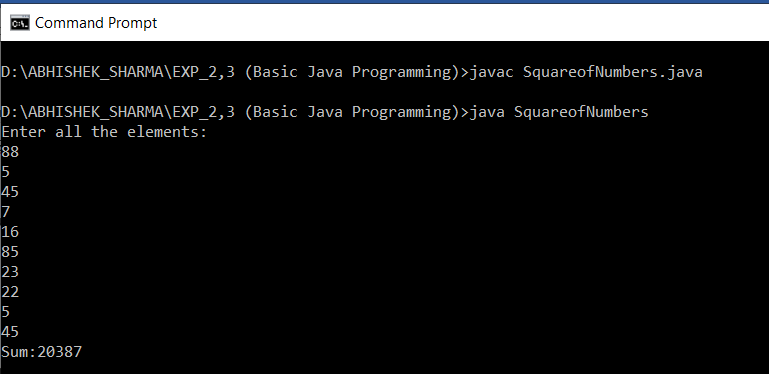
sum = sum + a[i]\*a[i];

}

System.out.println("Sum:"+sum);

}

}



**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.)**

import java.util.\*;

public class NameoftheMonth{

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Enter month's number: ");

int monthNumber;

monthNumber = in.nextInt();

switch (monthNumber) {

case 1:

System.out.println("January");

break;

case 2:

System.out.println("February");

break;

case 3:

System.out.println("March");

break;

case 4:

System.out.println("April");

break;

case 5:

System.out.println("May");

break;

case 6:

System.out.println("June");

break;

case 7:

System.out.println("July");

break;

case 8:

System.out.println("August");

break;

case 9:

System.out.println("September");

break;

case 10:

System.out.println("October");

break;

case 11:

System.out.println("November");

break;

case 12:

System.out.println("December");

break;

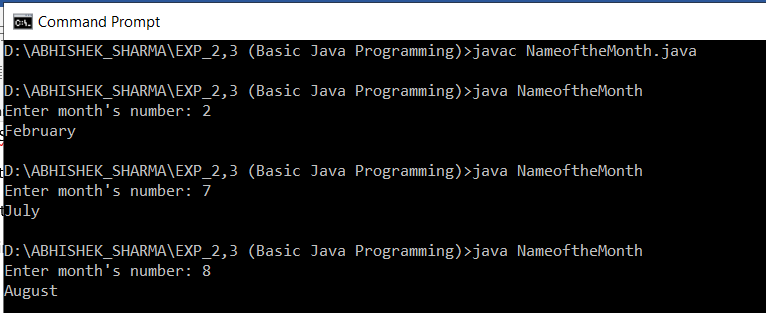
default:

System.out.println("Invalid month.");

break;

} }

}



**10. Write a program to find the sum of all integers greater than 40 and less than 250 that are divisible by 5.**

public class SumofIntegers

{

public static void main(String[] args)

{

int result=0;

for(int i =40;i<250;i++){

if(i%5==0)

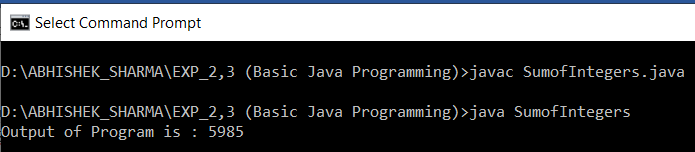
result+=i;

}

System.out.println("Output of Program is : "+result);

}

}



**EXPERIMENT – 4**

**TITLE: Inheritance**

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

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);}}

public class room {

public static void main(String[] args) {

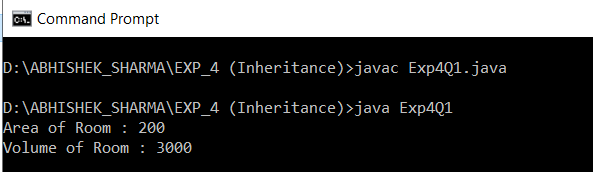
class\_room cr=new class\_room(10,20,15);

int a1=cr.area();

int v1=cr.volume();

System.out.println("Area of Room : "+a1);

System.out.println("Volume of Room : "+v1);} }



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

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 cricket\_player extends player{

String type;

cricket\_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);

}

}

public class Ex4Q2 {

public static void main(String[] args) {

cricket\_player c=new cricket\_player("Ameer","criket",25);

football\_player f=new football\_player("arun","foot ball",25);

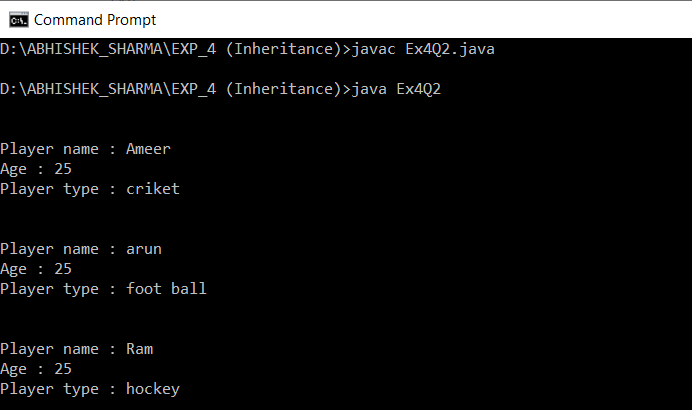
hockey\_player h=newhockey\_player("Ram","hockey",25);

c.show();

f.show();

h.show();

}}



**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.**

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);

}

}

public class PROG3{

public static void main(String[] args) {

dailyworker d=new dailyworker(254,"ABHISHEK",75);

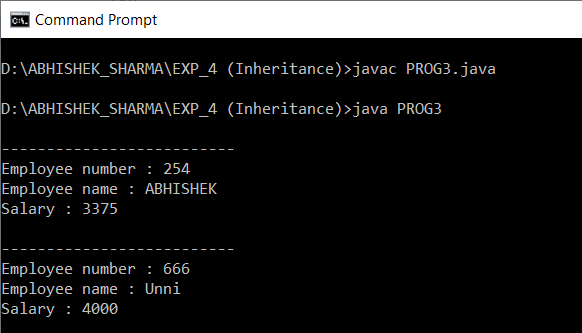
salariedworker s=new salariedworker(666,"Unni",100);

d.compay(45);

s.compay();

}

}



**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.**

import java.io.\*;

class call{

float dur;

String type;

float rate(){

if(type.equals("urgent"))

return 4.5f;

else if(type=="lightning")

return 3.5f;

else

return 3f;

}

}

class bill extends call{

float amount;

DataInputStream in=null;

bill(){

try

{

in=new DataInputStream(System.in);

}

catch(Exception e)

{ System.out.println(e); }

}

void read()throws Exception

{

String s;

System.out.println("enter call type(urgent,lightning,ordinary):");

type=in.readLine();

System.out.println("enter call duration:");

s=in.readLine();

dur=Float.valueOf(s).floatValue();

}

void calculate(){

if(dur<=1.5)

amount=rate()\*dur+1.5f;

else if(dur<=3)

amount=rate()\*dur+2.5f;

else if(dur<=5)

amount=rate()\*dur+4.5f;

else

amount=rate()\*dur+5f;

}

void print(){

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" PHONE BILL ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" Call type : "+type);

System.out.println(" Duration : "+dur);

System.out.println(" CHARGE : "+amount);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

public class Prog4{

public static void main(String[] args) throws Exception {

bill b=new bill();

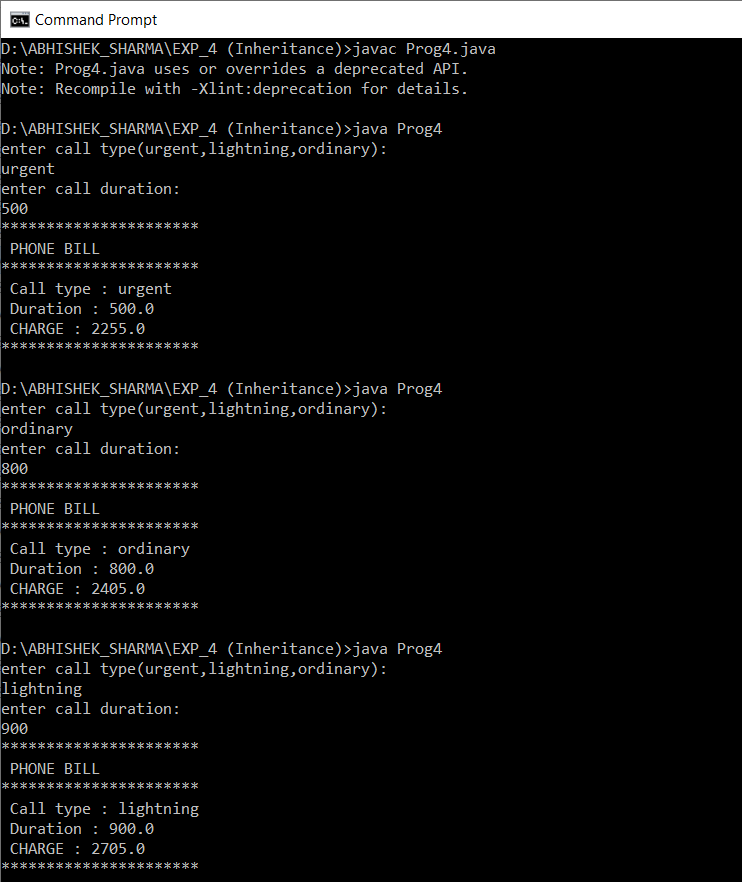
b.read();

b.calculate();

b.print();

}

}



**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 theses classes and methods.**

class Manager extends employee

{

public int depart\_id;

}

// Driver class

public class employee

{

public String name;

public int empid,salary;

employee()

{

System.out.println("object created, set the values");

}

employee(int eid,String name,int sal)

{

this.name=name;

empid=eid;

salary=sal;

}

public String getname()

{

return name;

}

public int getsalary()

{

return salary;

}

public void increasesalary(int per)

{

salary+=(salary\*per)/100;

}

public static void main(String[] args)

{

employee g1 = new employee(67759,"ABHISHEK",500000);

System.out.println("Employee Salary:"+g1.getsalary());

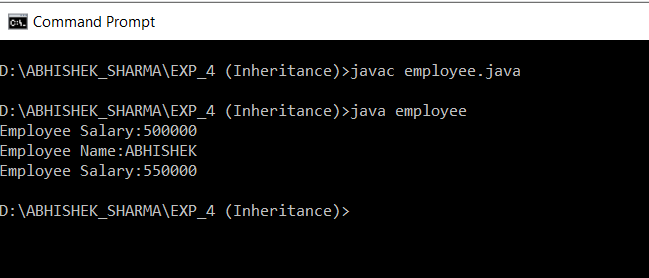
System.out.println("Employee Name:"+g1.getname());

g1.increasesalary(10);

System.out.println("Employee Salary:"+g1.getsalary());

}

}



**EXPERIMENT – 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.**

interface Test

{

int square(int a);

}

class arithmetic implements Test

{

int num;

public int square(int a){ return a\*a;}

int add(int a){ return a+a;}

}

public class ToTestInt

{

public static void main(String args[]){

arithmetic a1= new arithmetic();

System.out.println(a1.square(2));

}

}



**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.**

interface A

{

void meth1();

void meth2();

}

public class MyClass implements A

{

public void meth1(){System.out.println("Method1");}

public void meth2(){System.out.println("Method2");}

public static void main(String args[])

{

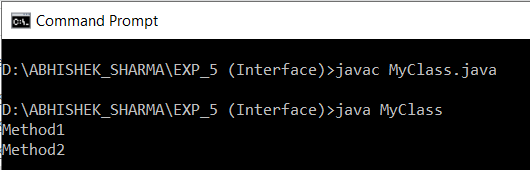
MyClass obj = new MyClass();

obj.meth1();

obj.meth2();

}

}



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

interface area

{

final double pi=3.14;

double getarea(double r);

}

class circle implements area

{

public double getarea(double r){

double area=2\*pi\*r;

return area;}

}

public class AreaCalc

{

public static void main(String args[])

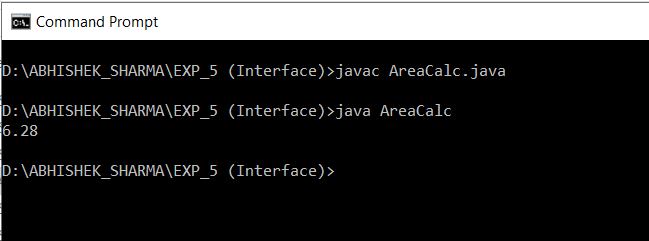
{

circle c1= new circle();

System.out.println(c1.getarea(1));

}

}



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

interface cal

{

int division(int num);

int modules(int num);

}

public class Calculator implements cal

{

public int division(int num){

return num/2;

}

public int modules(int num){

return num%2;

}

public static void main(String args[])

{

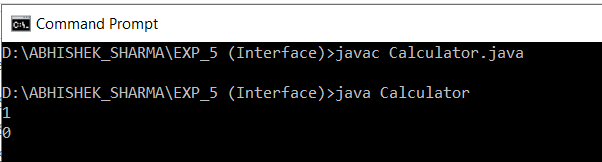
Calculator c1= new Calculator ();

System.out.println(c1.division(2));

System.out.println(c1.modules(2));

}

}



**EXPERIMENT – 6**

**TITLE: Package**

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

**File Name: pac.java**

package exp6;

import java.util.Scanner;

public class pac {

public int num1, num2;

protected String member;

public pac() // defining constructor

{

System.out.println("Enter value of num1,num2");

Scanner in = new Scanner(System.in); /\* making object for scanner class\*/

num1 = in.nextInt();

num2 = in.nextInt();

System.out.println(num1 + num2);

}

}

**File Name: main.java**

package exp601;

import exp6.pac;

public class main

{

public static void main(String[] args)

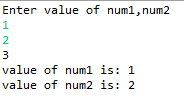
{

pac obj = new pac();

System.out.println("value of num1 is: " + obj.num1);

System.out.println("value of num2 is: " + obj.num2);

}}



**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.**

File Name: pac.java

package exp6;

import java.util.Scanner;

class account {

public String name;

public int balance;

account() {

Scanner in = new Scanner(System.in);

System.out.println("Enter you name :");

name = in.nextLine();

System.out.println("Enter your balance");

balance = in.nextInt();

System.out.println();

System.out.println("dear :" + name + " your balance is :" + balance);

}

}

public class pac {

account obj = new account();

}

File Name: main.java

package exm6;

import exp6.pac;

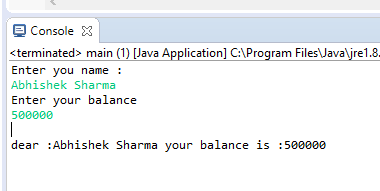
public class main {

public static void main(String[] args) {

pac obj = new pac();

}

}



**EXPERIMENT – 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 ArrayIndexOutOfBoundsExeption, so that any such problem doesn’t cause illegal termination of program.**

class Student{

public String Name;

public int Roll\_No;

public Student(){ };

public Student(String N, int R)

{

Name = N;

Roll\_No = R;

System.out.println("Name : "+Name+" "+"Roll\_no : "+Roll\_No);

}

public void print(String P, int K)

{

Name = P;

Roll\_No = K;

System.out.println("Name : "+Name+" "+"Roll\_no : "+Roll\_No);

}

}

class Exception1

{

public static void main(String[] args)

{

Student[] Stu;

Stu = new Student[10];

Stu[0] = new Student("ABHISHEK",91);

Stu[1] = new Student("SHARMA",92);

Stu[2] = new Student("AAKAR",93);

Stu[3] = new Student("kd",94);

Stu[4] = new Student("ANI",95);

Stu[5] = new Student("rathore",96);

Stu[6] = new Student("loki",97);

Stu[7] = new Student("piyush",98);

Stu[8] = new Student("duggal",99);

Stu[9] = new Student("arjun",100);

try

{

Stu[10] = new Student();

Stu[10].print("sidhu",111);

}

catch(ArrayIndexOutOfBoundsException e)

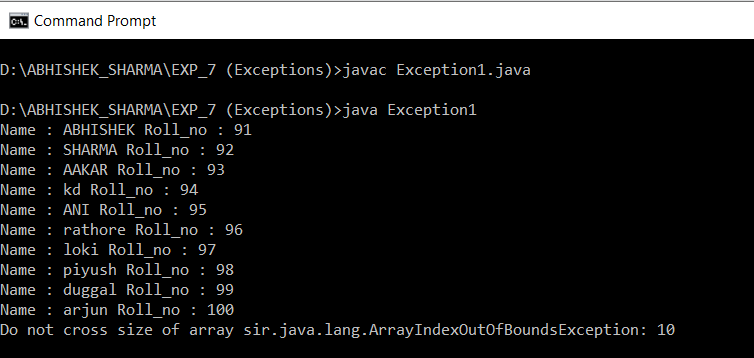
{

System.out.println("Do not cross size of array sir." + e );

}

}

}



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

import java.util.Scanner;

class demoexception

{

int num1, num2;

void dividefunction()

{

System.out.println("Enter two numbers: ");

Scanner in = new Scanner(System.in);

num1 = in.nextInt();

num2 = in.nextInt();

try { // here it will throw exception

int result = num1 / num2;

System.out.println(result);

} catch (ArithmeticException e)

{

// here we get answers to our exception

System.out.println("A number can not be divided by 0");

}

}

}

public class Exception2 {

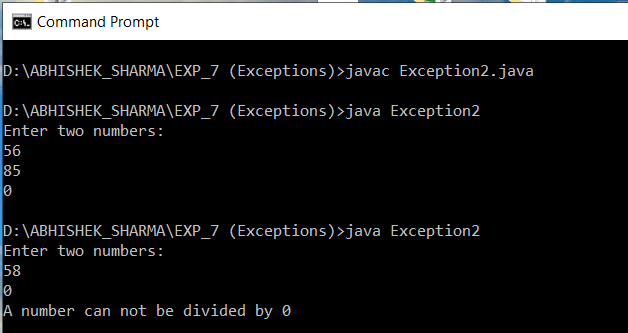
public static void main(String[] args) {

demoexception obj = new demoexception();

obj.dividefunction();

}

}



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

import java.io.\*;

import java.util.\*;

class Test3

{

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)

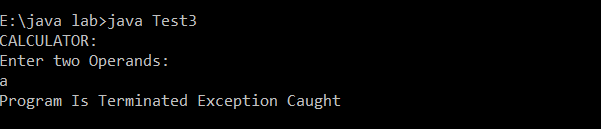
{

System.out.println("Program Is Terminated Exception Caught");

}

}

}



**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.**

import java.util.\*;

class direction extends Exception{

direction(){

super("collision");}}

public class collision{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("enter the direction same or opposite");

String a=sc.nextLine();

try{

if (a.equals("opposite")){

throw new direction();}

else{

System.out.println("no problem");

}}

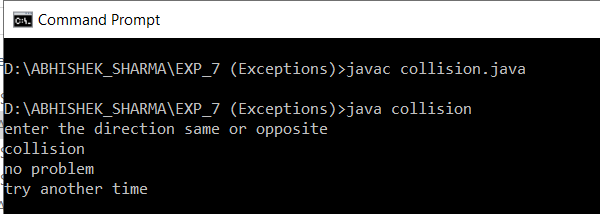
catch(direction e){

System.out.println(e);

}

System.out.println("try another time");

}}



**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**

import java.io.\*;

import java.util.\*;

class Exception5

{

public static void main(String args[])

{

String name;

int age;

System.out.println("-----ENTER EMPLOYEE DETAILS-----");

System.out.println("Enter Name and Age:");

Scanner in=new Scanner(System.in);

try

{

if(!(in.nextLine().matches("[a-zA-Z]+")))

{throw new IOException();}

age=in.nextInt();

if(age>50)

{

System.out.println("Age greater than 50 Exception");

throw new Exception();

}

Exception5 x=new Exception5();

System.out.println("-----Object Created-----");

}

catch(Exception e)

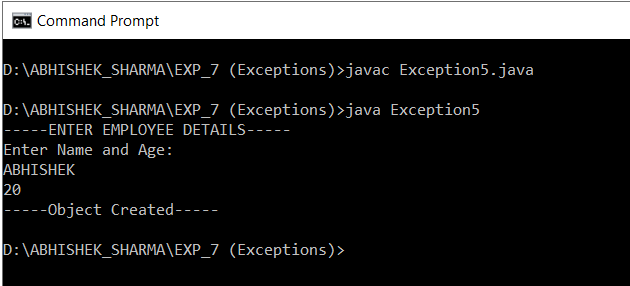
{

System.out.println("Exception");

}

}

}



**EXPERIMENT – 8**

**TITLE: Strings Handling and Wrapper Class**

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.**

class Occurence

{

public static void main (String[] args)

{

String str = "Hey! This is ABHISHEK SHARMA.";

int firstIndex = str.indexOf('s');

System.out.println("First occurrence of char 's'" + " is found at : " + firstIndex);

int lastIndex = str.lastIndexOf('s');

System.out.println("Last occurrence of char 's' is" + " found at : " + lastIndex);

int first\_in = str.indexOf('s', 10);

System.out.println("First occurrence of char 's'" + " after index 10 : " + first\_in);

int last\_in = str.lastIndexOf('s', 20);

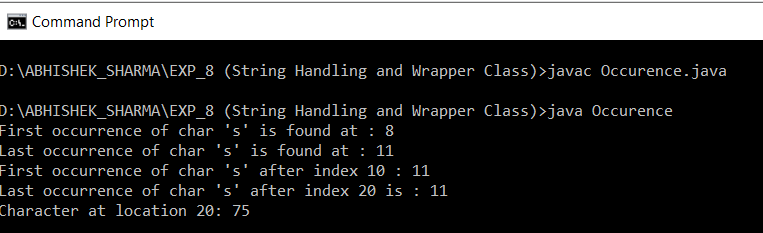
System.out.println("Last occurrence of char 's'" + " after index 20 is : " + last\_in);

int char\_at = str.charAt(20);

System.out.println("Character at location 20: " + char\_at);

}

}



**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.**

class Letters{

static void convertOpposite(StringBuffer str)

{

int ln = str.length();

for (int i=0; i<ln; i++)

{

Character c = str.charAt(i);

if (Character.isLowerCase(c))

str.replace(i, i+1, Character.toUpperCase(c)+"");

else

str.replace(i, i+1, Character.toLowerCase(c)+"");

}

}

public static void main(String[] args)

{

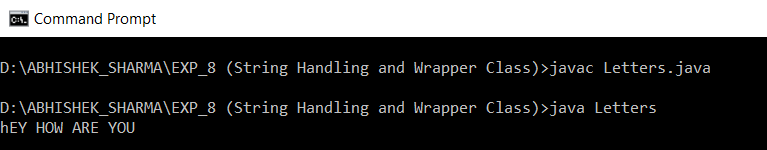
StringBuffer str = new StringBuffer("Hey how are you");

convertOpposite(str);

System.out.println(str);

}

}



**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)**

import java.io.\*;

class LettersCL

{

public static void main(String a[]) throws IOException

{

DataInputStream in=new DataInputStream(System.in);

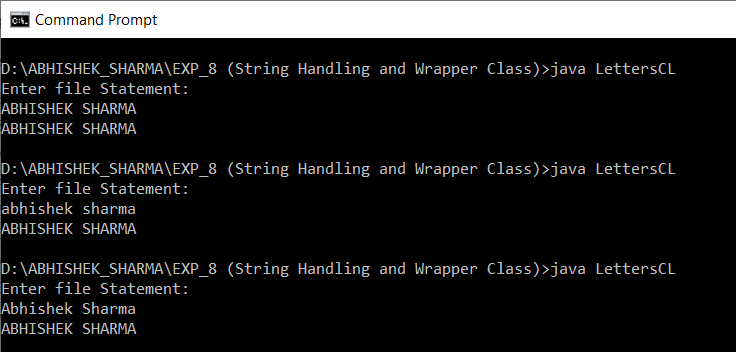
System.out.println("Enter file Statement:");

String s1=in.readLine();

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

}

}



**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 objects.**

class data{

String name;

data(String n){ name=n; }

void disp(){

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

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 occurance :"+c);

System.out.println("Possition :"+(i+1));

}

if(c==0)

System.out.println("there is no 'A' available in the string");

}

}

class NameLocation{

public static void main(String ar[])

{

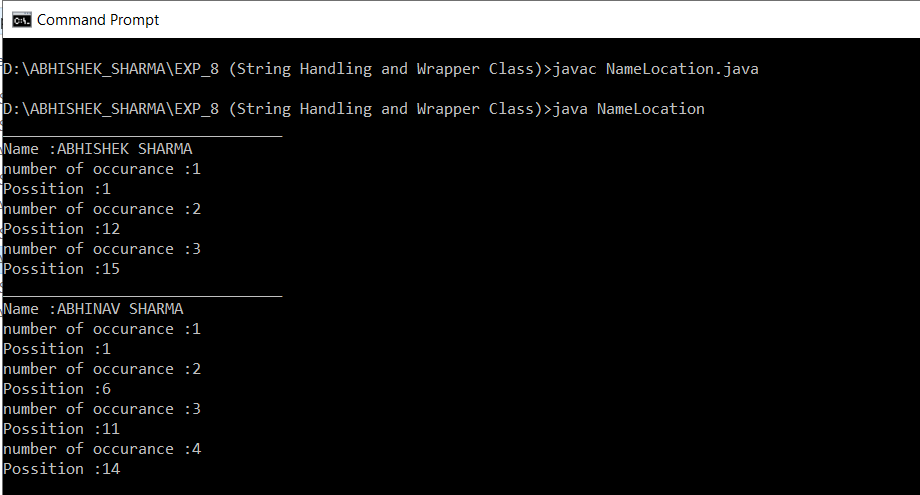
data d1=new data("ABHISHEK SHARMA");

d1.disp();

data d2=new data("ABHINAV SHARMA");

d2.disp();

}}



**TITLE: Wrapper Classes**

1. **Write a Java code that converts int to Integer, converts Integer to String, converts String to int, converts int to String, converts String to Integer converts Integer to int.**

public class ConvertingType1

{

public static void main(String[] args)

{

Integer num = new Integer(20);

System.out.println(num.intValue());

int c = 1234;

String string = String.valueOf(c);

System.out.println("String = " + string);

String str = "123";

int inum = 100;

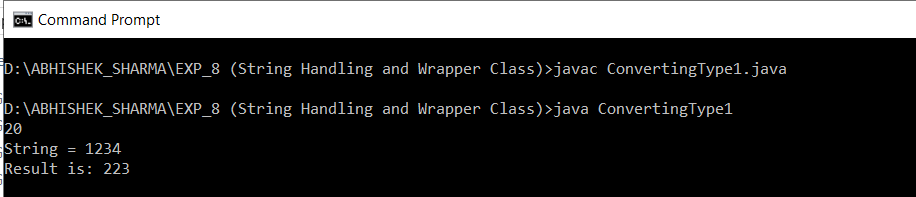
int inum2 = Integer.parseInt(str);

int sum = inum + inum2;

System.out.println("Result is: " + sum);

}

}



1. **Write a Java code that converts float to Float converts Float to String converts String to float converts float to String converts String to Float converts Float to float.**

public class ConvertingType2

{

public static void main(String[] args)

{

Float fObj = new Float("78.50");

byte b = fObj.byteValue();

System.out.println(b);

short s = fObj.shortValue();

System.out.println(s);

int i = fObj.intValue();

System.out.println(i);

float f = fObj.floatValue();

System.out.println(f);

double d = fObj.doubleValue();

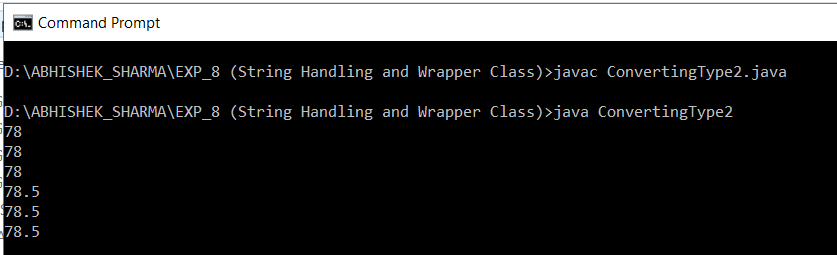
System.out.println(d);

String si = fObj.toString();

System.out.println(si);

}

}



**EXPERIMENT – 9**

**TITLE: Threads and Collections**

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

class MyThread extends Thread

{

public void run(){

try {

System.out.println("Thread class extended"); Thread.sleep(10);

} catch (Exception e) {

}

}

}

class MyRun implements Runnable

{

public void run(){

System.out.println("Runnable interface implemented");

}

}

public class Test{

public static void main(String[] args) {

MyThread objMyThread =new MyThread();

MyRun objMyRun = new MyRun();

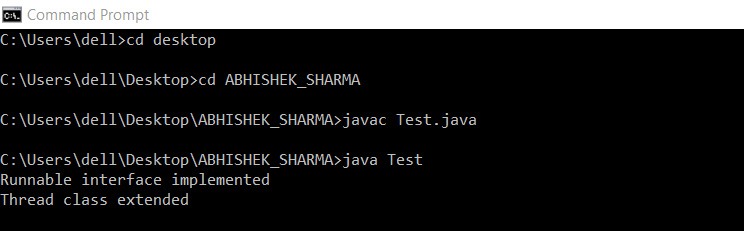
Thread t2 = new Thread(objMyRun);

objMyThread.start();

t2.start();

}

}



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

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 Even\_Odd{

public static void main(String arg[])

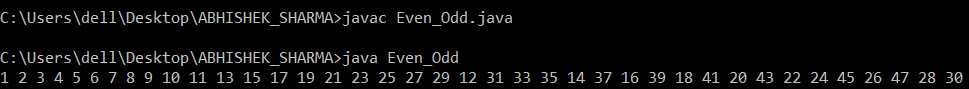
{

even e=new even();

odd o=new odd();

}

}



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

class synchronization

{

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)

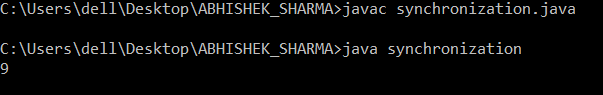
{

i.count++;

return(i);

}

}



1. **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**

import java.lang.\*;

class ThreadDemo extends Thread

{

public void run()

{

System.out.println("Inside run method");

}

public static void main(String[]args)

{

ThreadDemo t1 = new ThreadDemo();

ThreadDemo t2 = new ThreadDemo();

ThreadDemo t3 = new ThreadDemo();

System.out.println("t1 thread priority : " +

t1.getPriority()); // Default 5

System.out.println("t2 thread priority : " +

t2.getPriority()); // Default 5

System.out.println("t3 thread priority : " +

t3.getPriority()); // Default 5

t1.setPriority(2);

t2.setPriority(5);

t3.setPriority(8);

// t3.setPriority(21); will throw IllegalArgumentException

System.out.println("t1 thread priority : " +

t1.getPriority()); //2

System.out.println("t2 thread priority : " +

t2.getPriority()); //5

System.out.println("t3 thread priority : " +

t3.getPriority());//8

// Main thread

System.out.print(Thread.currentThread().getName());

System.out.println("Main thread priority : "

+ Thread.currentThread().getPriority());

// Main thread priority is set to 10

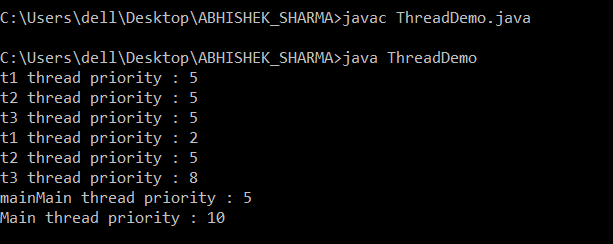
Thread.currentThread().setPriority(10);

System.out.println("Main thread priority : "

+ Thread.currentThread().getPriority());

}

}



**TITLE: Collections**

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

**Read all elements from ArrayList by using Iterator.**

**Create duplicate object of an ArrayList instance.**

**Reverse ArrayList content.**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.Iterator;**

//ArrayListIterator

class MyArrayListReverse{

public static void main(String a[]){

ArrayList<String> arrl = new ArrayList<String>();

//adding elements to the end

arrl.add("First");

arrl.add("Second");

arrl.add("Third");

arrl.add("Fourth");

Iterator<String> itr = arrl.iterator();

while(itr.hasNext()){

System.out.println(itr.next());

}

System.out.println("Actual ArrayList: "+arrl);

ArrayList<String> copy = (ArrayList<String>) arrl.clone();

System.out.println("Copied/Cloned ArrayList: "+copy);

Collections.reverse(arrl);

System.out.println("Reversed ArrayList: "+arrl);

}

}



1. **Write a program for the following HashMap**

**find whether specified key exists or not.**

**find whether specified value exists or not**

**get all keys from the given HashMap**

**get all key-value pair as Entry objects**

import java.util.\*;

//CheckKeyOfHashMapExample

public class MyHashMapKeySearch {

public static void main(String[] args) {

HashMap<String,String> hMap = new HashMap<String,String>();

hMap.put("1","One");

hMap.put("2","Two");

hMap.put("3","Three");

System.out.println("HashMap: "+hMap);

System.out.println("Checking whether '3' key exists in HashMap : " + hMap.containsKey("3"));

System.out.println("Checking whether 'One' Value exists in HashMap : " + hMap.containsValue("One"));

System.out.println("Checking whether 'abc' Value exists in HashMap : " + hMap.containsValue("abc"));

Set<String> keys = hMap.keySet();

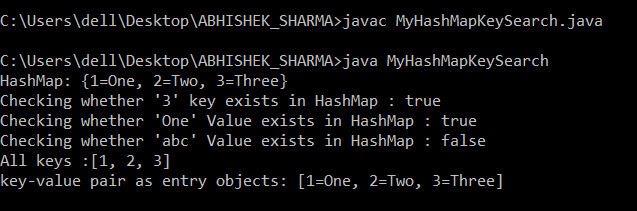
System.out.println("All keys :"+keys);

Set<HashMap.Entry<String, String>> ent = hMap.entrySet();

System.out.println("key-value pair as entry objects: "+ent);

}

}



1. **Write a program for the following HashSet**

**copy another collection object to HashSet object.**

**delete all entries at one call from HashSet**

**search user defined objects from HashSet**

import java.util.\*;

class userDefined{

String value;

int n;

public userDefined(String v, int n){

this.value = v;

this.n = n;

}

public int hashCode(){

// System.out.println("In hashcode");

int hashcode = 0;

hashcode = n\*20;

hashcode += value.hashCode();

return hashcode;

}

public boolean equals(Object obj){

// System.out.println("In equals");

if (obj instanceof userDefined) {

userDefined ud = (userDefined) obj;

return (ud.value.equals(this.value) && ud.n == this.n);

} else {

return false;

}

}

}

public class MyHashSetCopy

{

public static void main(String[] args)

{

HashSet<String> hSet = new HashSet<String>();

hSet.add("first");

hSet.add("second");

hSet.add("third");

System.out.println("Hash set: "+hSet);

List<String> l = new ArrayList<String>();

l.add("list element 1");

l.add("list element 2");

System.out.println("Array List: "+l);

hSet.addAll(l);

System.out.println("Hash set after copying array list to hash set : "+hSet);

hSet.clear();

System.out.println("Deleting all elements from hash set");

System.out.println("Hash set: "+hSet);

hSet.add("element 1");

System.out.println("Adding 'element 1' in hash set");

System.out.println("Hash set: "+hSet);

HashSet<userDefined> uSet = new HashSet<userDefined>();

uSet.add(new userDefined("object 1", 1));

uSet.add(new userDefined("object 2", 2));

uSet.add(new userDefined("object 3", 3));

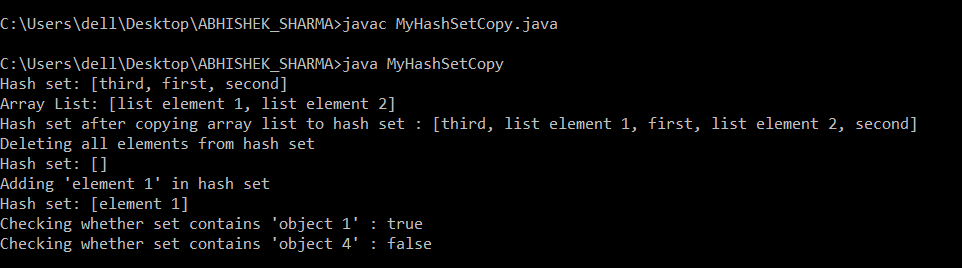
// System.out.println("Hash set with user defined objects: "+uSet);

System.out.println("Checking whether set contains 'object 1' : "+uSet.contains(new userDefined("object 1", 1)));

System.out.println("Checking whether set contains 'object 4' : "+uSet.contains(new userDefined("object 4", 1)));

}

}



**EXPERIMENT – 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.**

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.DriverManager;

public class jdbcConnClass {

public static void main(String[] args){

Connection con=null;

Statement stmt=null;

PreparedStatement pstmt=null;

ResultSet rs =null;

try{

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/animeshJAVA","root","root");

System.out.println("successful connection!");

// stmt=con.createStatement();

stmt=con.createStatement();

String q="create table if not exists employee(Eno int(5) primary key,Ename varchar(20));";

stmt.executeUpdate(q);

// System.out.println("employee table created successfuly!");

String q1="insert into employee values(?,?);";

pstmt=con.prepareStatement(q1);

stmt.executeUpdate("delete from employee;");

pstmt.setInt(1, 1);

pstmt.setString(2, "Ani");

pstmt.executeUpdate();

pstmt.setInt(1, 2);

pstmt.setString(2, "animesh");

pstmt.executeUpdate();

rs=stmt.executeQuery("select \* from employee;");

System.out.println("Table data:");

if(rs.first())

{

do{

System.out.println("Eno= "+rs.getInt(1)+" , Ename= "+rs.getString(2));

}while(rs.next());

}

else{System.out.println("Empty table");}

}catch(Exception e)

{System.out.println(e);}

finally{

try{

if(stmt!=null){

stmt.close();

}

if(con!=null){

con.close();

}}catch(Exception e)

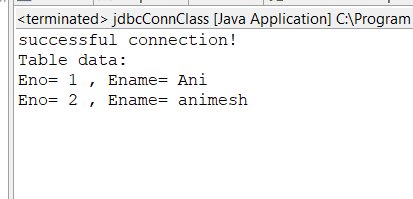
{System.out.println(e);}

}

}

}

Output:



**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.**

Input:

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.DriverManager;

public class jdbcConnClass {

public static void main(String[] args){

Connection con=null;

Statement stmt=null;

PreparedStatement pstmt=null;

ResultSet rs =null;

try{

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/animeshJAVA","root","root");

System.out.println("successful connection!");

// stmt=con.createStatement();

stmt=con.createStatement();

String q="create table if not exists employee(Eno int(5) primary key,Ename varchar(20));";

stmt.executeUpdate(q);

// System.out.println("employee table created successfuly!");

String q1="insert into employee values(?,?);";

pstmt=con.prepareStatement(q1);

stmt.executeUpdate("delete from employee;");

pstmt.setInt(1, 1);

pstmt.setString(2, "Ani");

pstmt.executeUpdate();

pstmt.setInt(1, 2);

pstmt.setString(2, "animesh");

pstmt.executeUpdate();

rs=stmt.executeQuery("select \* from employee;");

System.out.println("Table data:");

if(rs.first())

{

do{

System.out.println("Eno= "+rs.getInt(1)+" , Ename= "+rs.getString(2));

}while(rs.next());

}

else{System.out.println("Empty table");}

}catch(Exception e)

{System.out.println(e);}

finally{

try{

if(stmt!=null){

stmt.close();

}

if(con!=null){

con.close();

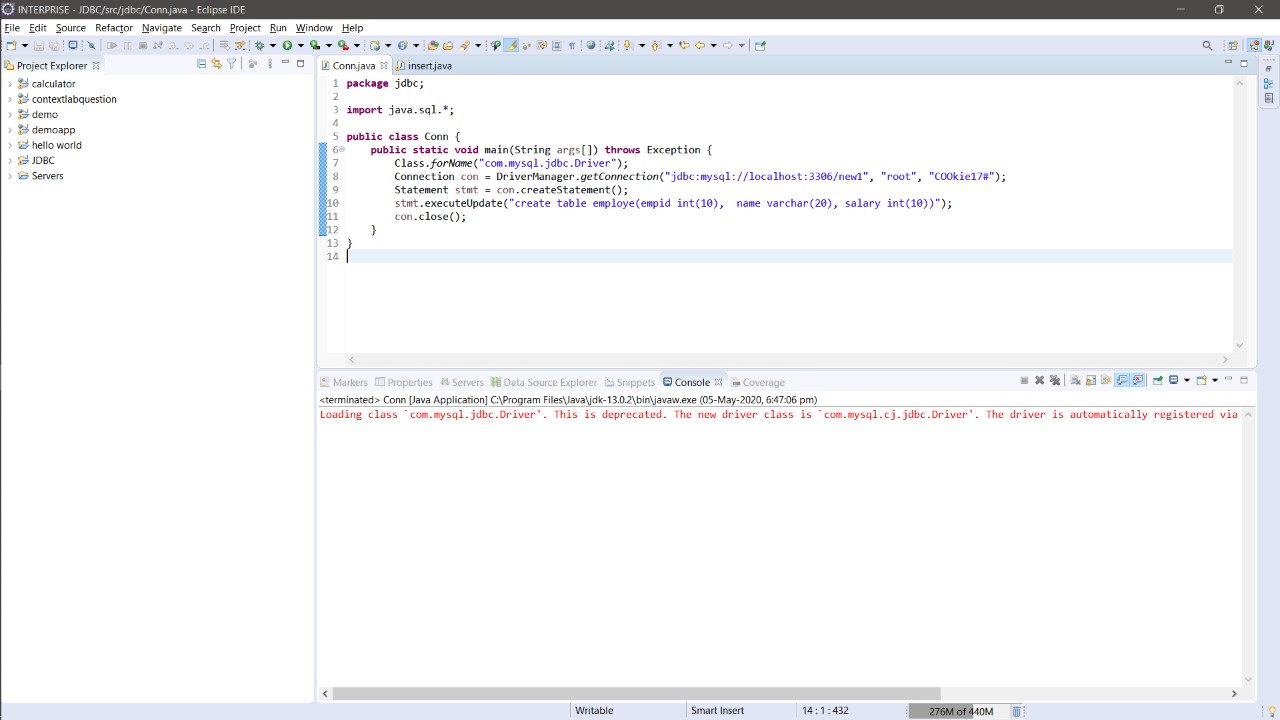
}}catch(Exception e)

{System.out.println(e);}

}

}

}



**EXPERIMENT – 11**

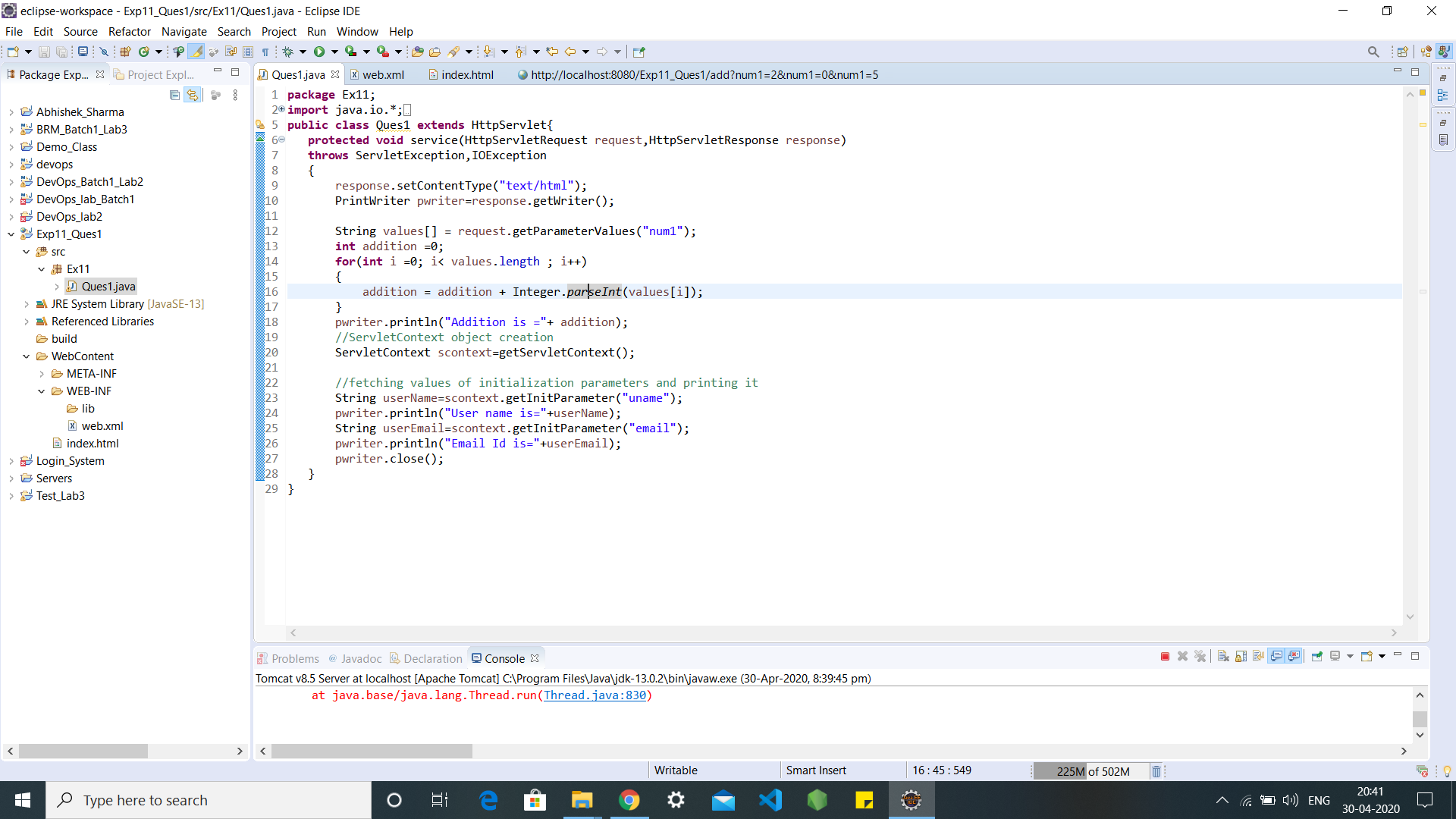
**TITLE: *Servlet***

1. **Servlet:**

**a) ServletContext interface**

**b) getParameterValues( ) of Servlet Request**

IMAGE: -



CODE: -

package Ex11;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Ques1 extends HttpServlet{

protected void service(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

response.setContentType("text/html");

PrintWriter pwriter=response.getWriter();

String values[] = request.getParameterValues("num1");

int addition =0;

for(int i =0; i< values.length ; i++)

{

addition = addition + Integer.parseInt(values[i]);

}

pwriter.println("Addition is ="+ addition);

//ServletContext object creation

ServletContext scontext=getServletContext();

//fetching values of initialization parameters and printing it

String userName=scontext.getInitParameter("uname");

pwriter.println("User name is="+userName);

String userEmail=scontext.getInitParameter("email");

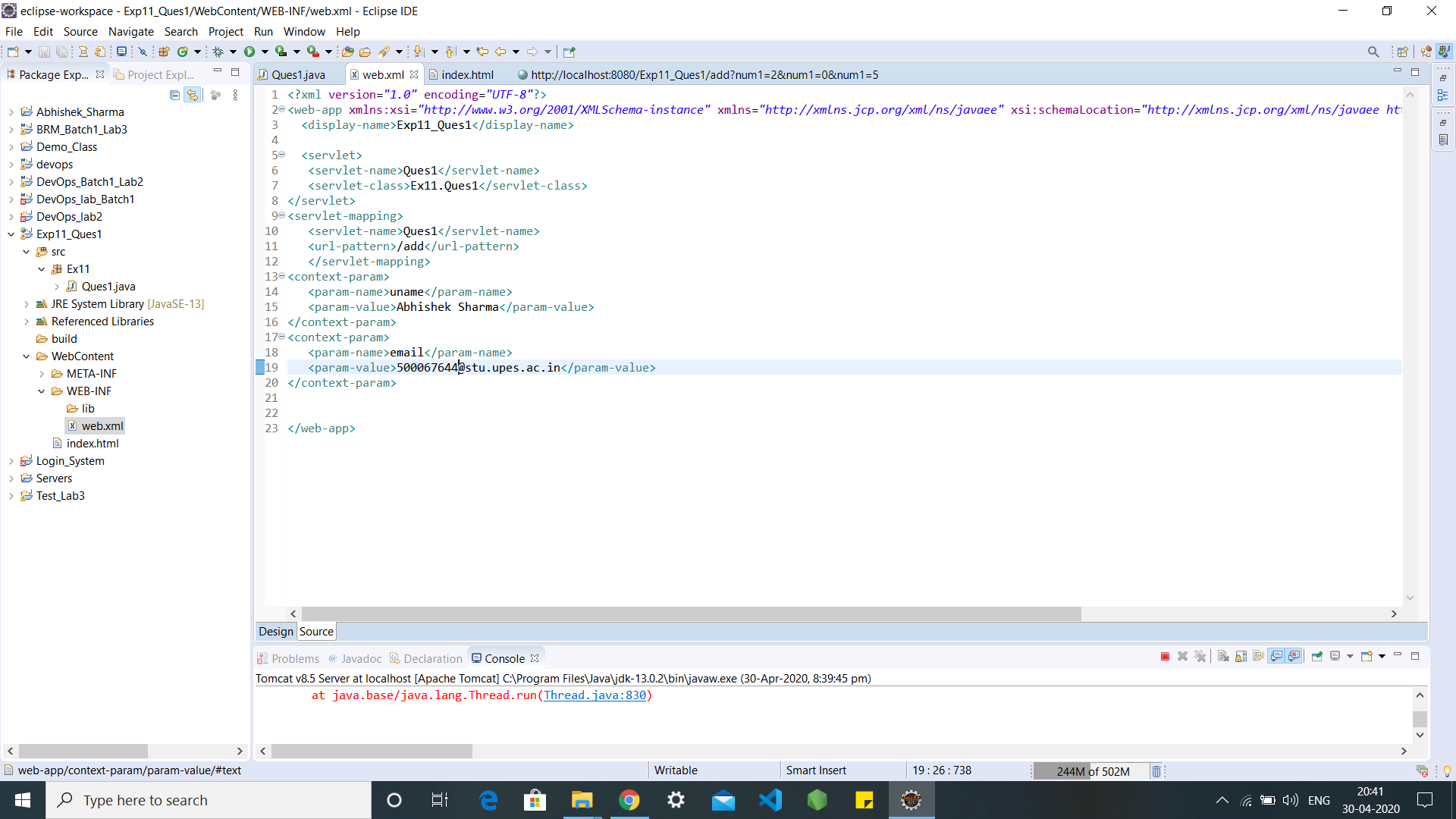
pwriter.println("Email Id is="+userEmail);

pwriter.close();

}

}

IMAGE: -



CODE: -

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" version="4.0">

<display-name>Exp11\_Ques1</display-name>

<servlet>

<servlet-name>Ques1</servlet-name>

<servlet-class>Ex11.Ques1</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Ques1</servlet-name>

<url-pattern>/add</url-pattern>

</servlet-mapping>

<context-param>

<param-name>uname</param-name>

<param-value>Abhishek Sharma</param-value>

</context-param>

<context-param>

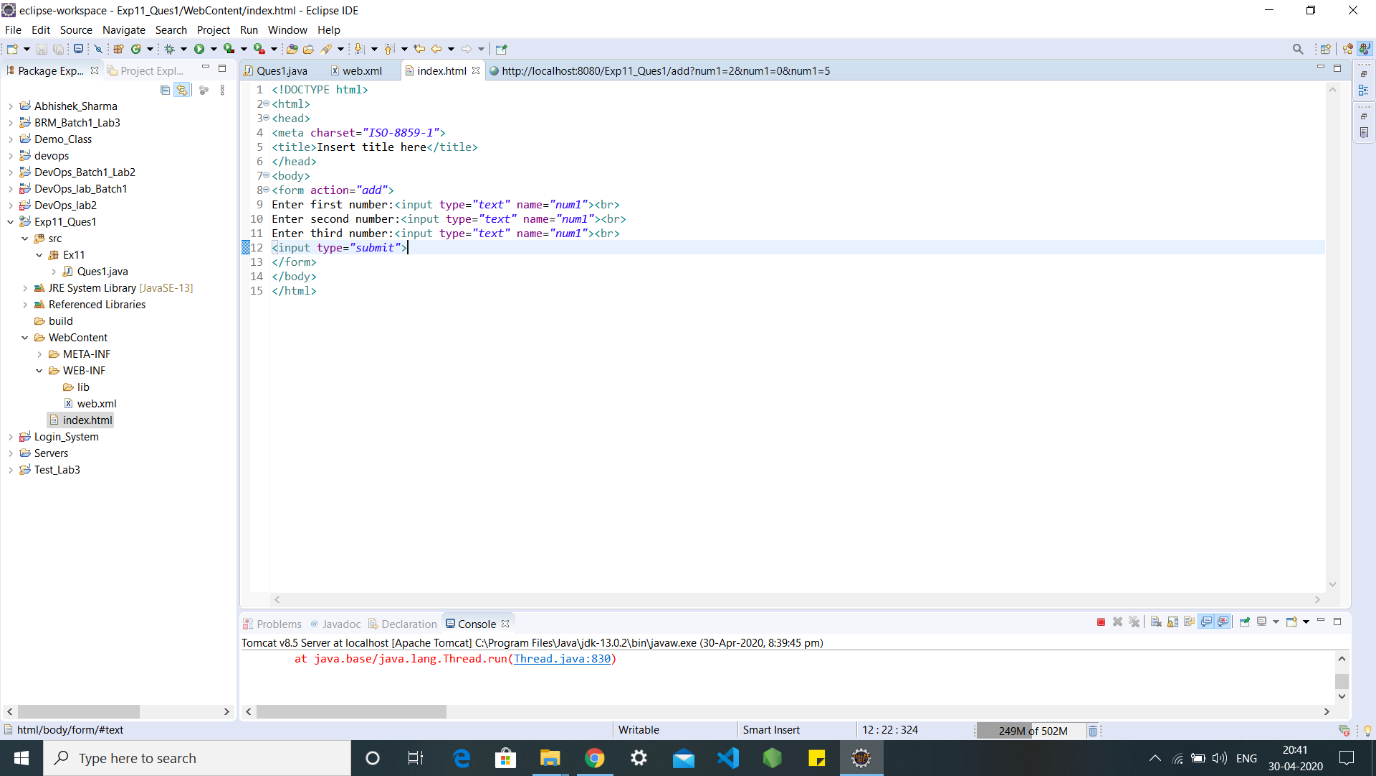
<param-name>email</param-name>

<param-value>500067644@stu.upes.ac.in</param-value>

</context-param>

</web-app>

IMAGE: -



CODE: -

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<form action="add">

Enter first number:<input type="text" name="num1"><br>

Enter second number:<input type="text" name="num1"><br>

Enter third number:<input type="text" name="num1"><br>

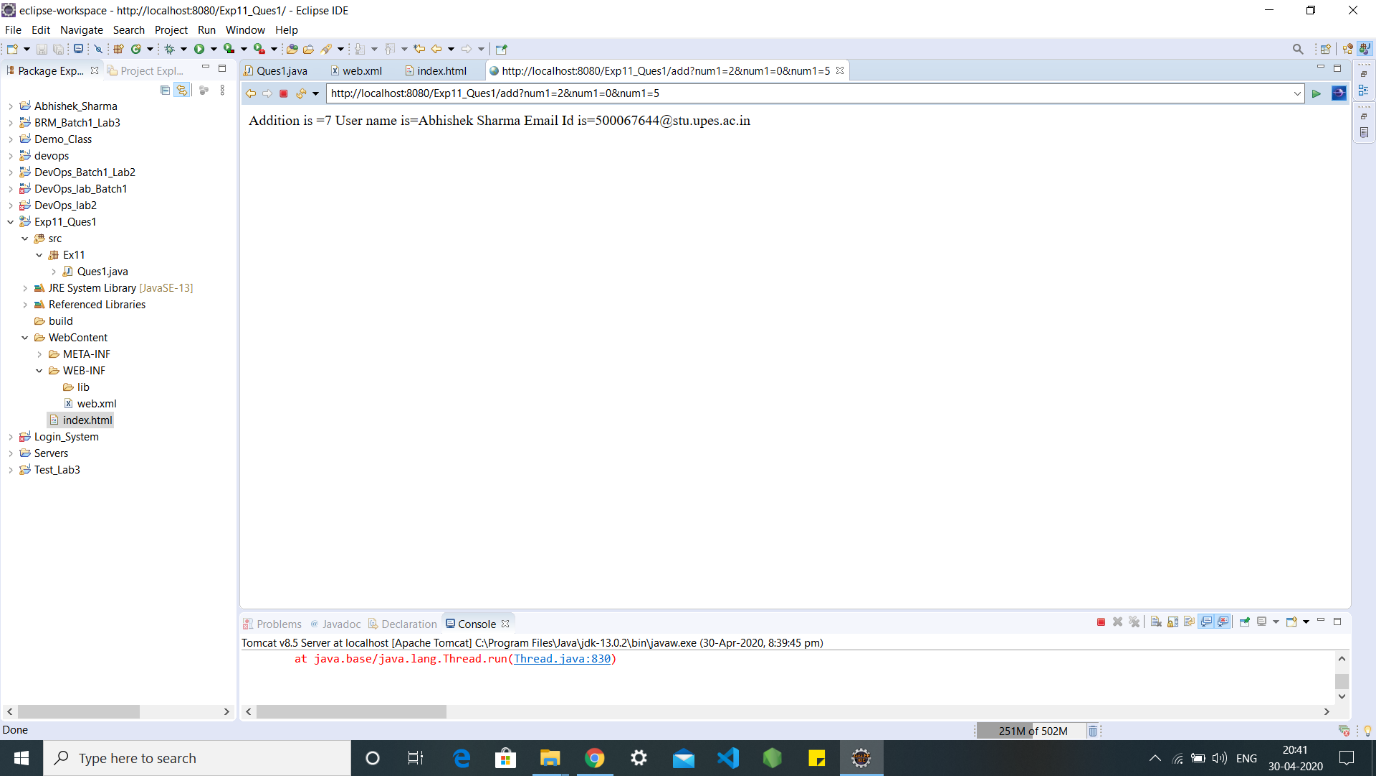
<input type="submit">

</form>

</body>

</html>

OUTPUT: -



1. **Write a Servlet page to display current date of the server.**

**IMAGE: -**



CODE: -

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Ques2 extends HttpServlet{

public void doGet(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException{

PrintWriter pw = response.getWriter();

Date today = new Date();

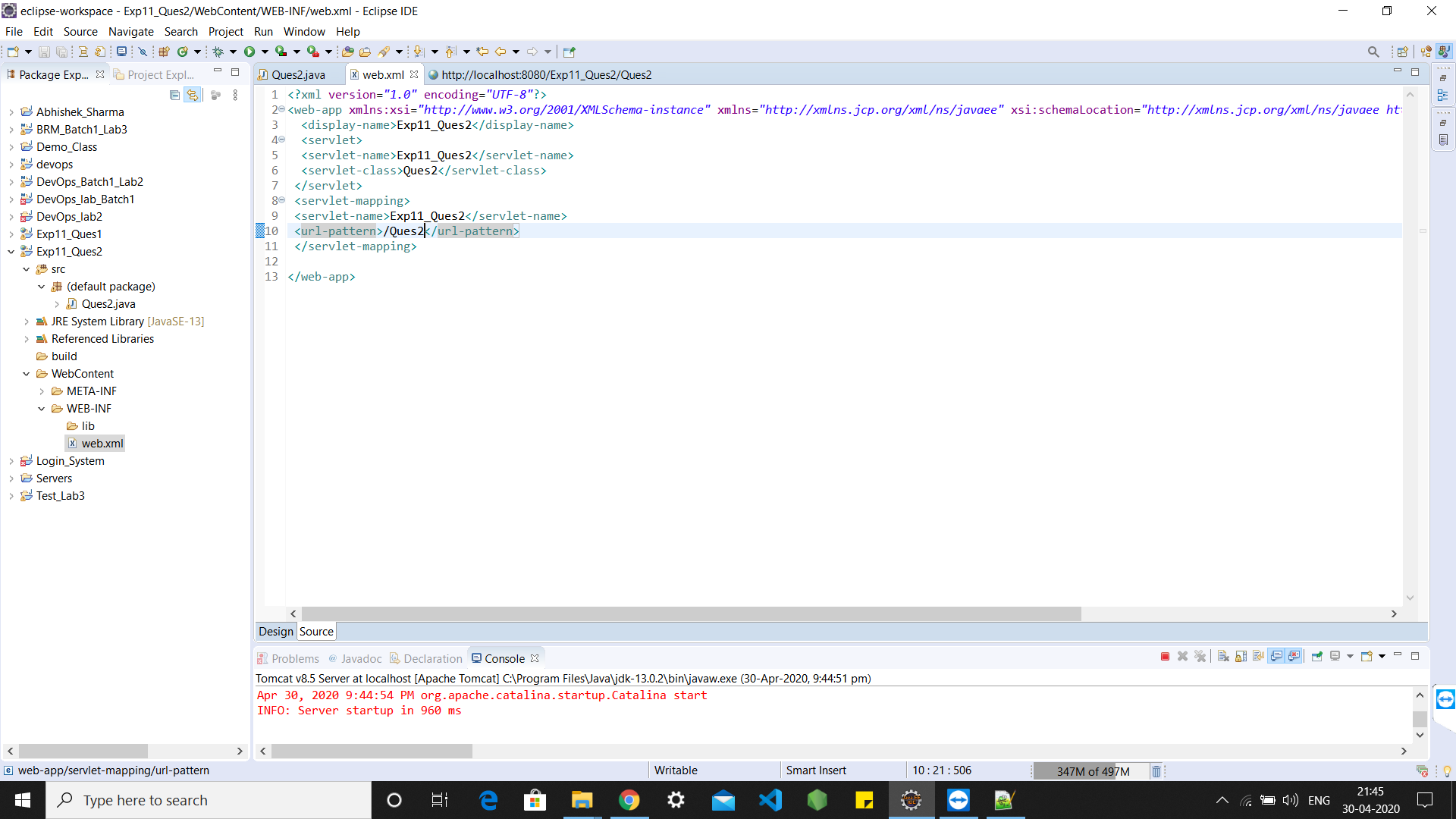
pw.println("<html>"+"<body><h1>Today's Date is</h1>");

pw.println("<b>"+ today+"</b></body>"+ "</html>");

}

}

IMAGE: -



CODE: -

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" version="4.0">

<display-name>Exp11\_Ques2</display-name>

<servlet>

<servlet-name>Exp11\_Ques2</servlet-name>

<servlet-class>Ques2</servlet-class>

</servlet>

<servlet-mapping>

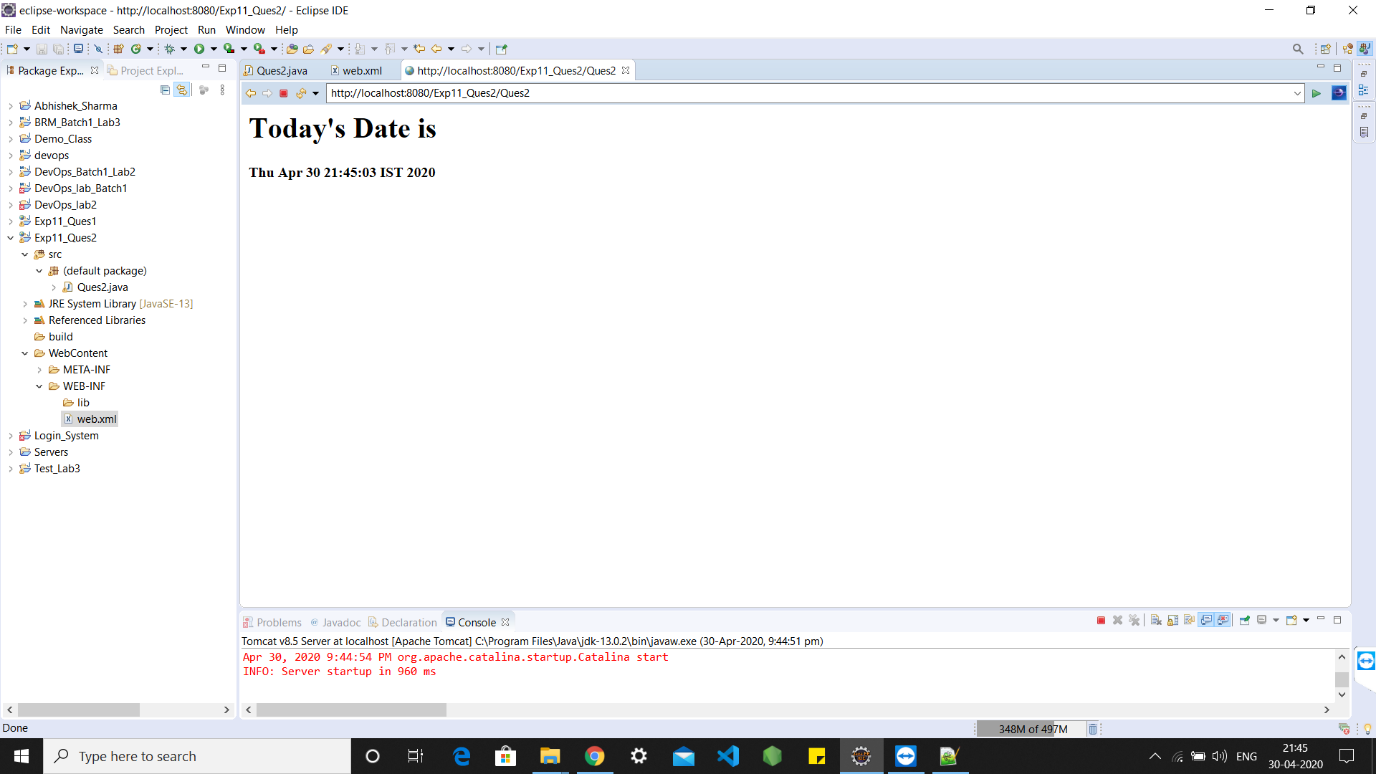
<servlet-name>Exp11\_Ques2</servlet-name>

<url-pattern>/Ques2</url-pattern>

</servlet-mapping>

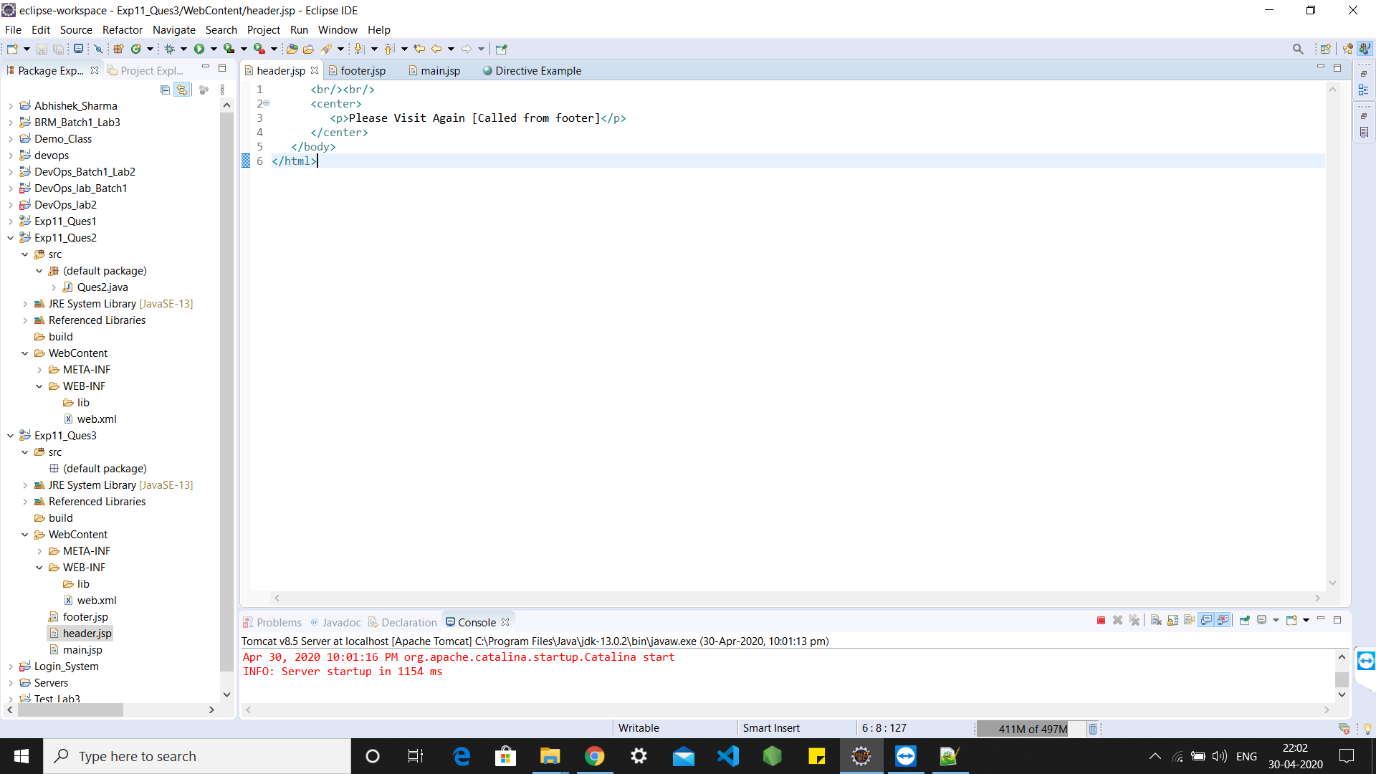
</web-app>

OUTPUT: -



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

IMAGE: -



CODE: -

<br/><br/>

<center>

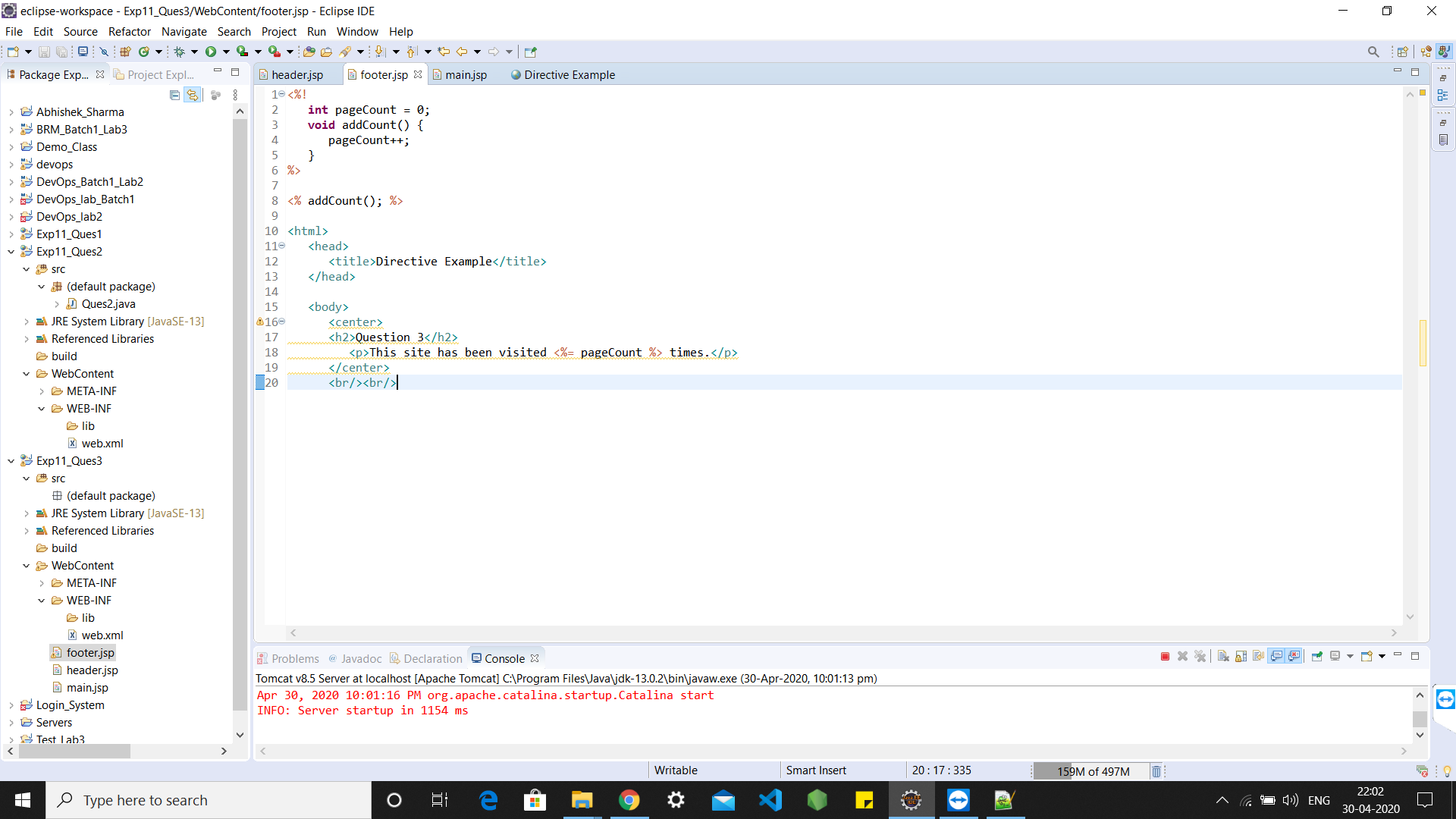
<p>Please Visit Again [Called from footer]</p>

</center>

</body>

</html>

IMAGE: -



CODE: -

<%!

int pageCount = 0;

void addCount() {

pageCount++;

}

%>

<% addCount(); %>

<html>

<head>

<title>Directive Example</title>

</head>

<body>

<center>

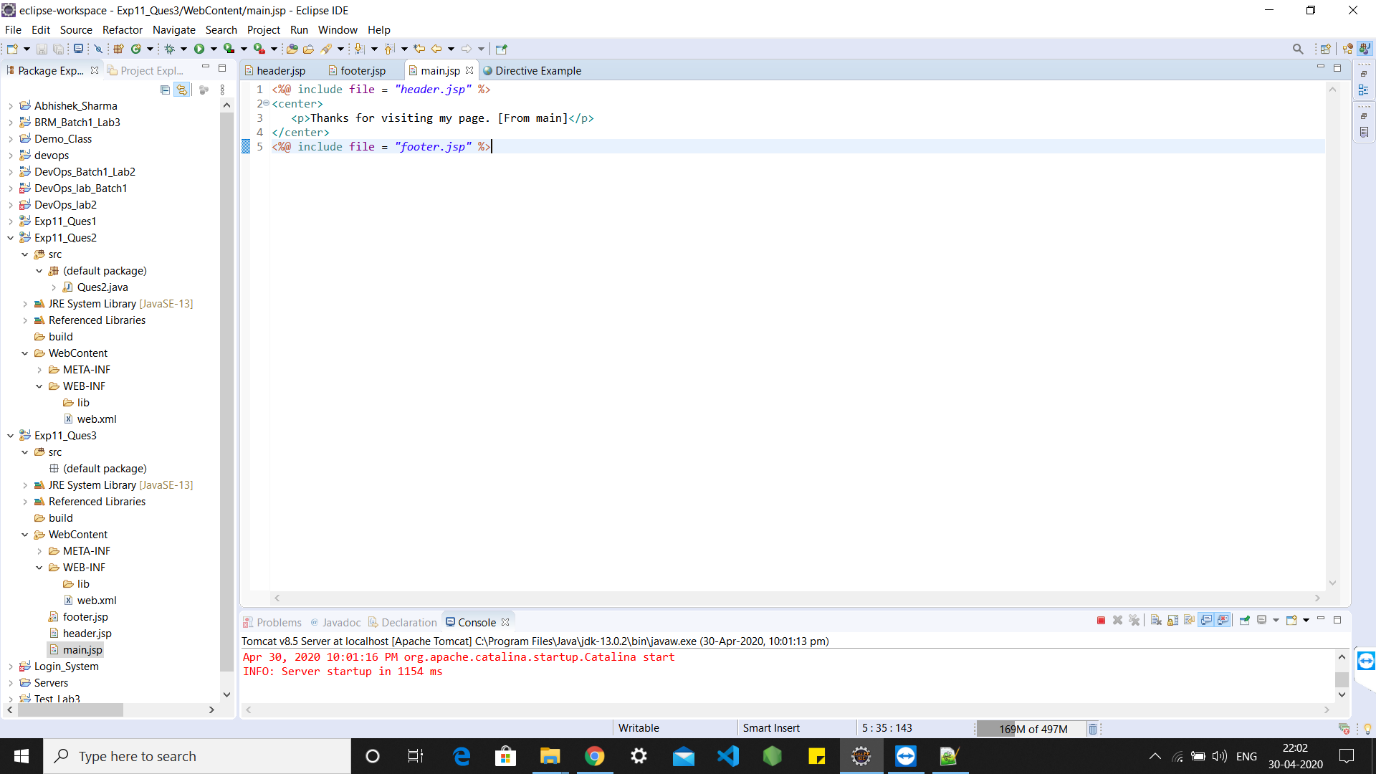
<h2>Question 3</h2>

<p>This site has been visited <%= pageCount %> times.</p>

</center>

<br/><br/>

IMAGE: -



CODE: -

<%@ include file = "header.jsp" %>

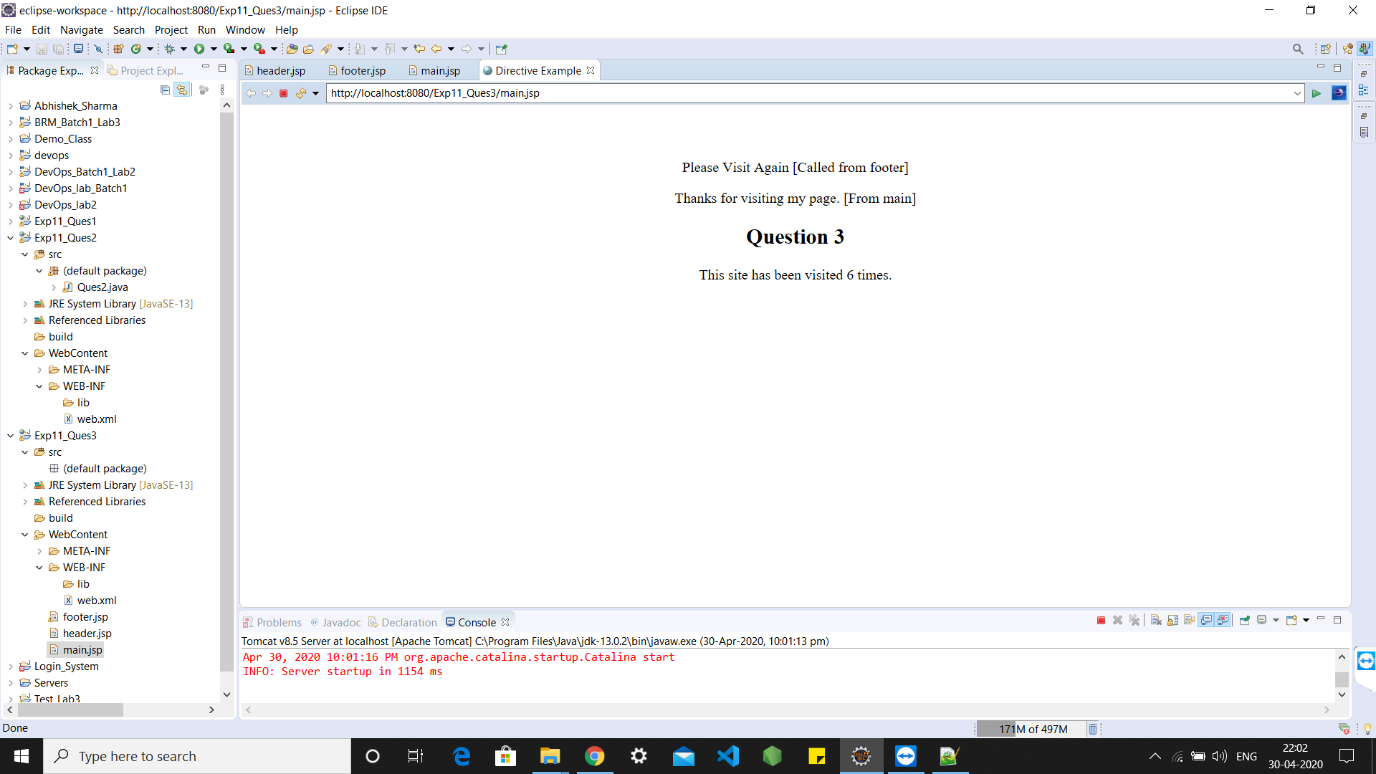
<center>

<p>Thanks for visiting my page. [From main]</p>

</center>

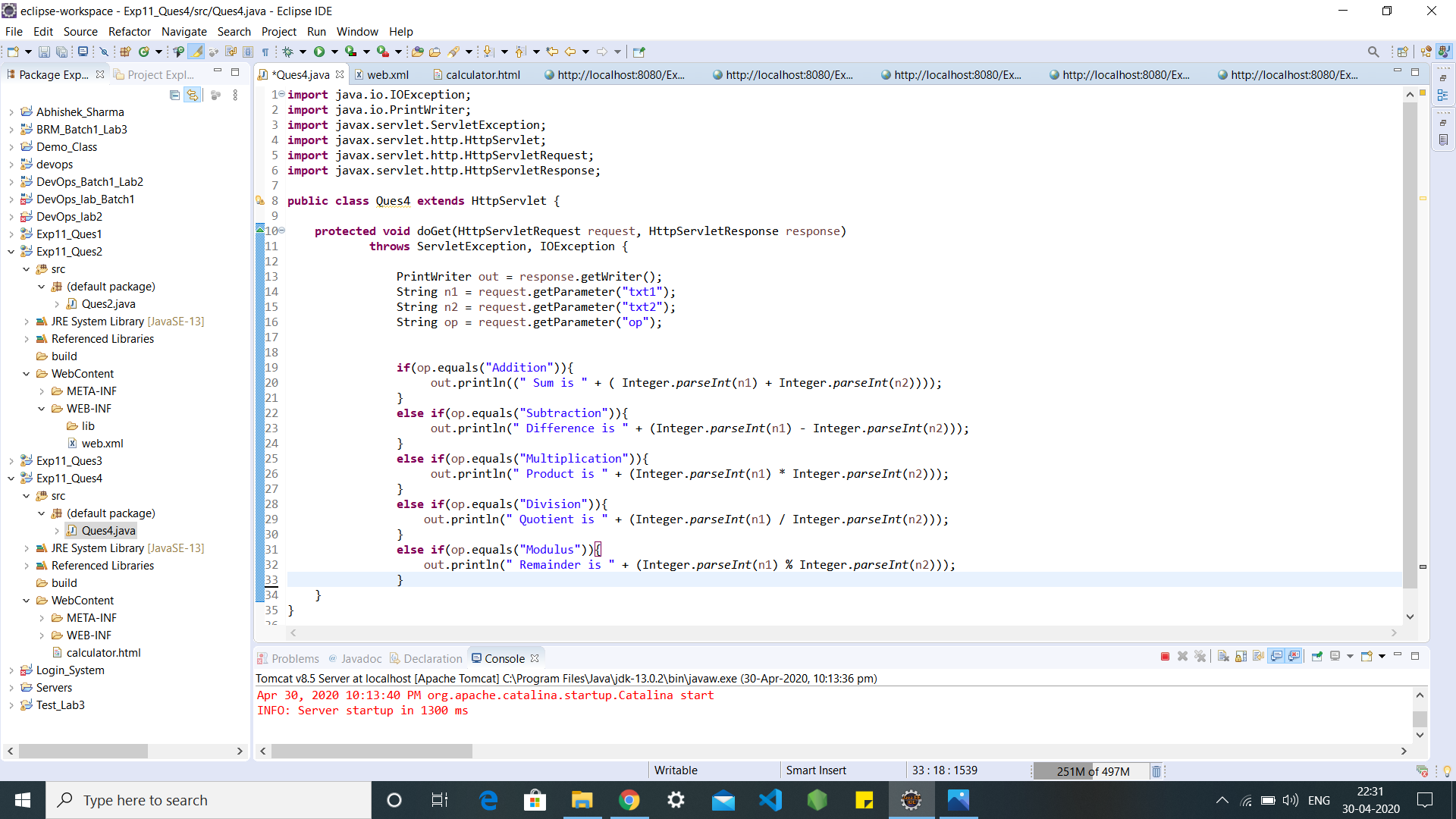
<%@ include file = "footer.jsp" %>

OUTPUT: -



1. **Write a Servlet page to create a simple calculator.**

**IMAGE: -**



CODE: -

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Ques4 extends HttpServlet {

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");

if(op.equals("Addition")){

out.println((" Sum is " + ( Integer.parseInt(n1) + Integer.parseInt(n2))));

}

else if(op.equals("Subtraction")){

out.println(" Difference is " + (Integer.parseInt(n1) - Integer.parseInt(n2)));

}

else if(op.equals("Multiplication")){

out.println(" Product is " + (Integer.parseInt(n1) \* Integer.parseInt(n2)));

}

else if(op.equals("Division")){

out.println(" Quotient is " + (Integer.parseInt(n1) / Integer.parseInt(n2)));

}

else if(op.equals("Modulus")){

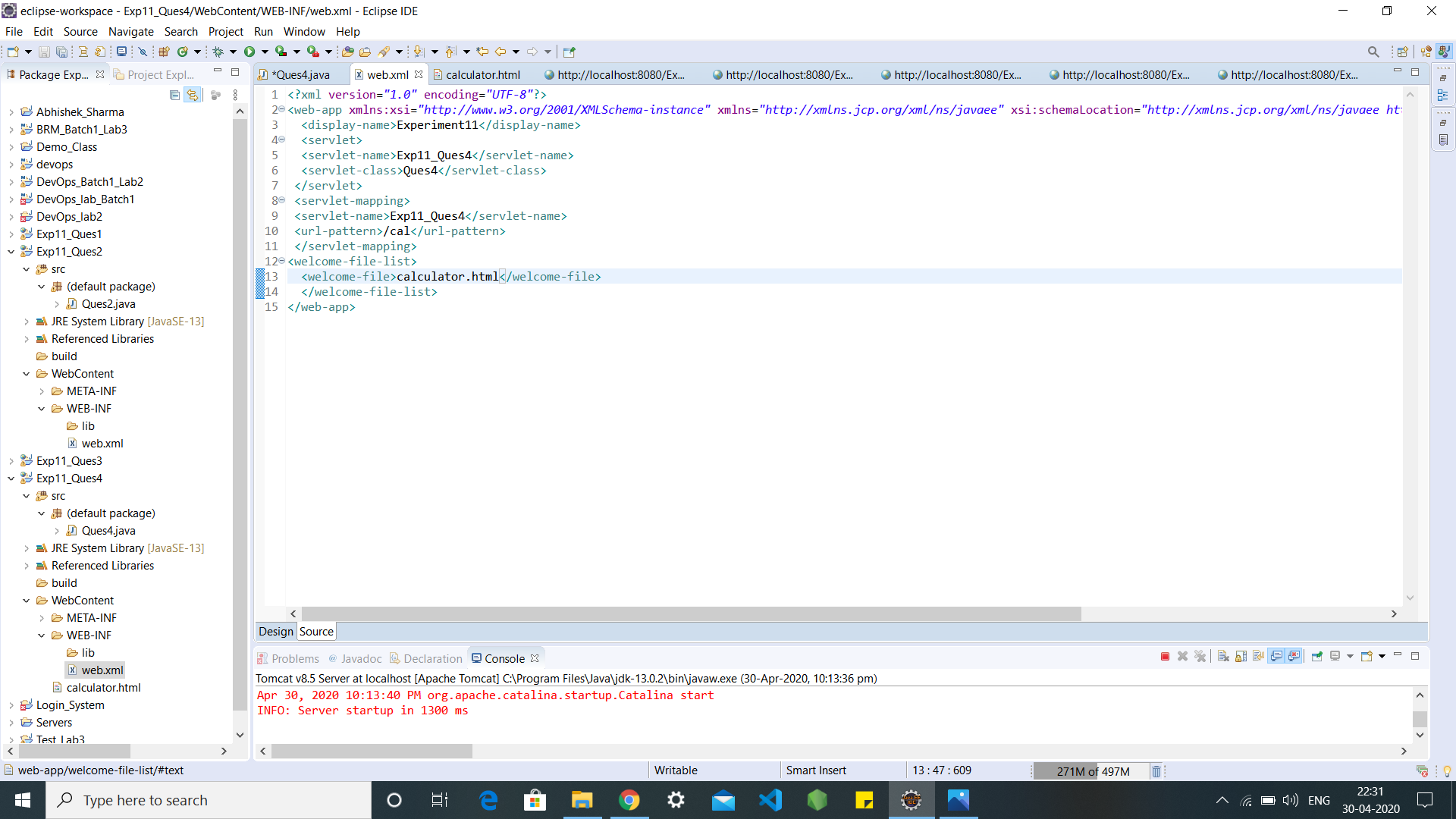
out.println(" Remainder is " + (Integer.parseInt(n1) % Integer.parseInt(n2)));

}

}

}

IMAGE: -



CODE: -

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" version="4.0">

<display-name>Experiment11</display-name>

<servlet>

<servlet-name>Exp11\_Ques4</servlet-name>

<servlet-class>Ques4</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Exp11\_Ques4</servlet-name>

<url-pattern>/cal</url-pattern>

</servlet-mapping>

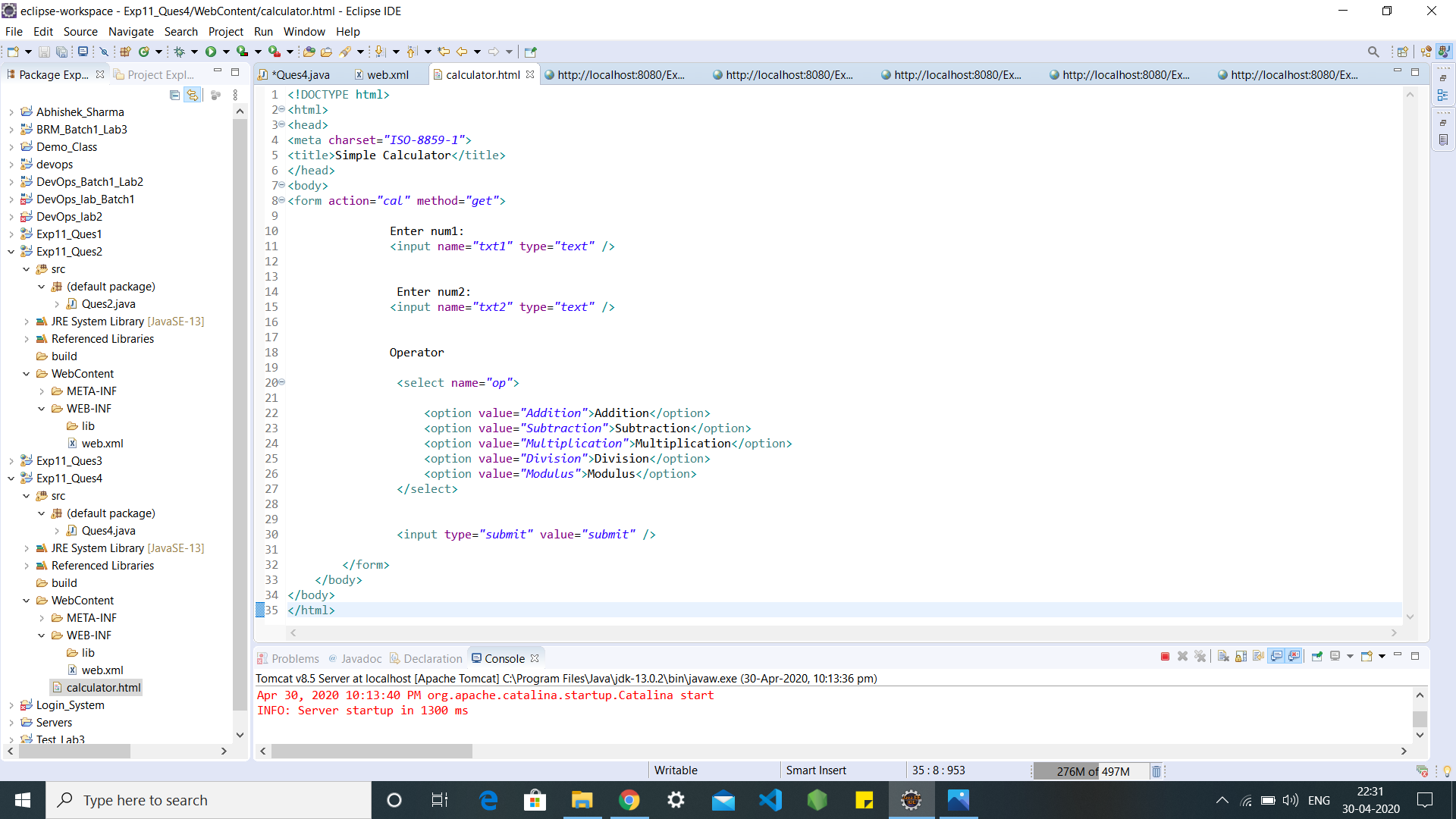
<welcome-file-list>

<welcome-file>calculator.html</welcome-file>

</welcome-file-list>

</web-app>

IMAGE: -



CODE: -

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Simple Calculator</title>

</head>

<body>

<form action="cal" method="get">

Enter num1:

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

Enter num2:

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

Operator

<select name="op">

<option value="Addition">Addition</option>

<option value="Subtraction">Subtraction</option>

<option value="Multiplication">Multiplication</option>

<option value="Division">Division</option>

<option value="Modulus">Modulus</option>

</select>

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

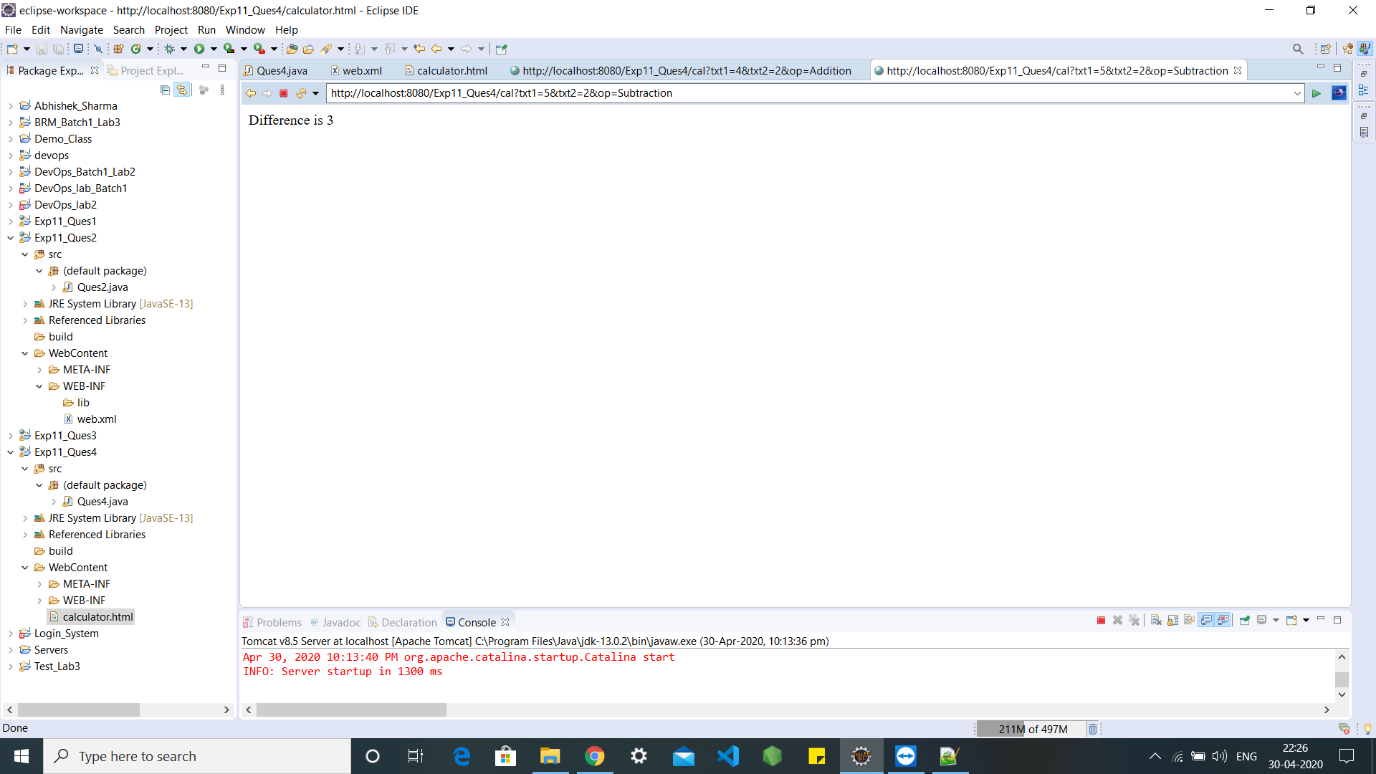
</form>

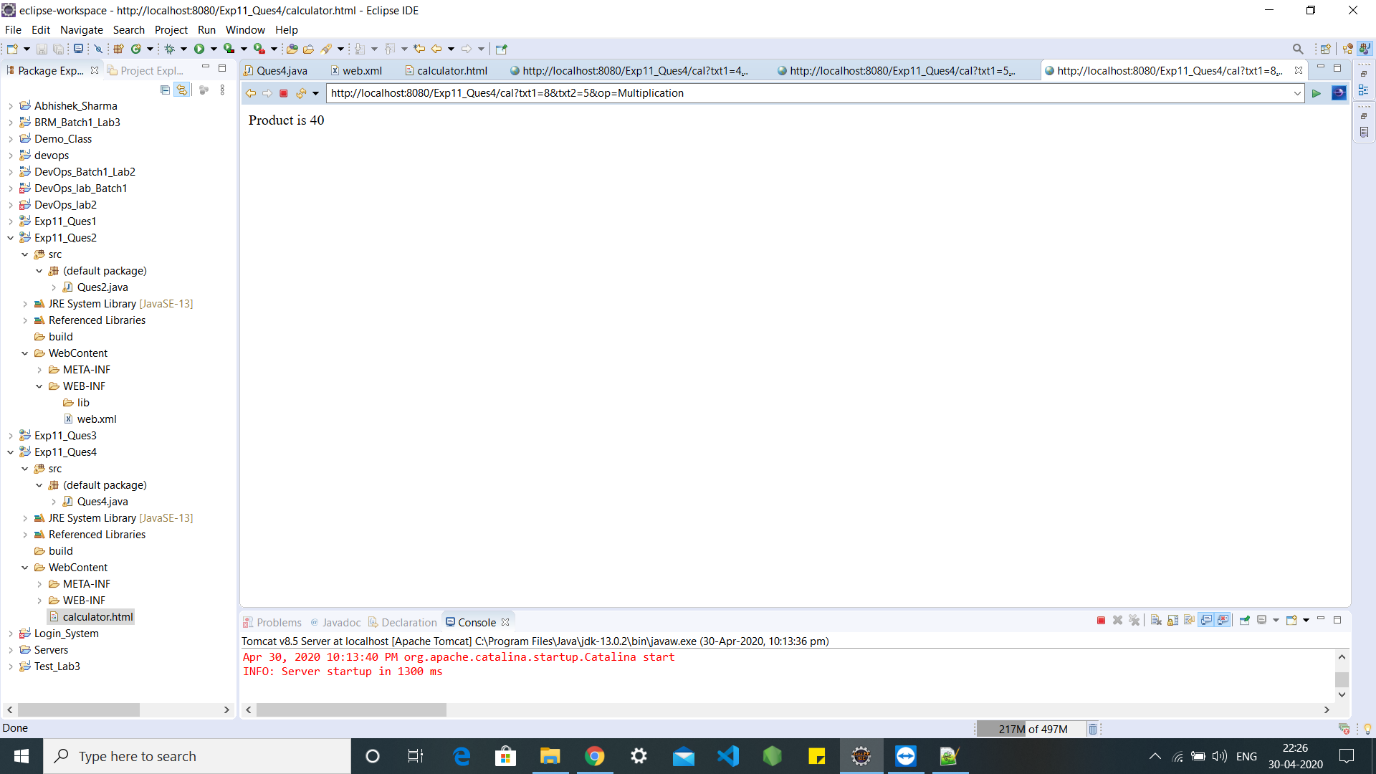
</body>

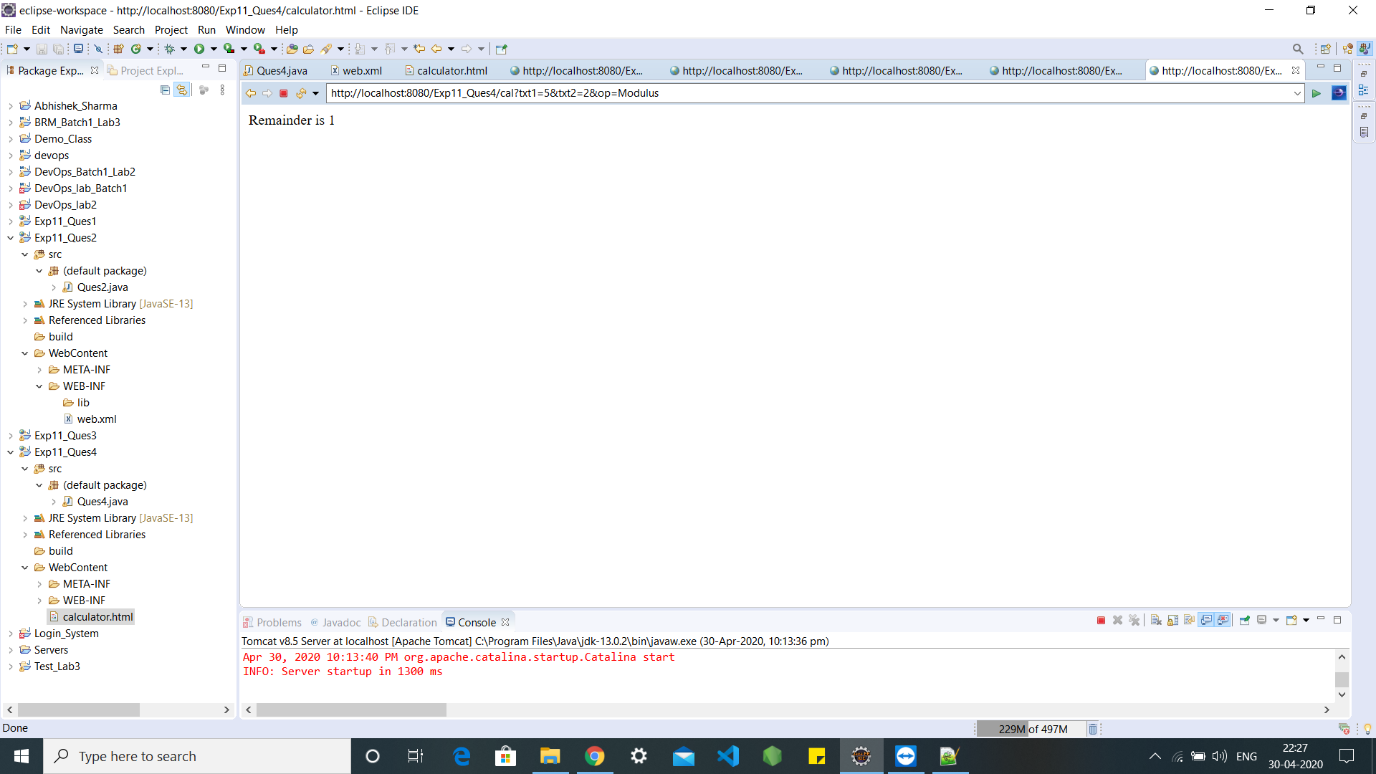
</body>

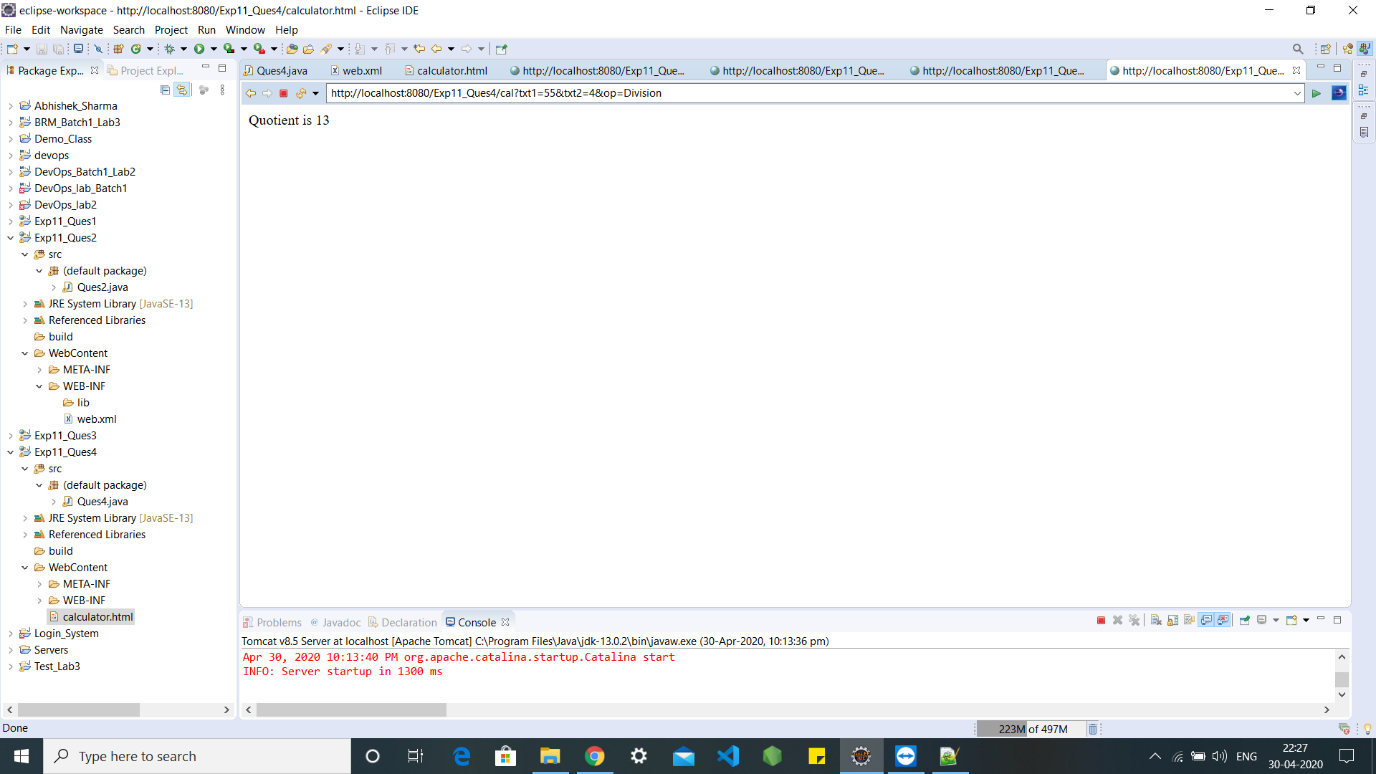
</html>

OUTPUTS: -









**EXPERIMENT – 12**

**TITLE: JSP**

**Project Specification: (Write the following classes in same project named as Proj\_Lab9)**

**Objective: After these lab exercises students will be in position to clear the concept of JSP and how to write the server side scripting language.**

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

<%@ 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"%>

<html>

<head>

<title>Jsp Exp1</title>

</head>

<body>

<sql:setDataSource var = "snapshot" driver = "com.mysql.jdbc.Driver"

url = "jdbc:mysql://localhost:3306/school"

user = "root" password = "redhat"/>

<sql:query dataSource = "${snapshot}" var = "result">

SELECT \* from Student;

</sql:query>

<table border = "1" width = "100%">

<tr>

<th>Student ID</th>

<th>Name</th>

<th>Age</th>

</tr>

<c:forEach var = "row" items = "${result.rows}">

<tr>

<td><c:out value = "${row.id}"/></td>

<td><c:out value = "${row.name}"/></td>

<td><c:out value = "${row.age}"/></td>

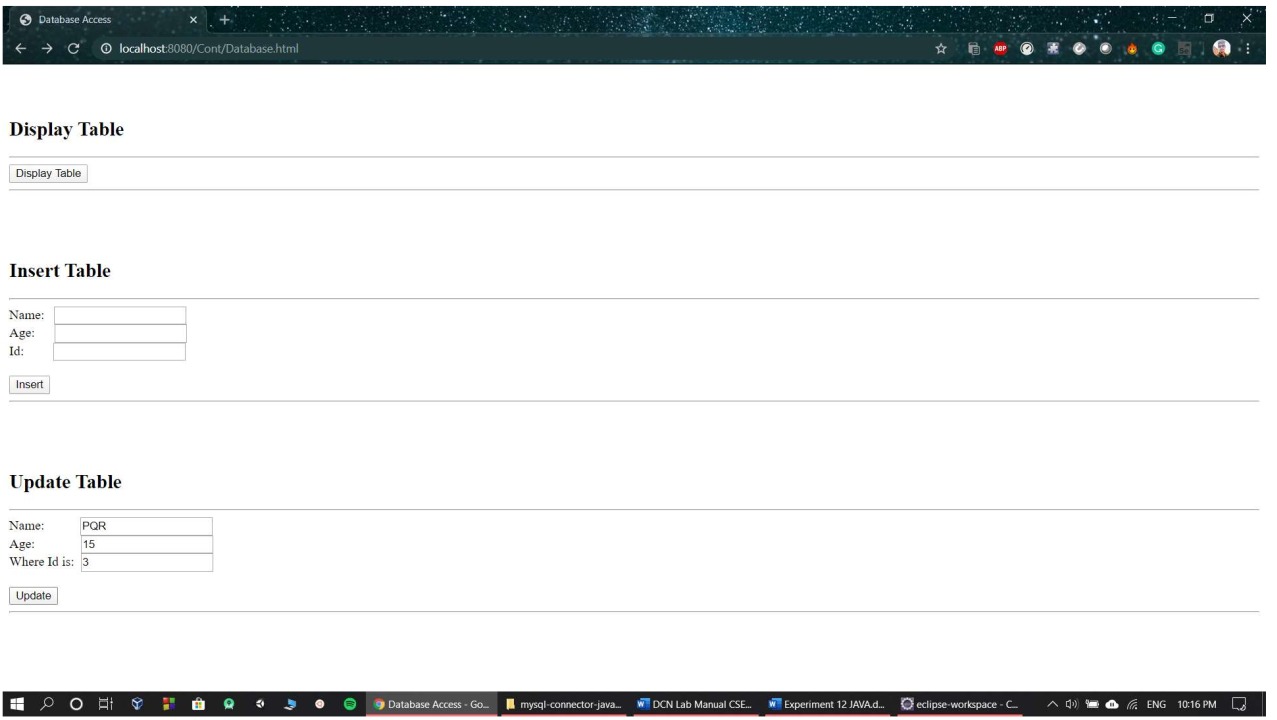
</tr>

</c:forEach>

</table>

</body>

</html>



**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.**

[9:53 PM, 5/5/2020] Abhishek Sharma: <%@ 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"%>

<html>

<head>

<title>Insert Operation JSP</title>

</head>

<body>

<sql:setDataSource var = "snapshot" driver = "com.mysql.jdbc.Driver"

url = "jdbc:mysql://localhost:3306/school"

user = "root" password = "redhat"/>

<sql:update dataSource = "${snapshot}" var = "result">

INSERT INTO Student VALUES (<%= request.getParameter("id")%>, <%= request.getParameter("age")%>, '<%= request.getParameter("name")%>' );

</sql:update>

<sql:query dataS…

[9:53 PM, 5/5/2020] Abhishek Sharma: <%@ 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"%>

<html>

<head>

<title>Insert Operation JSP</title>

</head>

<body>

<sql:setDataSource var = "snapshot" driver = "com.mysql.jdbc.Driver"

url = "jdbc:mysql://localhost:3306/school"

user = "root" password = "redhat"/>

<sql:update dataSource = "${snapshot}" var = "result">

UPDATE Student SET age = <%= request.getParameter("age")%>, name = '<%= request.getParameter("name")%>' WHERE id = <%= request.getParameter("id")%>;

</sql:update>

<sql:query dataSource = "${snapshot}" var = "result">

SELECT \* from Student;

</sql:query>

<table border = "1" width = "100%">

<tr>

<th>Student ID</th>

<th>Name</th>

<th>Age</th>

</tr>

<c:forEach var = "row" items = "${result.rows}">

<tr>

<td><c:out value = "${row.id}"/></td>

<td><c:out value = "${row.name}"/></td>

<td><c:out value = "${row.age}"/></td>

</tr>

</c:forEach>

</table>

</body>

</html>

