

***PRACTICAL FILE***

**NAME: ANIMESH JAIN**

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

**ROLL NO.: R171218022**

**SAP ID: 500069453**

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

**Index**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **Experiment** | **Page NO.** |
| **2,3** | **Basic Java Programming** | **3-10** |
| **4** | **Inheritance** | **11-16** |
| **5** | **Interface** | **17-20** |
| **6** | **Package** | **21-23** |
| **7** | **Exception** | **24-27** |
| **8** | **String Handling and Wrapper Class** | **28-33** |
| **9** | **Threads and Collections** | **34-43** |
| **10** | **JDBC** | **44-47** |
| **11** | **Servlet** | **48-57** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Experiment: 2,3**

**Title: Basic Java Programming**

Q1. Write a program to find the largest of 3 numbers.

Input:

public class BiggestNum {

public static void main(String args[]) {

int num1 = 15 , num2 = 25 , num3 = 41;

if (num1 >= num2 && num1 >= num3)

System.out.println( num1 + " is the maximum number.");

else if (num2 >= num1 && num2 >= num3)

System.out.println( num2 + " is the maximum number.");

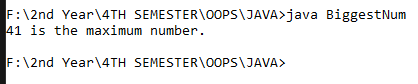
else

System.out.println( num3 + " is the maximum number.");

}

}

Output:



Q2. Write a program to add two number using command line arguments.

Input:

public class Add2Num{

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("Add is:"+c);

}

}

Output:



Q3. Write a program to print Fibonacci series using loop.

Input:

public class Fib{

public static void main(String args[])

{

int n=10, f=0, g=1;

System.out.print("First "+ n +" terms: ");

for (int i=1;i<=n;++i)

{System.out.print(g + " ");

int sum = f + g;

f = g;

g = sum;

}

}

}

Output:



Q4: Write a program to implement a command line calculator.

Input:

import java.util.Scanner;

public class Calc{

public static void main(String args[])

{

int add, sub, mul, div;

char dec ;

Scanner Calc = new Scanner(System.in);

do

{

System.out.println("Choose \n1 add \n2 sub\n 3 mul\n4 div " );

int n = Calc.nextInt();

int a=25, b=5;

add= a+b;

sub=a-b;

mul=a\*b;

div=a/b;

if(n==1)

System.out.println("Addition is "+ add );

else if(n==2)

System.out.println("Subtraction is "+ sub );

else if(n==3)

System.out.println("Multiplication is "+ mul );

else if(n==4)

System.out.println("Division is "+ div );

else

System.out.println("Wrong call please try again!!!" );

System.out.println("if u want to try again press y" );

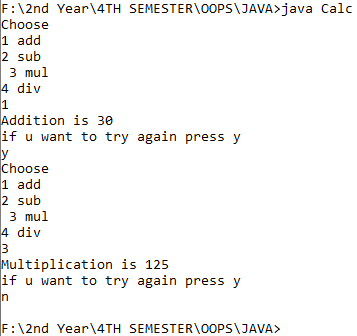
dec = Calc.next().charAt(0);

}while(dec=='y');

}

}

Output:



Q5: Write a program using classes and object in java.

Input:

public class ObjClass{

int x=21 , y=16;

public static void main(String args[])

{

ObjClass obj1 = new ObjClass();

if(obj1.x < obj1.y)

System.out.println(obj1.x + " is lesser than" + obj1.y);

else

System.out.println(obj1.x + "is greater than" + obj1.y);

}}

Output:



Q7: Write a program to accept three digits (i.e. 0 - 9) and print all its possible combinations.

Input:

import java.util.Scanner;

public class PosComb{

public static void main(String args[])

{

Scanner pc = new Scanner(System.in);

int a[] = new int[3] ;

System.out.println("Enter any 3 numbers from o-9");

for (int i=0; i<3; i++)

a[i] = pc.nextInt();

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(a[x] + "" + a[y] + ""+ a[z]);

}

}

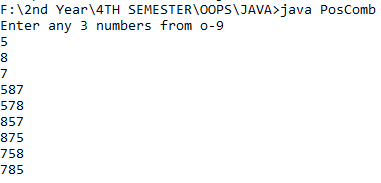
}

}

}

}

Output:



Q8: Write a Java Program to accept 10 numbers in an array and compute the square of each number. Print the sum of these numbers.

Input:

import java.util.Scanner;

public class Squaring

{

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

int a[] = new int[11];

int x[] = new int[11];

int t=0;

System.out.println("Enter 10 values ");

for (int i=1 ; i<=10 ; i++)

{

System.out.println("Enter " + i + " term ");

a[i]= s.nextInt();

}

System.out.println("Square of the Numbers");

for (int k=1 ; k<=10 ; k++)

{

x[k]= a[k]\*a[k];

System.out.println("Square of number " + k);

System.out.println(x[k]);

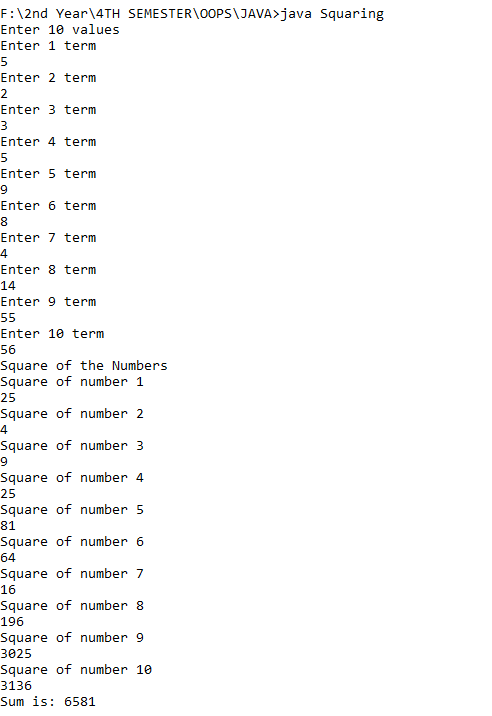
t=t+x[k];

}

System.out.println("Sum is: " + t);

}}

Output:



Q9: Write a program to input a number of a month (1 - 12) and print its equivalent name of

the month.

Input:

import java.util.Scanner;

public class Months{

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

System.out.println("Enter the number t find the month " );

int n = s.nextInt();

System.out.println("The month is "+ n);

if (n==1)

System.out.println("January");

else if(n==2)

System.out.println("February");

else if(n==3)

System.out.println("March");

else if(n==4)

System.out.println("April");

else if(n==5)

System.out.println("May");

else if(n==6)

System.out.println("June");

else if(n==7)

System.out.println("July");

else if(n==8)

System.out.println("August");

else if(n==9)

System.out.println("September");

else if(n==10)

System.out.println("October");

else if(n==11)

System.out.println("November");

else if(n==12)

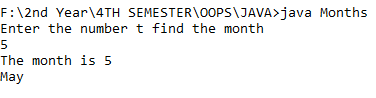
System.out.println("December");

else

System.out.println("There are only 12 months :)");

}}

Output:



Q10: Write a program to find the sum of all integers greater than 40 and less than 250 that are divisible by 5.

Input:

public class SumOfInt{

public static void main(String args[])

{

int x;

int sum=0;

System.out.println("All integers which are greater than 40 and less than 250 that are divisible by 5 are :-");

for (x=40;x<=250;x++)

{

if (x%5==0){

System.out.println(" "+ x);

sum = sum+x;

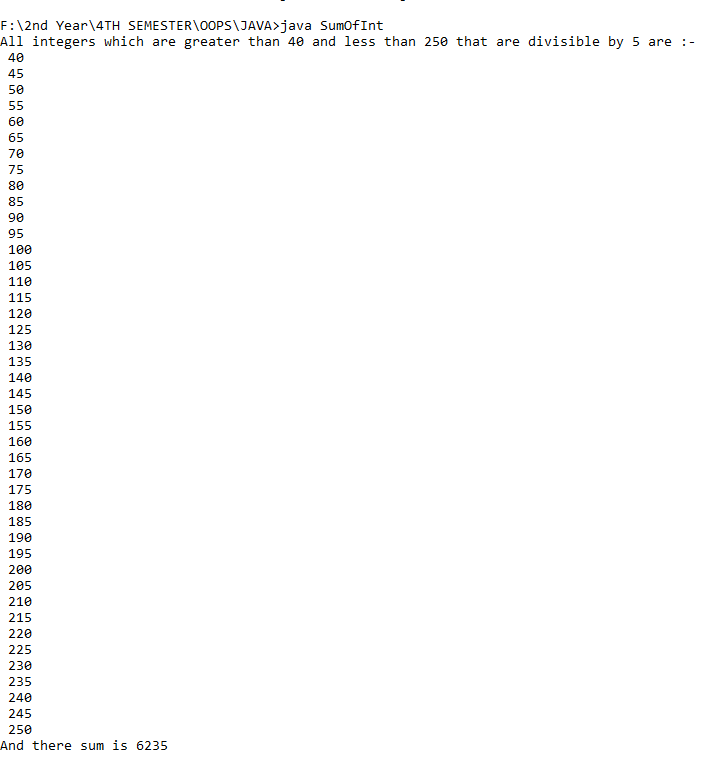
}

}

System.out.println("And there sum is "+ sum);

}}

Output:



**Experiment 4**

**Inheritance**

Q1: Write a Java program to show that private member of a super class cannot be accessed from derived classes.

Input:

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);

}

}

class PrivateClass

{

public static void main(String args[])

{

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

int a1=cr.area();

int v1=cr.volume();

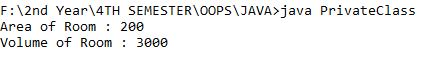
System.out.println("Area of Room : "+a1);

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

}

}

Output:



Q2: Write a program in Java to create a Player class. Inherit the classes Cricket \_Player, Football \_Player and Hockey\_ Player from Player class.

Input:

class sports

{

String name;

int age;

sports(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 cricketer extends sports

{

String type;

cricketer(String n, String t,int a)

{

super(n,a);

type=t;

}

public void show()

{

super.show();

System.out.println("Player plays : "+type);

}

}

class football\_player extends sports

{

String type;

football\_player(String n, String t,int a)

{

super(n,a);

type=t;

}

public void show()

{

super.show();

System.out.println("Player plays : "+type);

}

}

class hockey\_player extends sports

{

String type;

hockey\_player(String n, String t,int a)

{

super(n,a);

type=t;

}

public void show()

{

super.show();

System.out.println("Player plays : "+type);

}

}

// main

class Player

{

public static void main(String args[])

{

cricketer c1 = new cricketer("Virat","Cricket",30);

football\_player c2 = new football\_player("Mbappe","Football",20);

hockey\_player c3 = new hockey\_player("Salah","Hockey",25);

c1.show();

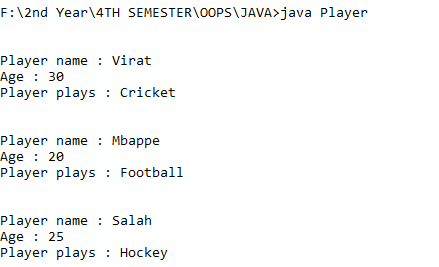
c2.show();

c3.show();

}

}

Output:



Q3: 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.

Input:

class worker

{

String name="bajaj";

int s\_rate= 150;

}

class daily\_worker extends worker

{

void compay(int hours)

{

System.out.println(s\_rate\*hours);

}

}

class salaried\_worker extends worker

{

void compay(int hours)

{

System.out.println(s\_rate\*40);

}

}

class Work

{

public static void main(String args[])

{

daily\_worker k= new daily\_worker();

k.compay(48);

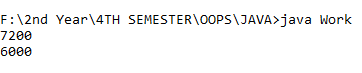
salaried\_worker j=new salaried\_worker();

j.compay(48);

}

}

Output:



Q4: 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.

Input:

Output:

Q5: 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.

Input:

import java.util.\*;

class employee

{

String name;

int empid;

int salary;

employee()

{

name="Animesh";

empid=22;

salary=9000;

}

employee(String n,int e,int s)

{

name=n;

empid=e;

salary=s;

}

String no()

{

return name;

}

int fo()

{

return salary;

}

double increaseSalary(double per\_in)

{

double in=(salary\*(per\_in/100));

double s=salary+in;

return s;

}

}

class manager extends employee

{

public String department="d";

}

class Employee

{

public static void main(String args[])

{

Scanner sc= new Scanner(System.in);

System.out.println("enter the name");

String n=sc.next();

System.out.println("enter the employee id");

int e=sc.nextInt();

System.out.println("enter the salary");

int s=sc.nextInt();

System.out.println("enter the percentage\_rate increaced in salary");

double p=sc.nextDouble();

employee m=new employee(n,e,s);

System.out.println(m.no());

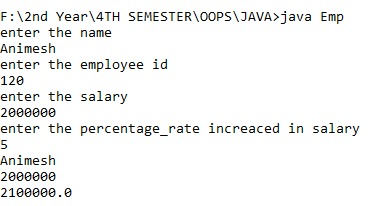
System.out.println(m.fo());

System.out.println(m.increaseSalary(p));

}

}

Output:



**Experiment 5**

**Interface**

Q1: 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.

Input:

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 Interface

{

public static void main(String [] args)

{

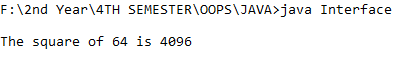
ToTestInt x= new ToTestInt();

System.out.println("\nThe square of 64 is "+x.return\_ans(64));

}

}

Output;



Q2: 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.

Input:

interface A {

void meth1();

void meth2();

}

interface B extends A {

void meth2(); //Overriding the method in A ??

}

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 meth3().");

}

}

class InterfaceAndClass {

public static void main(String arg[]) {

B ob = new MyClass(); // Interface refrence variable

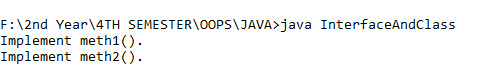
ob.meth1();

ob.meth2();

}

}

Output:



Q3: Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program.

Input:

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 sub

{

public static void main(String args[])

{ stud s=new stud();

s.name("AwesomeAni9");

s.division(10);

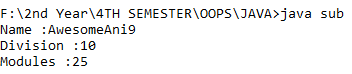
s.modules(25);

s.disp();

}

}

Output:



Q4: Write a program to create an Interface having two methods division and modules. Create a class, which overrides these method.

Input:

interface ar

{

final double pi=3.14;

double getarea(double r);

}

class circle implements ar

{

public double getarea(double r){

double area=2\*pi\*r;

return area;}

}

public class AreaCalc

{

public static void main(String args[])

{

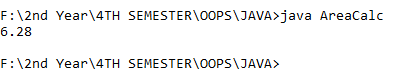
circle c1= new circle();

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

}

}

Output:



**Experiment 6**

**Pacakage**

Q1: Write a Java program to implement the concept of importing classes from user defined package and created packages.

Input:

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();

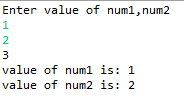
System.out.println("value of num1 is: " + obj.num1);

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

}

}

Output:



Q2: 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.

Input:

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 {

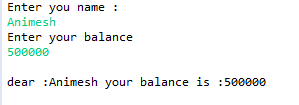
public static void main(String[] args) {

pac obj = new pac();

}

}

Output:



**Experiment 7**

**Exceptions**

Q1: 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.

Input:

import java.lang.ArrayIndexOutOfBoundsException;

import java.util.Arrays;

import java.util.Scanner;

class exp7 {

String[] name = new String[5];

int[] rollno = new int[5];

}

public class Exception1 {

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

exp7 obj = new exp7();

for (int i = 0; i < obj.rollno.length; i++) {

try { // identification of error in try block

Scanner in = new Scanner(System.in);

System.out.println("Enter the name of the student");

obj.name[i] = in.nextLine();

System.out.println("Enter the roll no of the student");

obj.rollno[i] = in.nextInt();

}

catch (ArrayIndexOutOfBoundsException e) // catching the errors

{

System.out.println("index is exceeding");

}

}

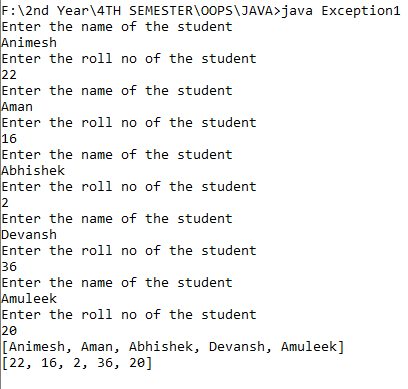
System.out.println(Arrays.toString(obj.name));

System.out.println(Arrays.toString(obj.rollno));

}

}

Output:



Q2: Write a Java program to enable the user to handle any chance of divide by zero exception.

Input:

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) {

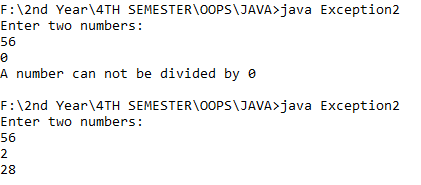
demoexception obj = new demoexception();

obj.dividefunction();

}

}

Output:



Q3: Create an exception class, which throws an exception if operand is nonnumeric in calculating modules.

Input:

Output:

Q4: 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.

Input:

import java.util.\*;

class direction extends Exception{

direction(){

super("collision");

}

}

public class Exception4{

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);

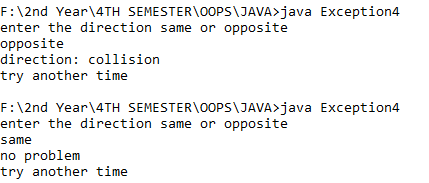
}

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

}

}

Output:



**Experiment 8**

**String Handling and Wrapper Class**

Q1: 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.

Input:

class Occurence

{

public static void main (String[] args)

{

String str = "Hey! This is Animesh Jain.";

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);

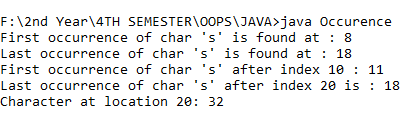
int char\_at = str.charAt(20);

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

}

}

Output:



Q2: Write a program that converts all characters of a string in capital letters. (Use StringBuffer to store a string). Don’t use inbuilt function.

Input:

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");

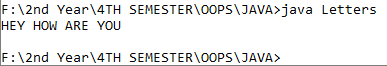
convertOpposite(str);

System.out.println(str);

}

}

Output:



Q3: Write a program in Java to read a statement from console, convert it into upper case and again print on console.

Input:

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:");

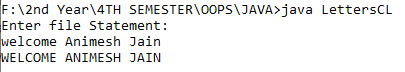
String s1=in.readLine();

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

}

}

Output:



Q4: 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.

Input:

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("Animesh Jain");

d1.disp();

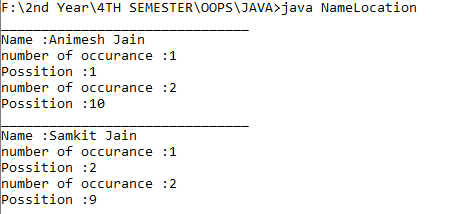
data d2=new data("Samkit Jain");

d2.disp();

}

}

Output:



**Wrapper Class**

Q1: 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.

Input:

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);

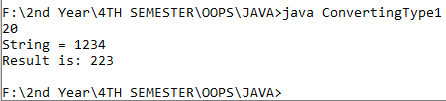
int sum = inum + inum2;

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

}

}

Output:



Q2: 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.

Input:

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);

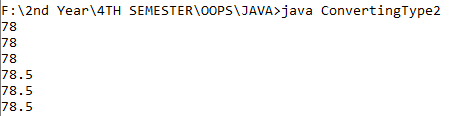
String si = fObj.toString();

System.out.println(si);

}

}

Output:



**Experiment 9**

**Threads**

Q1: Write a program to implement the concept of threading by extending Thread Class and Runnable interface.

Input:

class test

{

public static void t1()

{

System.out.println("In test class ");

}

}

class ExtThreadAndImmpeInter extends test implements Runnable

{

public void run()

{

System.out.println("Child method to be executed by child thread");

}

public static void main(String [] args)

{

ExtThreadAndImmpeInter a = new ExtThreadAndImmpeInter();

a.t1();

Thread t2 = new Thread(a);

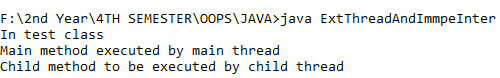
t2.start();

System.out.println("Main method executed by main thread");

}

}

Output:



Q2: Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.

Input:

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 OddEvenInThreads

{

public static void main(String arg[])

{

even e=new even();

odd o=new odd();

}

}

Output:



Q3: Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization.

Input:

class IncrementCounterVariable

{

public static void main(String args[]) 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=sync.increment(d); }

}

class sync

{

synchronized static item increment(item i)

{

i.count++;

return(i);

}

}

Output:



Q4: 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

Input:

class que implements Runnable

{

public synchronized void run()

{

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

System.out.println("Threading is done" + Thread.currentThread().getPriority());

}

}

public class PriorityInThreads {

public static void main(String[] args) {

que m = new que();

Thread t1 = new Thread(m);

Thread t2 = new Thread(m);

Thread t3 = new Thread(m);

Thread t4 = new Thread(m);

Thread t5 = new Thread(m);

t1.start();

t1.setPriority(5); // setting priority

t2.start();

t2.setPriority(10);

t3.start();

t3.setPriority(2);

t4.start();

t4.setPriority(3);

t5.start();

t5.setPriority(8);

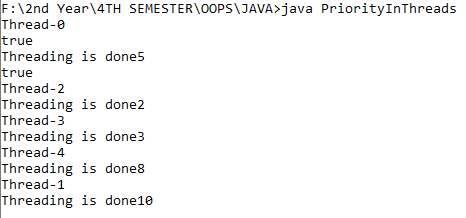
System.out.println(t1.isAlive()); // checking whether the thread is alive or not

System.out.println(t2.isAlive());

}

}

Output:



**Collections**

Q1: Write a program for the following:

Read all elements from ArrayList by using Iterator.

Create duplicate object of an ArrayList instance.

Reverse ArrayList content.

Input:

import java.util.ArrayList;

import java.util.Collection;

import java.util.Collections;

import java.util.Iterator;

import java.util.List;

class JavaCollection

{

public static void main(String args[])

{

List names = new ArrayList<String>();

names.add("Animesh");

names.add("Ani");

names.add("Kush");

names.add("Roman");

List Dnames = new ArrayList<String>();

Dnames = names;

Collections.reverse(names);

Iterator IDnames = Dnames.iterator();

System.out.println("Actual list OR Duplicated list :");

while(IDnames.hasNext()){

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

}

System.out.println();

Iterator Inames = names.iterator();

System.out.println("Reversed List :");

while(Inames.hasNext()){

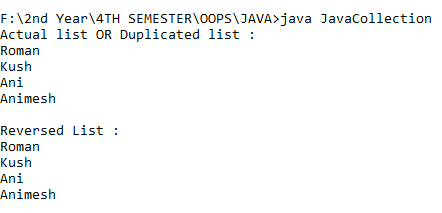
System.out.println(Inames.next());

}

}

}

Output:



Q2: 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

Input:

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

class NewHashMap

{

public static void main(String args[])

{

HashMap<String, Integer> map = new HashMap<>();

map.put("Animesh", 10);

map.put("Ani", 30);

map.put("Roman", 20);

map.put("Kush", 20);

System.out.println();

String key = "Animesh";

if (map.containsKey(key))

{

Integer a = map.get(key);

System.out.println("value for key "+ key +" is: "+a);

}

int value = 20;

System.out.println();

if (map.containsValue(value))

{

Iterator it = map.entrySet().iterator();

while(it.hasNext()){

String str = it.next().toString();

if(str.split("=")[1].contains(String.valueOf(value)))

{

System.out.println("key for value "+ value +" is: "+str.split("=")[0]);

}

}

}

System.out.println();

System.out.println("Printing as key set :");

Iterator it = map.keySet().iterator();

while(it.hasNext())

{

System.out.println(it.next().toString());

}

System.out.println();

System.out.println("Printing as entry set :");

it = map.entrySet().iterator();

while(it.hasNext())

{

System.out.println(it.next().toString());

}

if (map.isEmpty())

{

System.out.println("map is empty");

}

else

{

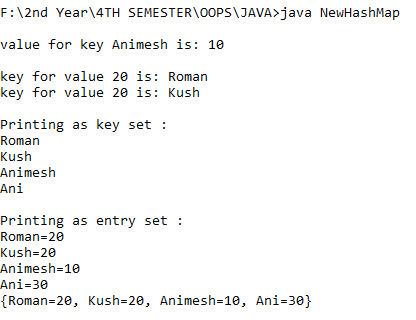
System.out.println(map);

}

}

}

Output:



Q3: 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

Input:

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

class NewHashSet{

public static void main(String agr[]){

List names = new ArrayList<String>();

names.add("Animesh");

names.add("Ani");

names.add("Kush");

names.add("Roman");

names.add("Chirag");

HashSet hNames = new HashSet<>(names);

System.out.println("Initial list :"+hNames);

String key = "Ani";

System.out.println("Contains key '" + key+ "' ? :" + hNames.contains(key));

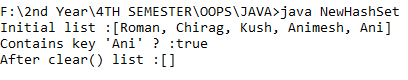
hNames.clear();

System.out.println("After clear() list :"+hNames);

}

}

Output:



**Experiment 10**

**JDBC**

Q1: 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.

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)

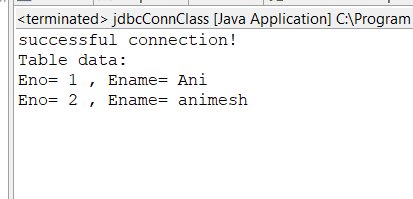
{System.out.println(e);}

}

}

}

Output:



Q2: 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)

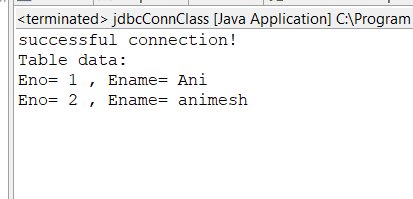
{System.out.println(e);}

}

}

}

Output:



**Experiment 11**

**Servlet**

Q1: Servlet: a) ServletContext interface b)getParameterValues( ) of Servlet Request

Input:

File Name: Ques1

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();

}

}

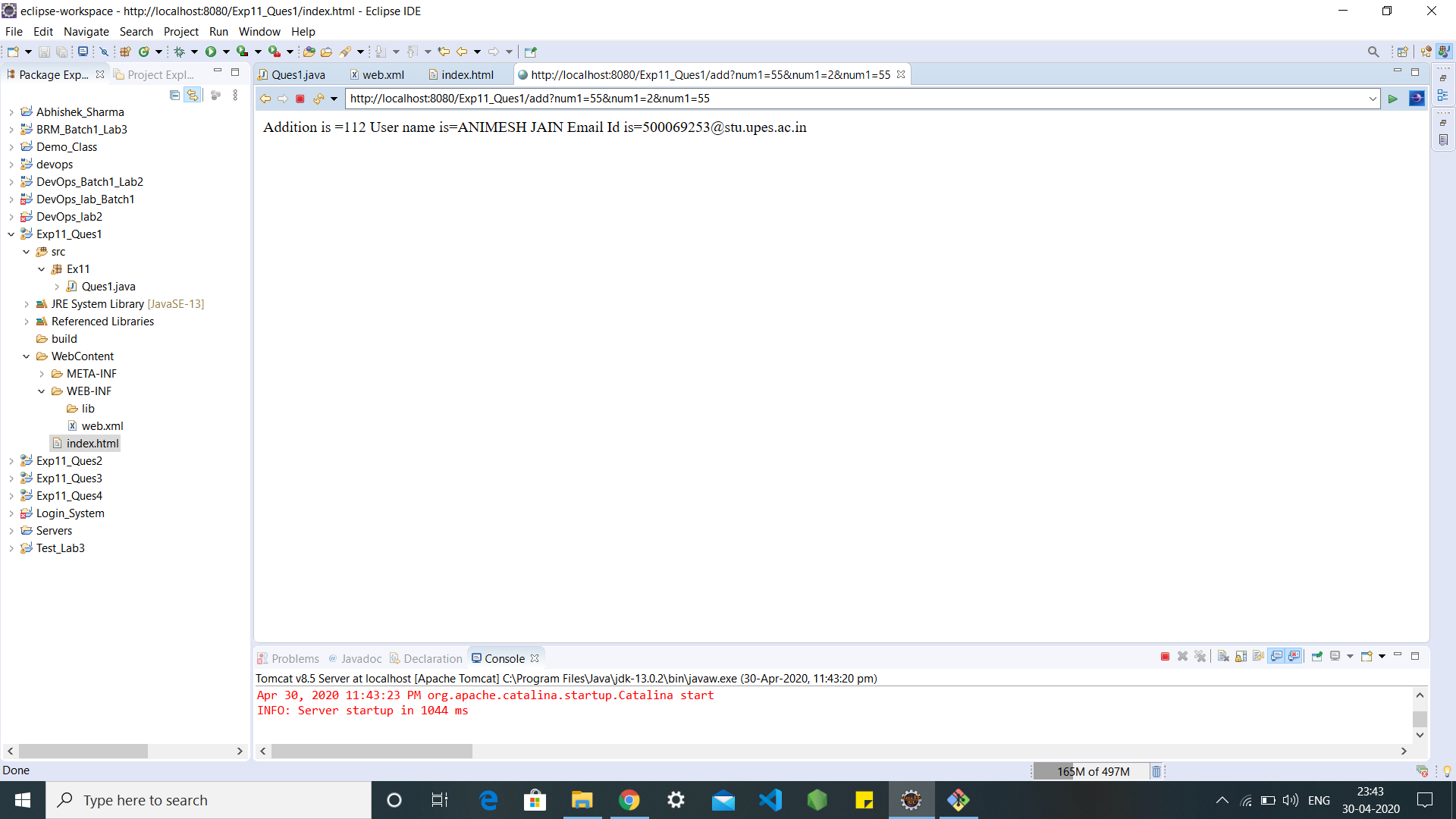
File Name: index.html

|  |
| --- |
| <!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> |

File Name: web.xml

|  |
| --- |
| <?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>Animesh Jain</param-value> |
|  | </context-param> |
|  | <context-param> |
|  | <param-name>email</param-name> |
|  | <param-value>500069453@stu.upes.ac.in</param-value> |
|  | </context-param> |
|  |  |
|  |  |
|  | </web-app> |

Output:



Q2: Write a Servletpage to display current date of the server.

Input:

File Name: Ques2

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>");

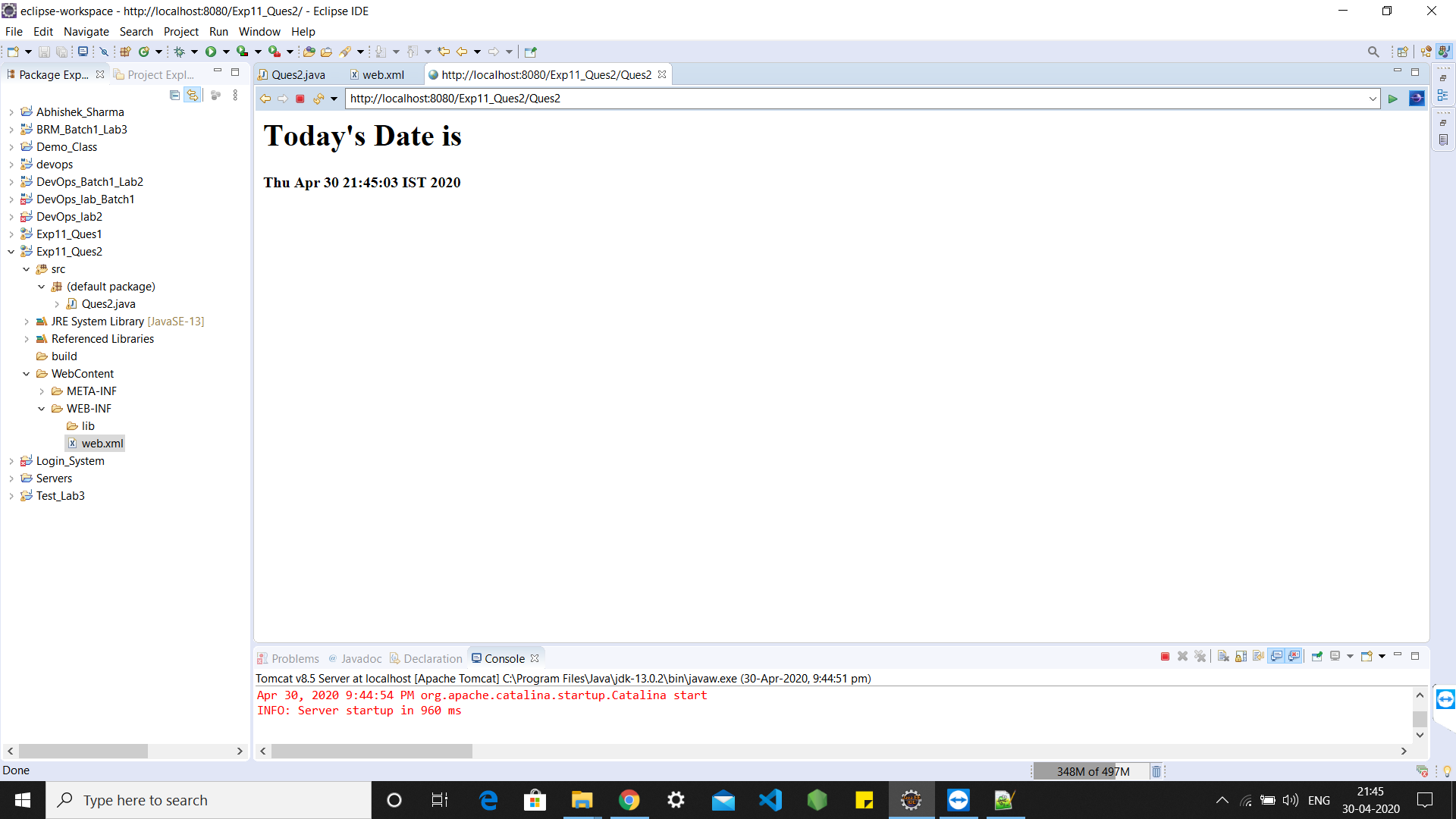
}

}

File Name: web.xml

|  |
| --- |
| <?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:



Q3: Write a Servletpage to which include the two other Servlet page through of include directives feature provided in Servlet.

Input:

File Name: squareclass.java

package newpackage;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class squareclass extends HttpServlet {

public void service(HttpServletRequest req,HttpServletResponse res) throws IOException

{

int k=0;

Cookie cookies[]=req.getCookies(); //recieving cookies from first servlet

for(Cookie c:cookies)

{

if(c.getName().equals("k"))

{

k=Integer.parseInt(c.getValue());

}

}

k=k\*k;

PrintWriter obj=res.getWriter();

obj.println("Square is: "+ k);

}

}

File Name: sendRedirectmethod.java

package newpackage;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class sendRedirectmethod extends HttpServlet {

public void service(HttpServletRequest req,HttpServletResponse res) throws IOException

{

int i=Integer.parseInt(req.getParameter("num1"));

int j=Integer.parseInt(req.getParameter("num2"));

int k=i+j;

PrintWriter obj=res.getWriter();

obj.println("Sum is: "+ k);

Cookie cookie=new Cookie("k",k+""); /\*making new cookie so that value can be passed through one servlet to another\*/

res.addCookie(cookie);

res.sendRedirect("sq"); //calling square servlet

}

}

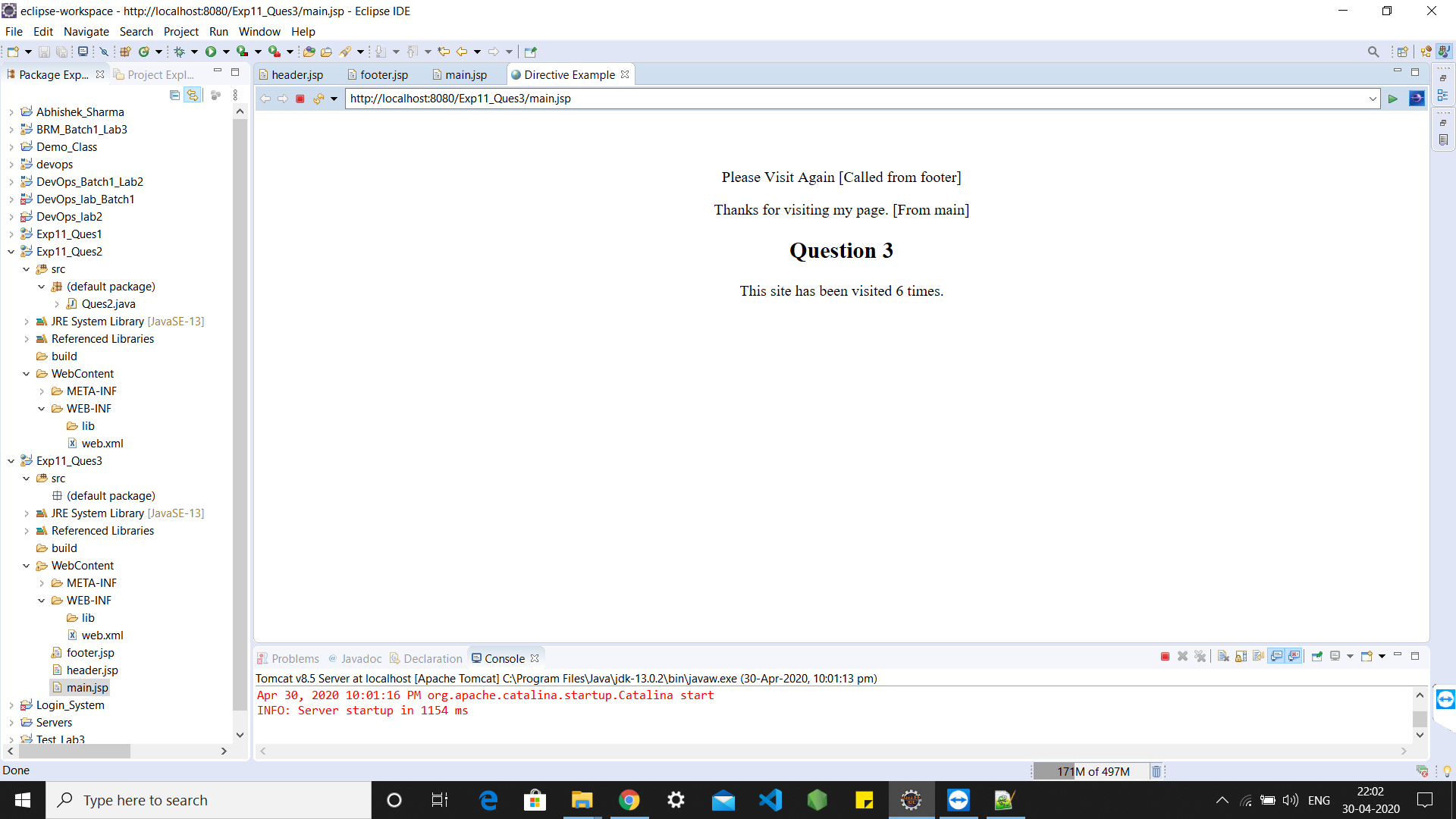
File Name: index.html

|  |
| --- |
| <html> |
|  | <head> |
|  | <title>TODO supply a title</title> |
|  | <meta charset="UTF-8"> |
|  | <meta name="viewport" content="width=device-width, initial-scale=1.0"> |
|  | </head> |
|  | <body> |
|  | <form action="servlet1" method="get"> |
|  | number1: <input type="text" name="no1"> <br> <br> |
|  | number2: <input type="text" name="no2"> <br> <br> |
|  | <input type="submit" name="btnadd" value="Add"> |
|  | </form> |
|  | </body> |
|  | </html> |

File Name: web.html

|  |
| --- |
| <?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" id="WebApp\_ID" version="4.0"> |
|  | <servlet> |
|  | <servlet-name>httpsession</servlet-name> |
|  | <servlet-class>newpackage.sendRedirectmethod</servlet-class> |
|  | </servlet> |
|  | <servlet-mapping> |
|  | <servlet-name>httpsession</servlet-name> |
|  | <url-pattern>/add</url-pattern> |
|  | </servlet-mapping> |
|  | <servlet> |
|  | <servlet-name>session</servlet-name> |
|  | <servlet-class>newpackage.squareclass</servlet-class> |
|  | </servlet> |
|  | <servlet-mapping> |
|  | <servlet-name>session</servlet-name> |
|  | <url-pattern>/sq</url-pattern> |
|  | </servlet-mapping> |
|  | </web-app> |

Output:



Q4: Write a Servletpage to create a simple calculator.

Input:

File Name: Ques4.java

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class 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)));

}

}

}

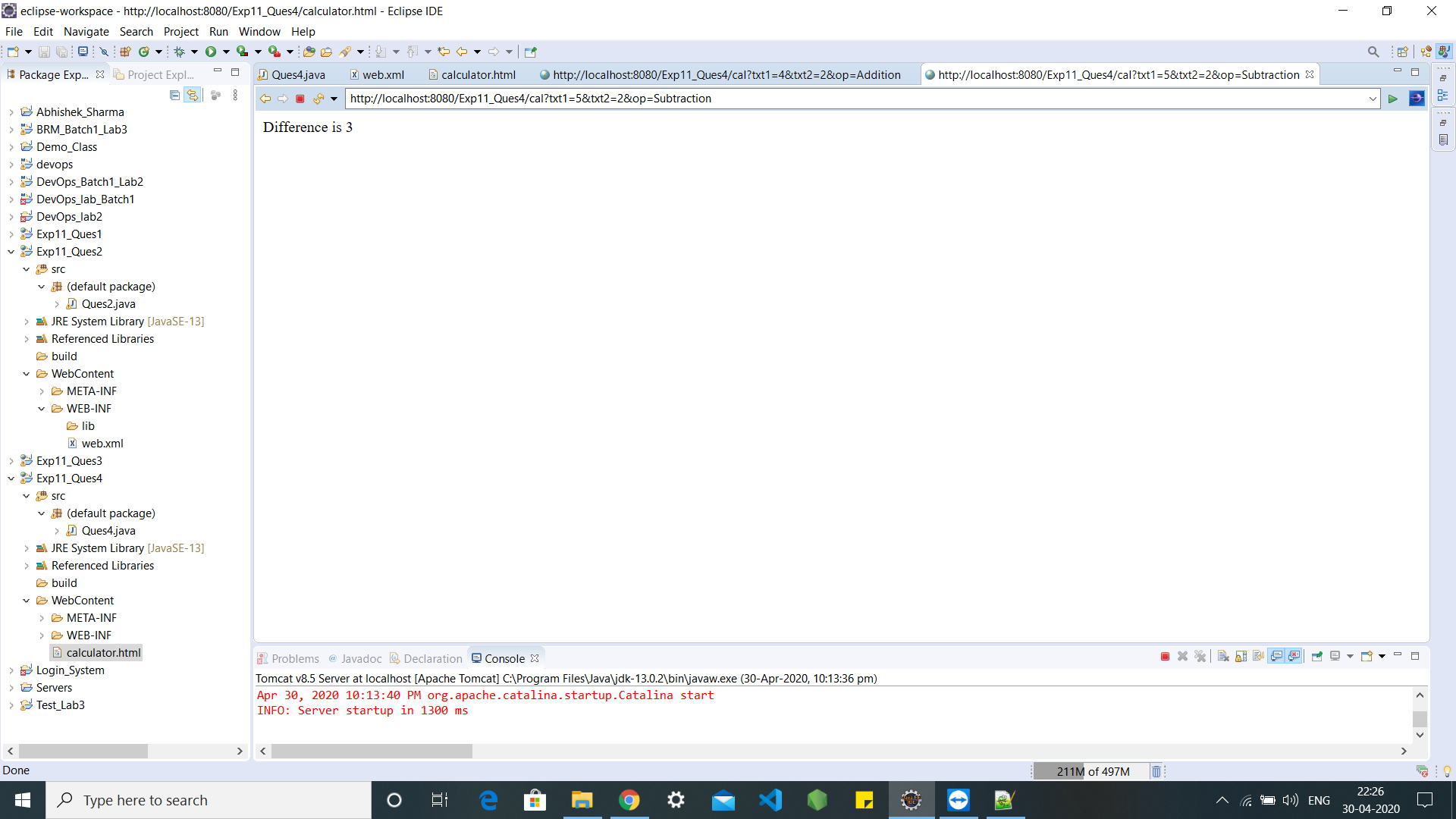
File Name: index.html

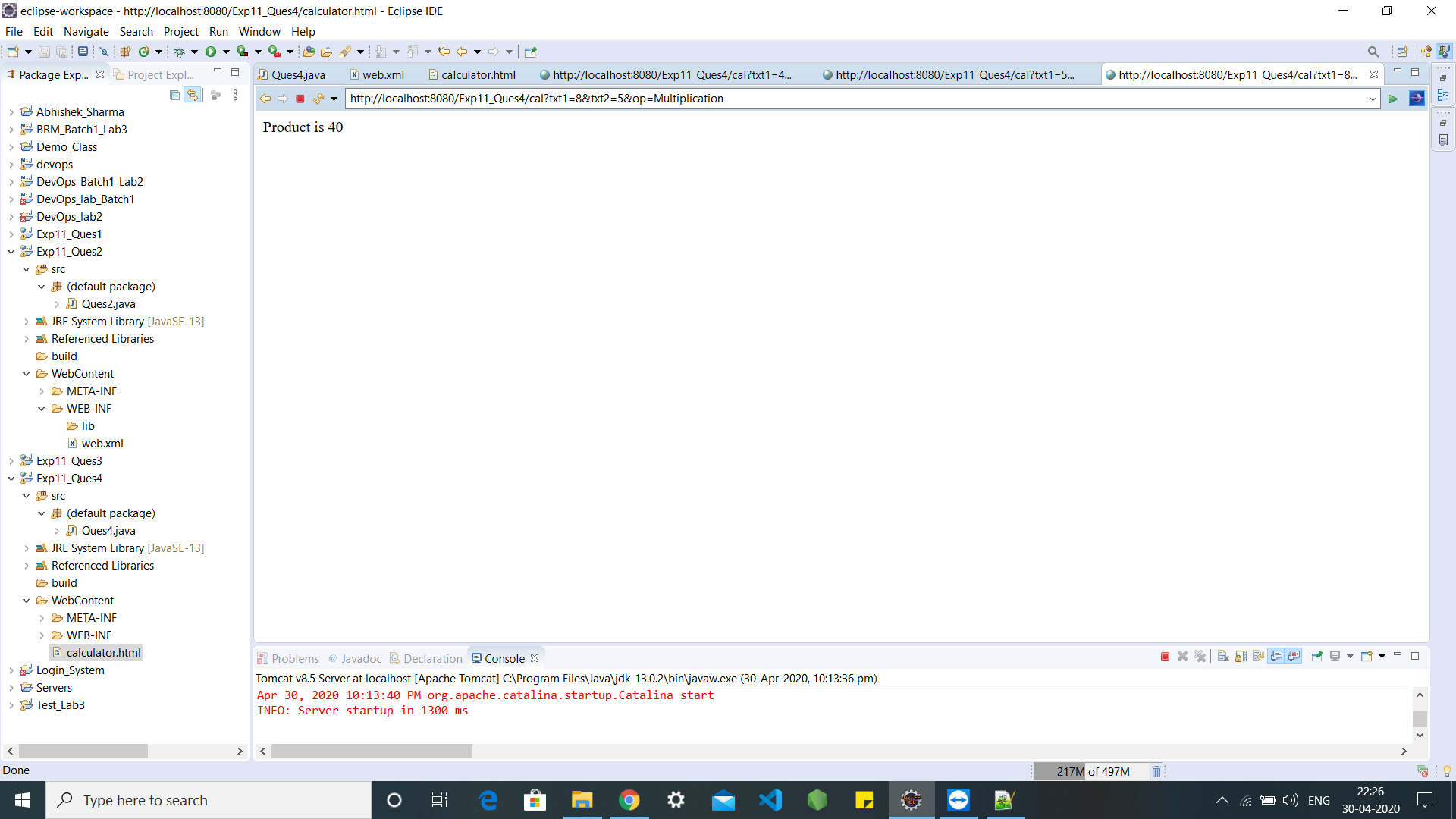
|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <meta charset="ISO-8859-1"> |
|  | <title>Simple Calculator</title> |
|  | </head> |
|  | <body> |
|  | <form action="cal" method="get"> |
|  |  |
|  | Enter num1: |
|  | <input name="txt1" type="text" /> |
|  |  |
|  |  |
|  | Enter num2: |
|  | <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> |

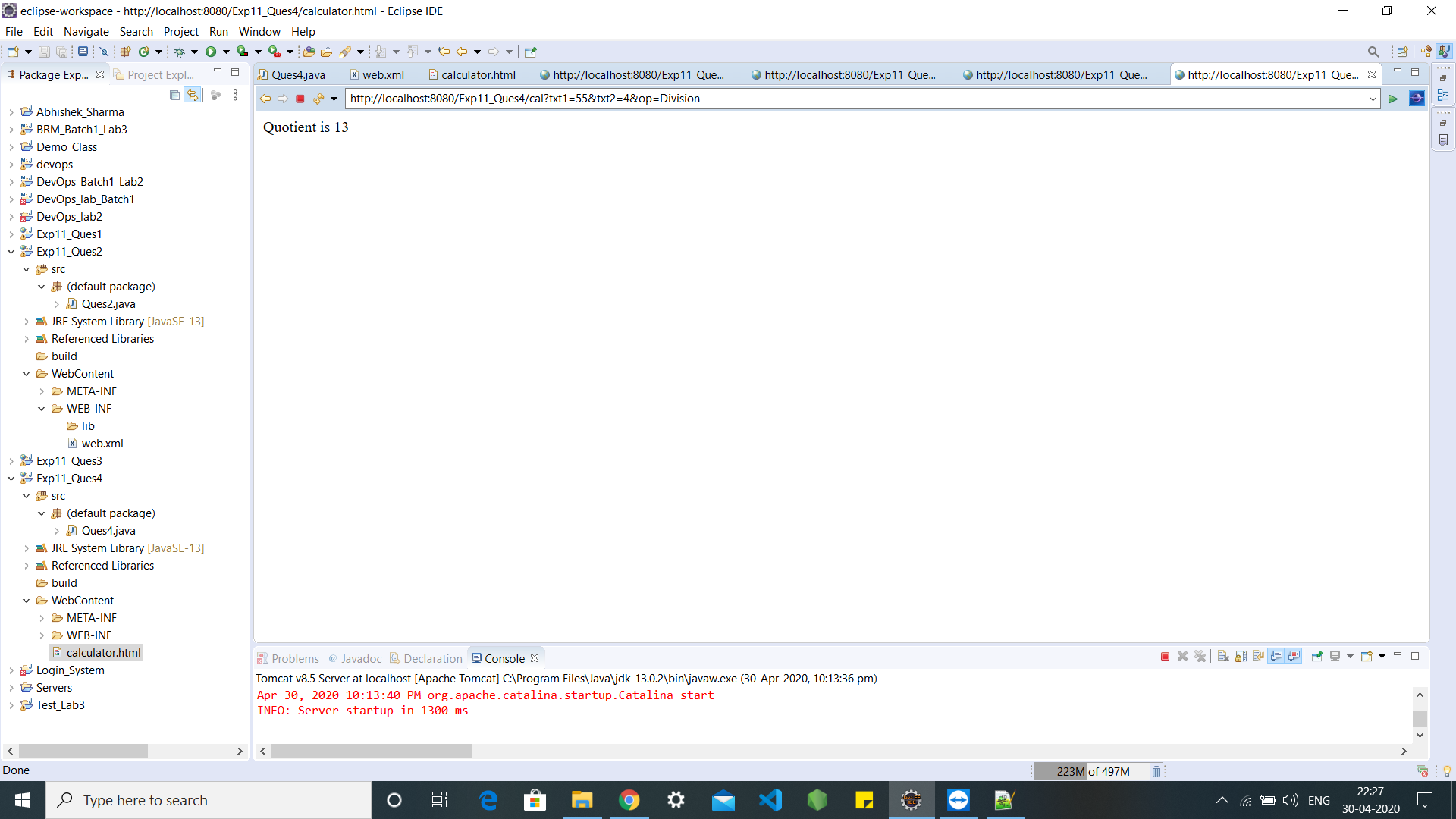
File Name: web.html

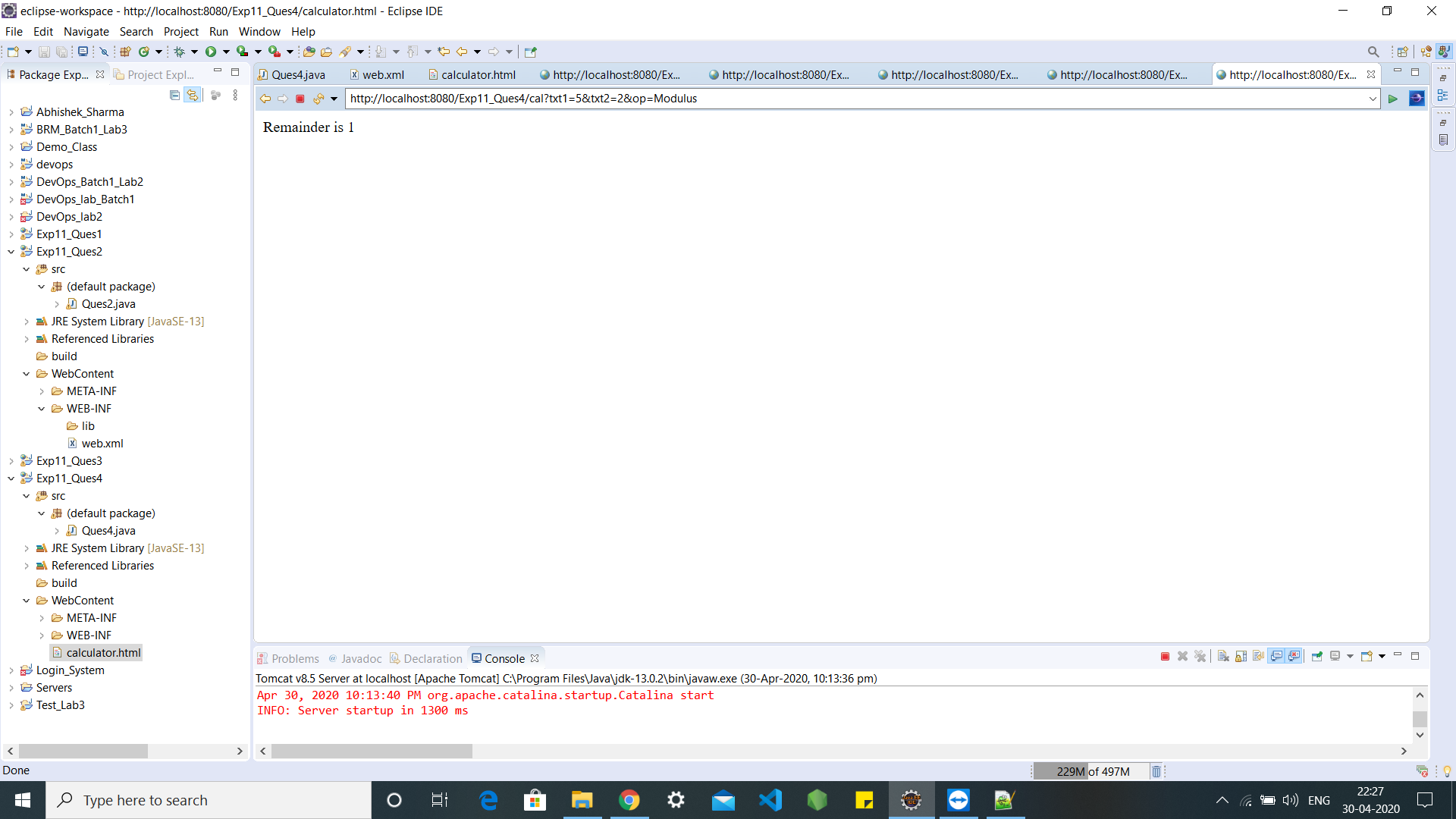
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> |
|  | <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" version="4.0"> |
|  | <display-name>Experiment11</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> |

Output:









**Thank You**