

Object Oriented Programming

Subject code : 142

Name- Ananya Upadhyay Sap ID- 500096803

Roll No.- R2142211181

Course Semester Batch- CSE lll (AI/ML) B5

Submission to -Mr Piyush Chauhan

Assistant Professor

School of Computer Science, UPES

INDEX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S.No. | Title of experime | | | nt | | Date | Page No. |
| 1 |  | | Introduction to Java |  | | 08/08/22 | 3 |
|  | Programming-Installation | | Process |  |
| 2 | Basic Java Programmi | | | ng | | 22/08/22 | 4 |
| 3 | Basic Java Programmi | | | ng | | 29/08/22 | 17 |
| 4 | Inheritence | | |  | | 05/09/22 | 26 |
| 5 | Packages | | |  | | 12/09/22 | 42 |
| 6 | Interfaces | | |  | | 19/09/22 | 49 |
| 7 | Exception | | |  | | 26/09/22 | 61 |
| 8 | String Handling and wra class | | | pper | | 10/10/22 | 69 |
| 9 | Thread | | |  | | 17/10/22 | 78 |
| 10 | Collection | | |  | | 24/10/22 | 88 |
| 11 | JDBC | | |  | | 07/11/22 | 94 |
| 12 | Servlets | | |  | | 14/11/22 | 99 |

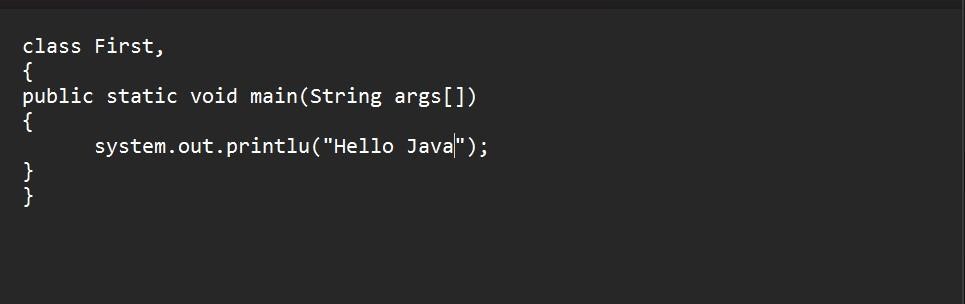
Experiment-1

TITLE: Introduction to Java Environment

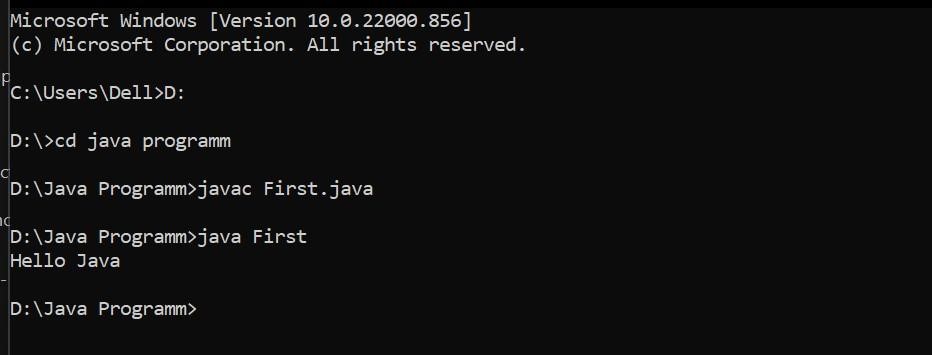
1. Java Versions
2. JDK and JRE
3. Setting Path
4. Code Editors

|  |  |
| --- | --- |
| 5. | Sample Hello World Program. |

Code:



Output:



# Experiment-2

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

**Code:**

import java.util.\*; class LargestNumber{ public static void main(String args[])

{

int a,b,c;

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

Scanner sc = new Scanner(System.in); //input 1st no. a = sc.nextInt(); //assigning value to a System.out.println("Enter the second number = "); //Scanner sc = new Scanner(System.in); //input 2st no. b

= sc.nextInt(); //assigning value to b

System.out.println("Enter the third number = ");

//Scanner sc = new Scanner(System.in); //input 3rd no. c = sc.nextInt(); //assigning value to c

if (a>b && a>c)

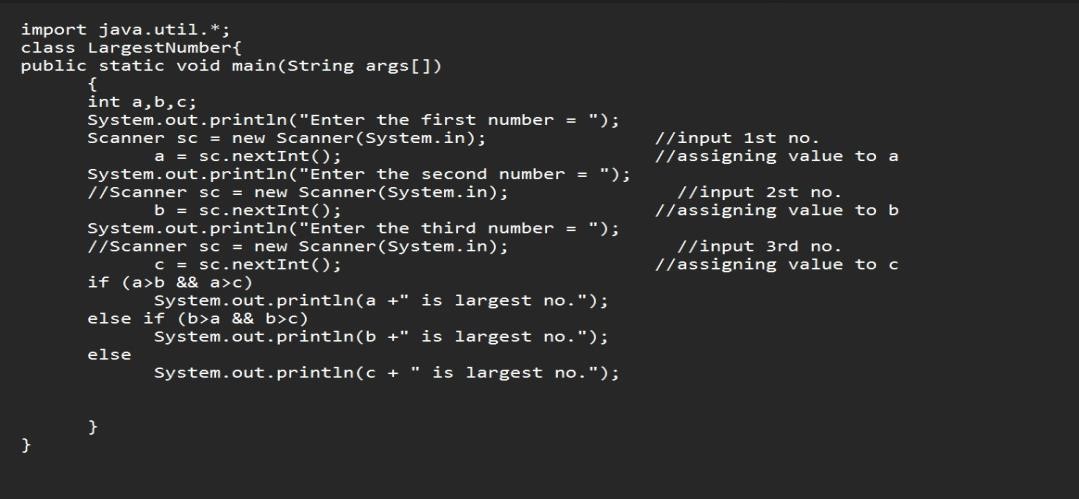
System.out.println(a +" is largest no."); else if (b>a && b>c)

System.out.println(b +" is largest no."); else

System.out.println(c + " is largest no.");

}

}



**Output:**



**2. Write a program to implement command line calculation.**

**Code:**

class Calc{ float result=0; void add(int a , int b )

{

result = a + b;

System.out.println("Sum = " + result);

}

void sub(int a , int b )

{

result = a - b;

System.out.println("Difference = " + result);

}

void mul(int a , int b )

{

result = a \* b;

System.out.println("Multiplication = " + result);

}

void div(int a , int b )

{

result = a / b;

System.out.println("Quotient = " + result);

}

public static void main(String args[])

{

Calc c = new Calc(); int a =

Integer.parseInt(args[0]); int b = Integer.parseInt(args[2]); char ch = args[1].charAt(0);

if (ch == '+')

c.add(a , b ); if (ch == '-')

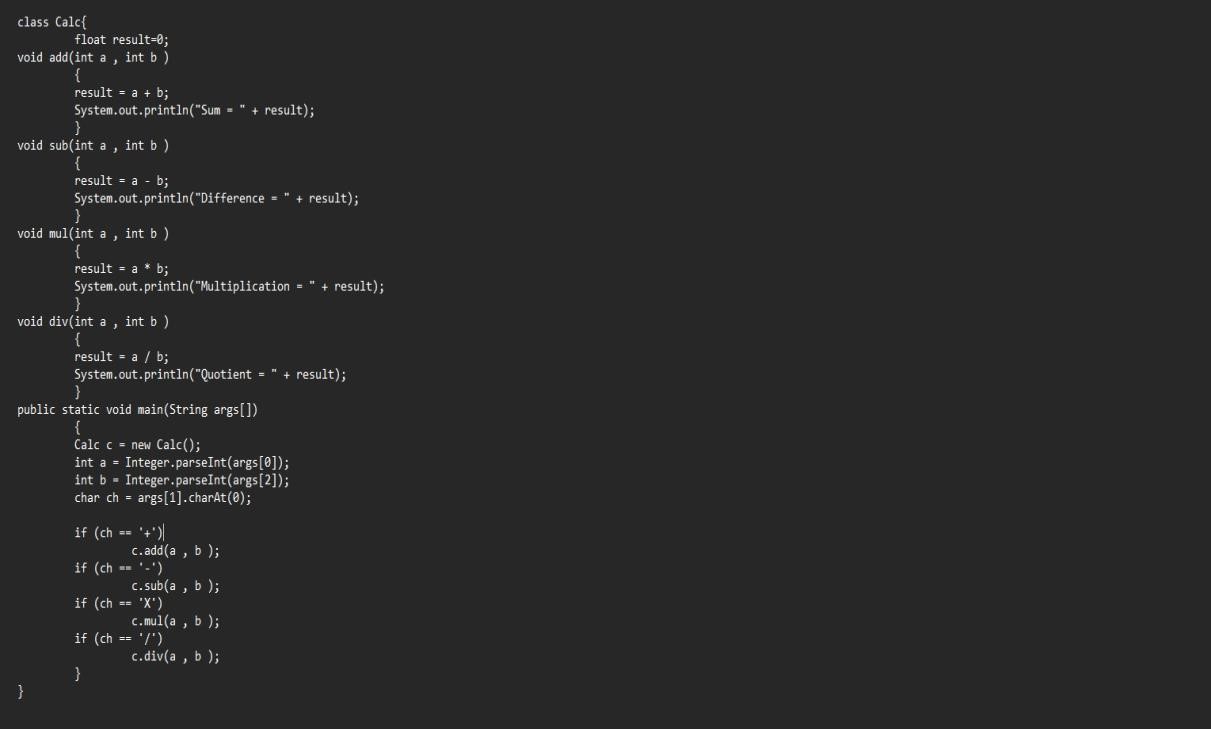
c.sub(a , b ); if (ch == 'X')

c.mul(a , b ); if (ch == '/')

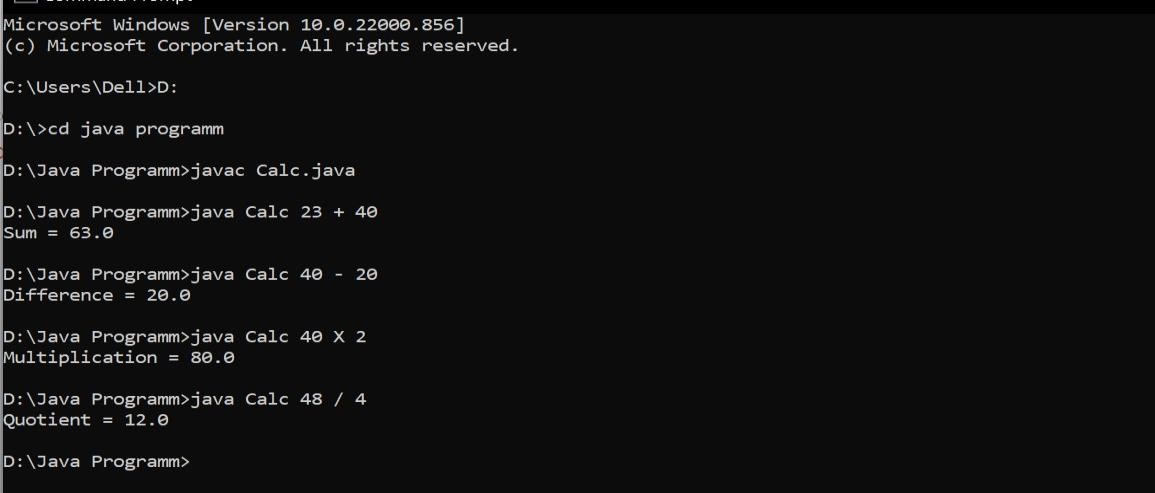
c.div(a , b );

}

}



**Output :**



**3. Write a program to accept 10 student’s mark in an array, arrange it into ascending order, convert into the following grades and print marks and grades in the tabular**

**form. Between 40 and 50: PASS Between 51 and 75 MERIT**

**and above DISTINCTION**

**Code:**

import java.util.\*; class

Grades

{

static void bubbleSort(int[] array) {

int n = array.length;

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

{

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

{

if(array[j-1] > array[j])

{ //swap elements temp = array[j-1]; array[j-1] = array[j]; array[j] = temp;

}

}

}

}

public static void main (String args[])

{

Scanner sc=new Scanner(System.in); int array[] = new int[10];

System.out.println("Enter the marks of students : "); for(int i=0; i<10; i++)

{

array[i] = sc.nextInt();

}

System.out.print("Mark of students :\n"); for(int i=0; i<10; i++)

{

System.out.print(array[i] + " ");

}

System.out.println();

bubbleSort(array);

System.out.println("Marks in assending order :"); for(int i=0; i <10; i++)

{

System.out.print(array[i] + " " );

}

System.out.println( );

for(int i=0; i <10; i++)

{

if (array[i]>0 && array[i]<39)

{

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

}

else if (array[i]>40 && array[i]<50)

{

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

}

else if (array[i]>51 && array[i]<75)

{

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

}

else if (array[i]>76 && array[i]<=100)

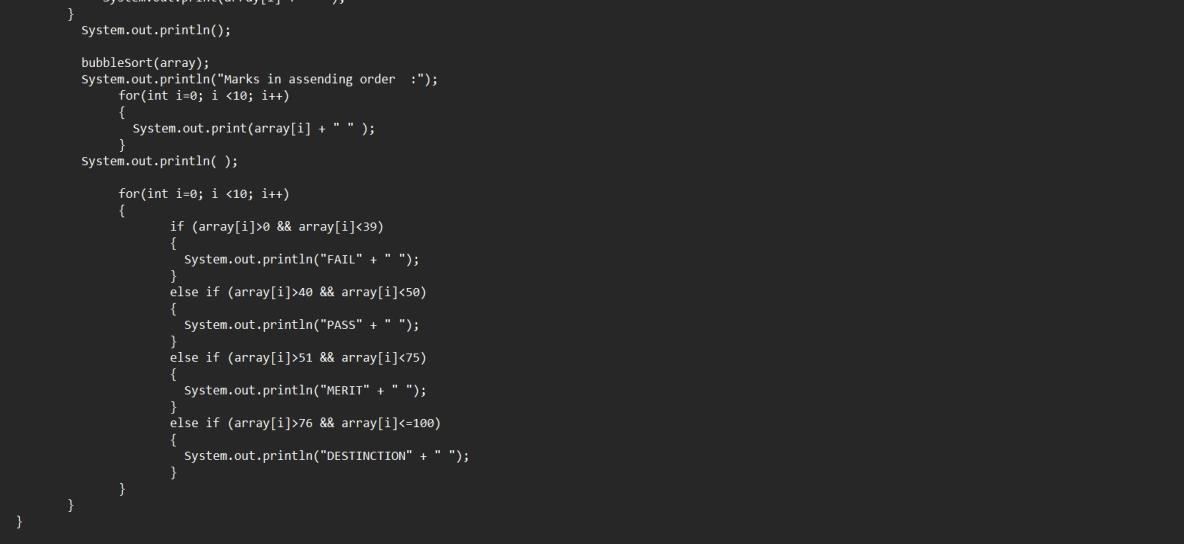
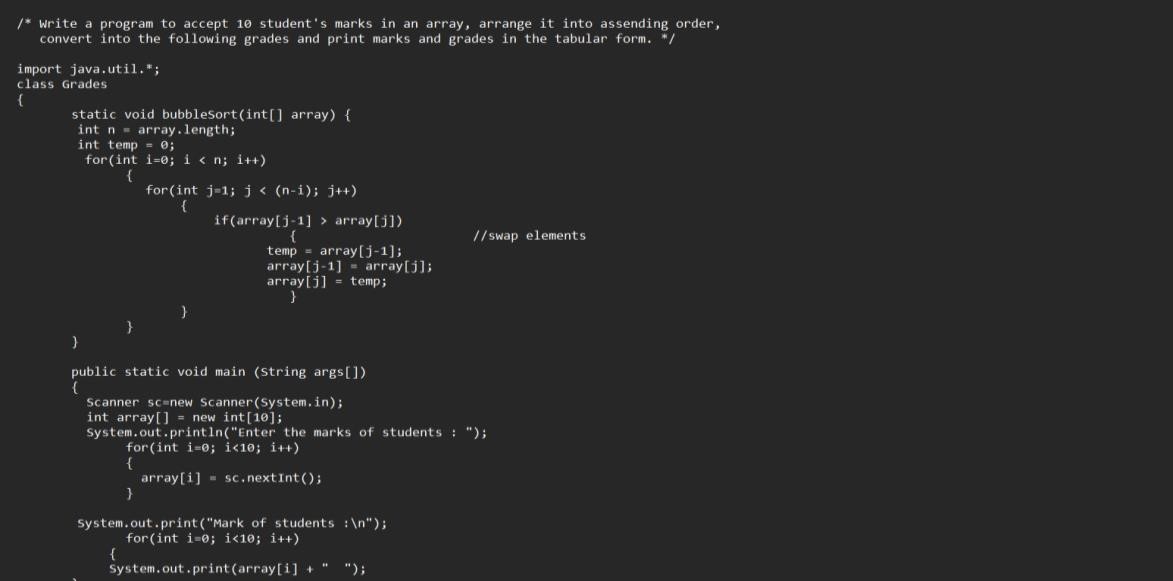
{

System.out.println("DESTINCTION" + " "); }

}

}

}



**Output:**



**4. WAP to take input as DD MM YYYY (04 08 2021) in command line and calculate number of days since 1 January 1970.**

**Code:** class GFG

{ static class Date

{ int d, m, y;

public Date(int d, int m, int y)

{ this.d = d;

this.m = m; this.y = y;

}

};

// To store number of days in all months from January to Dec.

static int monthDays[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

// This function counts number of leap years before the given date static int countLeapYears(Date d)

{ int years = d.y;

// Check if the current year needs to be considered for the count of leap years or not if (d.m <= 2)

{ years--;

}

// An year is a leap year if it is a multiple of 4, multiple of 400 and not a multiple of 100. return years / 4 - years / 100 + years / 400;

}

// This function returns number of days between two given dates static int getDifference(Date dt1, Date dt2)

{

// COUNT TOTAL NUMBER OF DAYS BEFORE FIRST DATE 'dt1'

// initialize count using years and day int n1 = dt1.y \* 365 + dt1.d;

// Add days for months in given date for (int i = 0; i < dt1.m - 1; i++)

{

n1 += monthDays[i];

}

// Since every leap year is of 366 days, // Add a day for every leap year n1 += countLeapYears(dt1);

// SIMILARLY, COUNT TOTAL NUMBER OF DAYS BEFORE 'dt2' int n2 = dt2.y \* 365 + dt2.d; for (int i = 0; i < dt2.m - 1; i++)

{

n2 += monthDays[i];

}

n2 += countLeapYears(dt2);

// return difference between two counts return (n2 - n1);

}

public static void main(String args[])

{

int d = Integer.parseInt(args[0]); int m = Integer.parseInt(args[1]); int y = Integer.parseInt(args[2]);

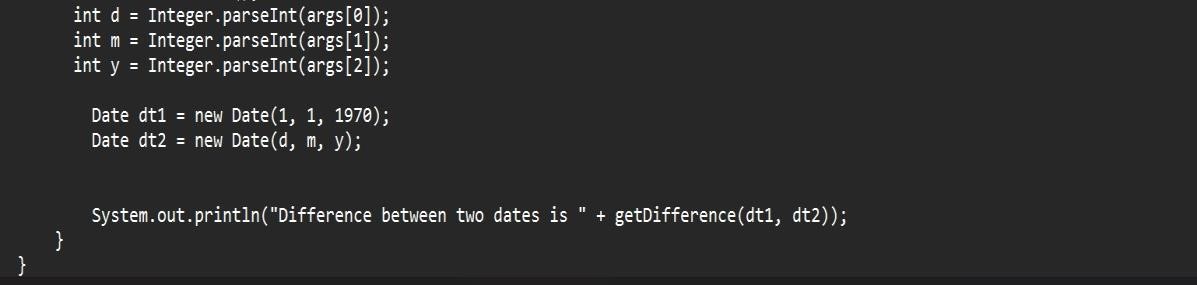
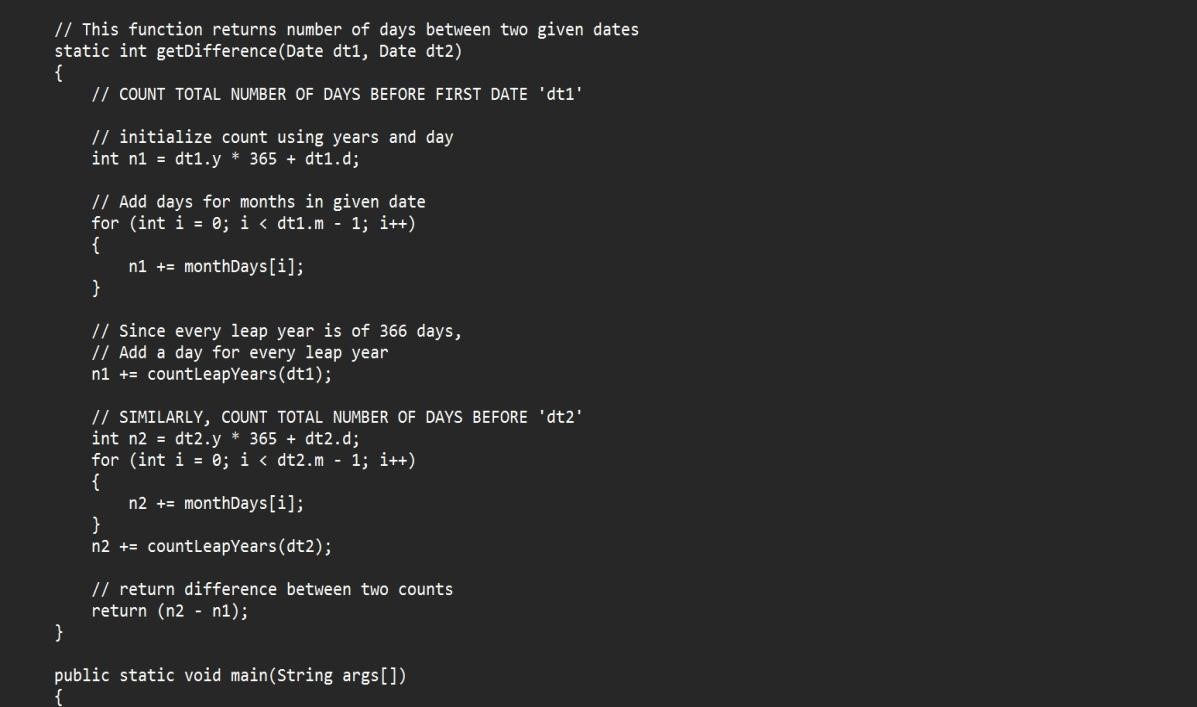
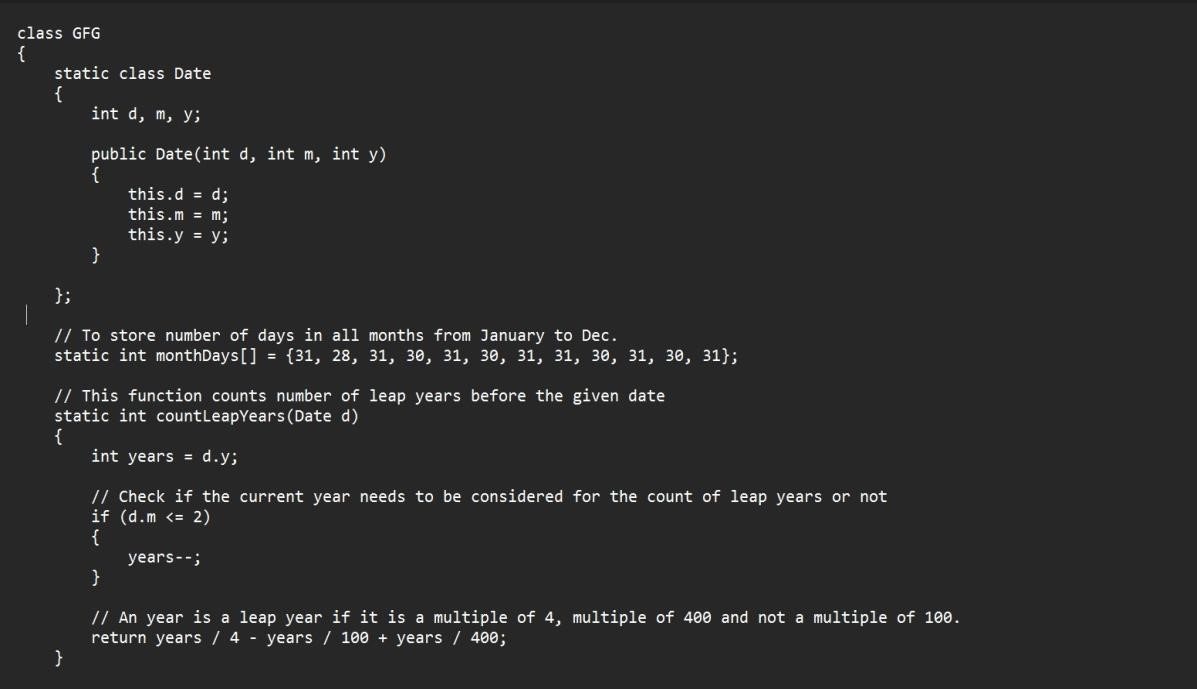
Date dt1 = new Date(1, 1, 1970);

Date dt2 = new Date(d, m, y);

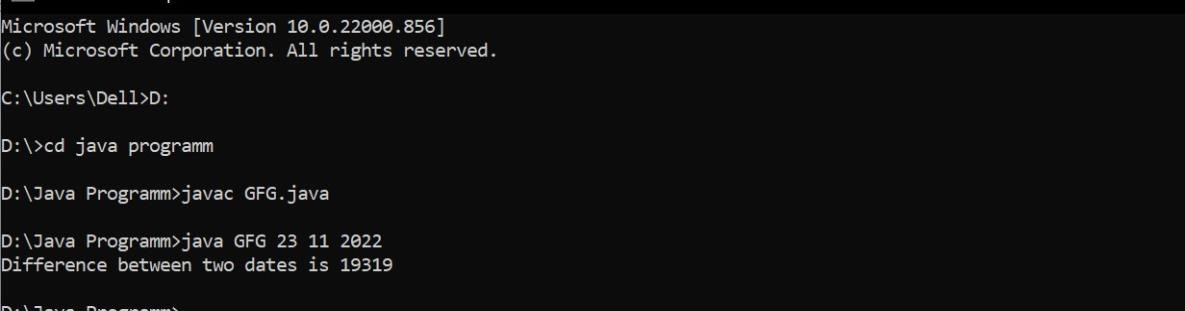
System.out.println("Difference between two dates is " + getDifference(dt1, dt2));

}

}



**Output:**



**5.WAP to print the following pattern using loop.**

**\***

* **\* \***
* **\* \* \* \* Code:**  class HalfPyramid{ public static void main(String[] args) {

int rows = 3, k = 0;

for (int i = 1; i <= rows; ++i, k = 0) {

while (k != 2 \* i - 1) {

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

++k;

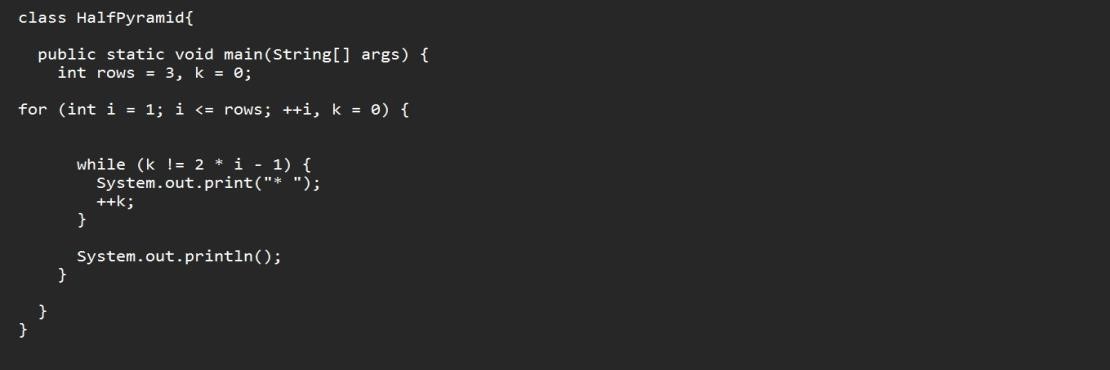
}

System.out.println();

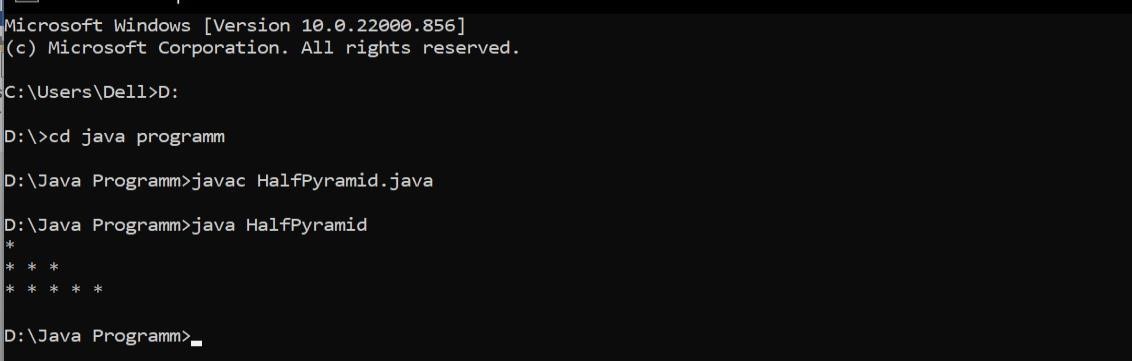
}

}

}



**Output:**



# Experiment-3

1.Write a program to accept three digits and print all the possible combinations.

**Code:**

import java.util.\*; class

Combinations{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in); System.out.println("Enter first digit : "); int a = sc.nextInt();

System.out.println("Enter second digit : "); int b = sc.nextInt();

System.out.println("Enter third digit : "); int c = sc.nextInt(); int[] input = { a, b, c };

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

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

System.out.println(100\*input[i]+ 10\*input[j]+ 1\*input[k]);

}

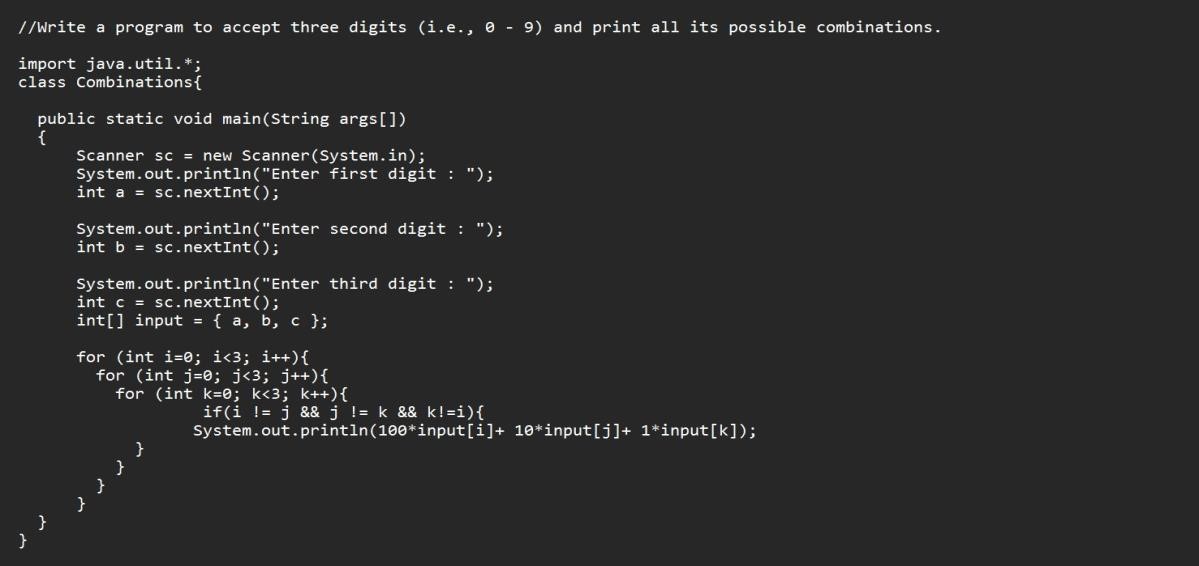
}

}

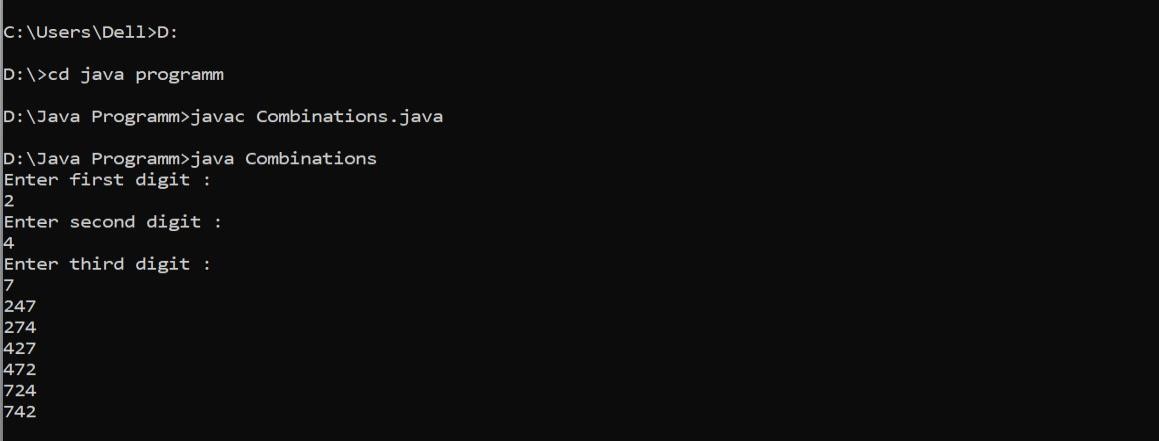
}

}

}



**Output:**



2. Write a java program to accept 10 numbers in an array and compute the square of each number. Print the sum of these numbers.

**Code:**

import java.util.\*; class SquareSum{ void square(int[] array)

{

int sum = 0;

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

{

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

}

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

}

public static void main(String args[])

{

int[] array;

array = new int[10]; int c

= 0;

Scanner sc = new Scanner(System.in); for(int i=0 ; i<array.length ; i++ )

{ c=i+1;

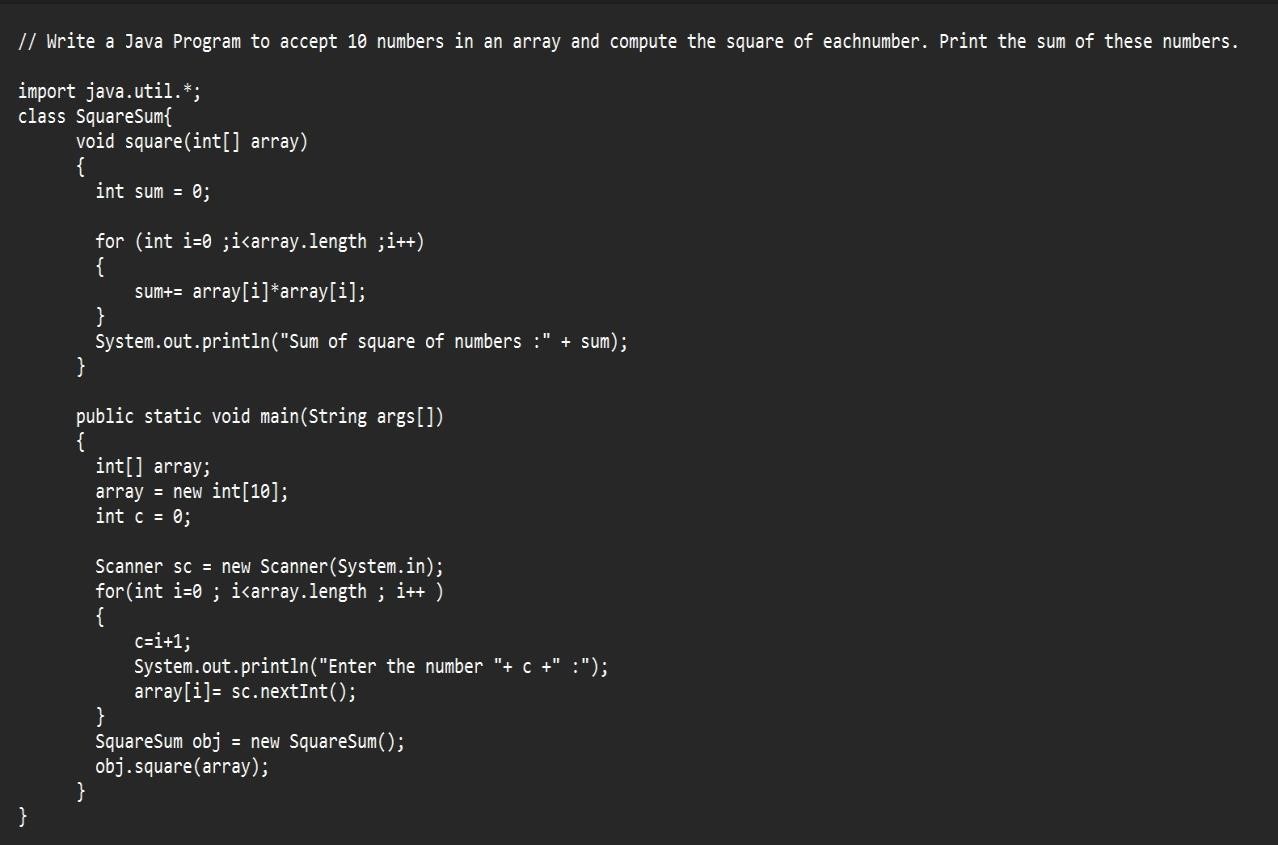
System.out.println("Enter the number "+ c +" :"); array[i]= sc.nextInt();

}

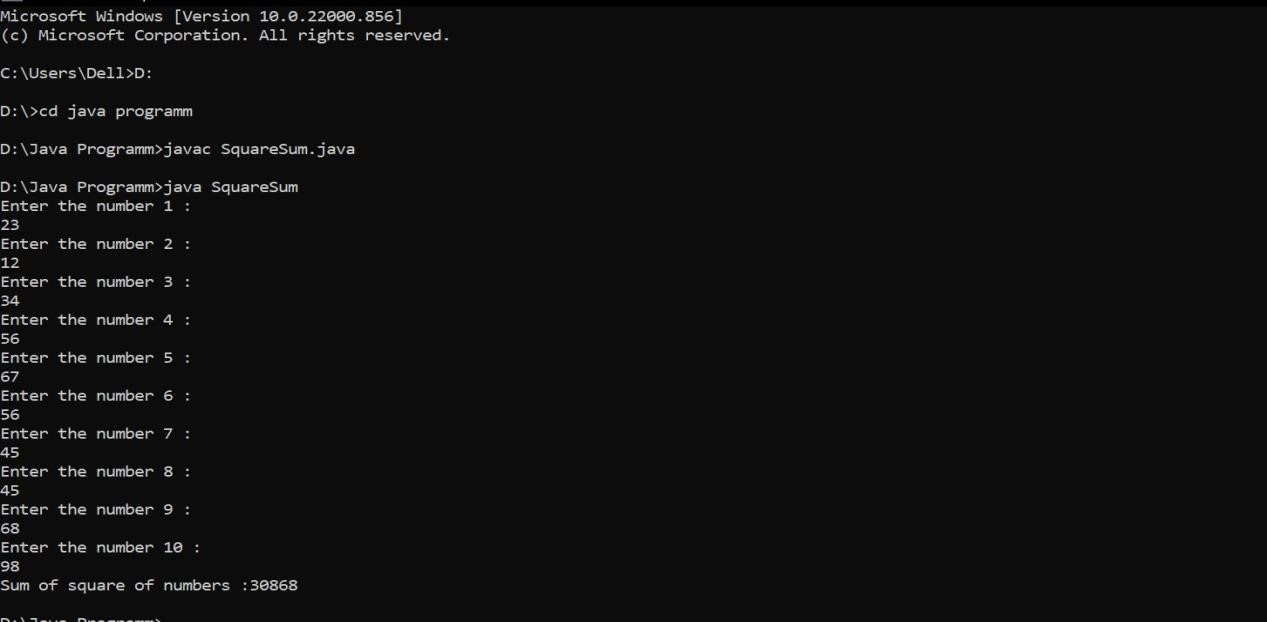
SquareSum obj = new SquareSum(); obj.square(array);

}

}



**Output:**



3.Write a program to input a number of a month (1-12) and print its equivalent name of the month.

**Code:**

import java.util.\*; class TestMonth{ void inputMonths(int n)

{

switch(n)

{ 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 Input");

}

}

public static void main(String args[])

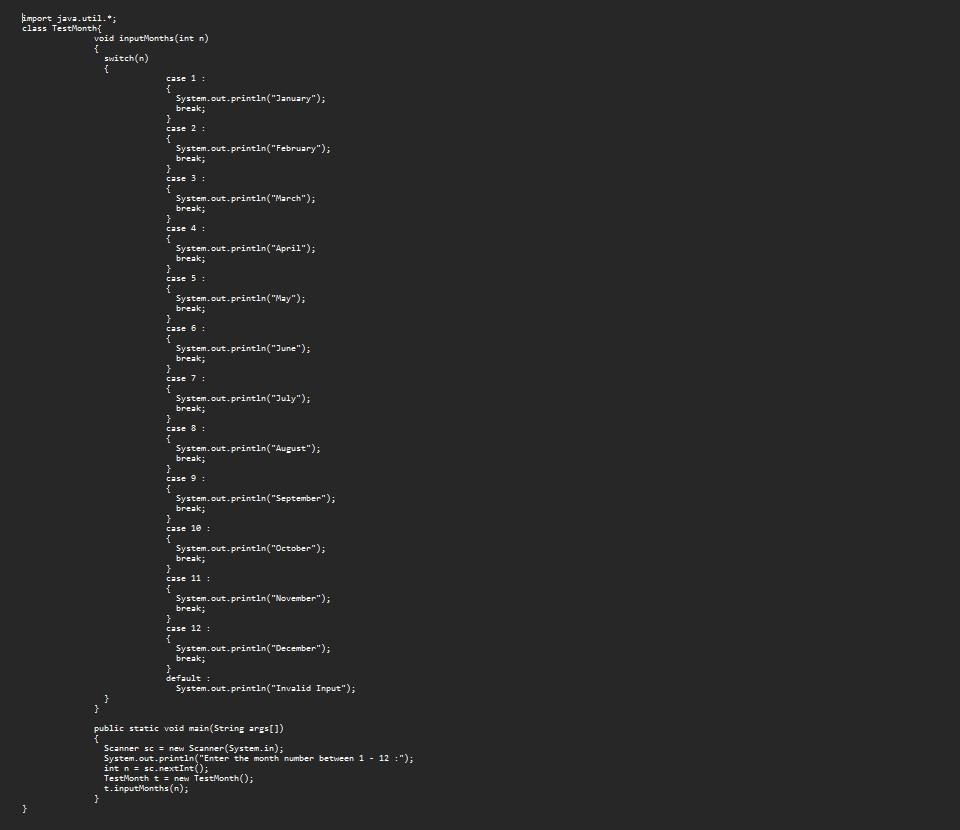
{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the month number between 1 - 12 :"); int n = sc.nextInt(); TestMonth t = new TestMonth(); t.inputMonths(n);

}

}



**Output:**



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

**Code:**  class DivisibleSum{ public static void main(String args[])

{

int sum= 0;

for (int i=41; i<250; i++)

{ if(i%5==0)

{

sum+=i;

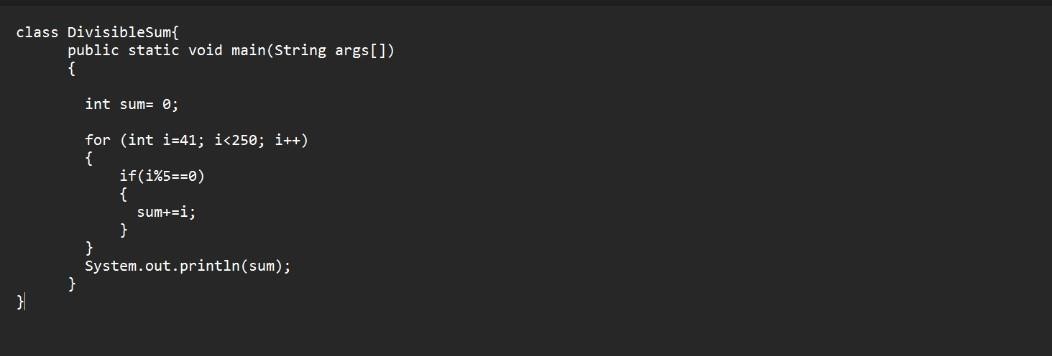
}

}

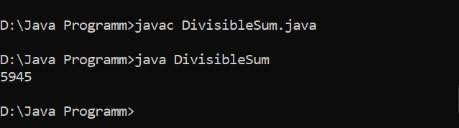
System.out.println(sum);

}

}



**Output:**



## Experiment-4

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

**Code:**

class Parent

{

int a; private int b;

protected int c;

Parent(int a, int b, int c)

{

this.a=a;

this.b=b; this.c=c;

}

public void print1()

{

System.out.println("public method");

}

private void print2()

{

System.out.println("private method");

}

protected void print3()

{

System.out.println("protected method");

} //main

} //class class Child extends Parent

{

private int d;

Child(int a , int b , int c , int d)

{

super(a,b,c);

this.d = d;

} //main

} //class class

Test

{

public static void main(String args[])

{

Child c1 = new Child(1,2,3,4); c1.print1(); c1.print2(); c1.print3();

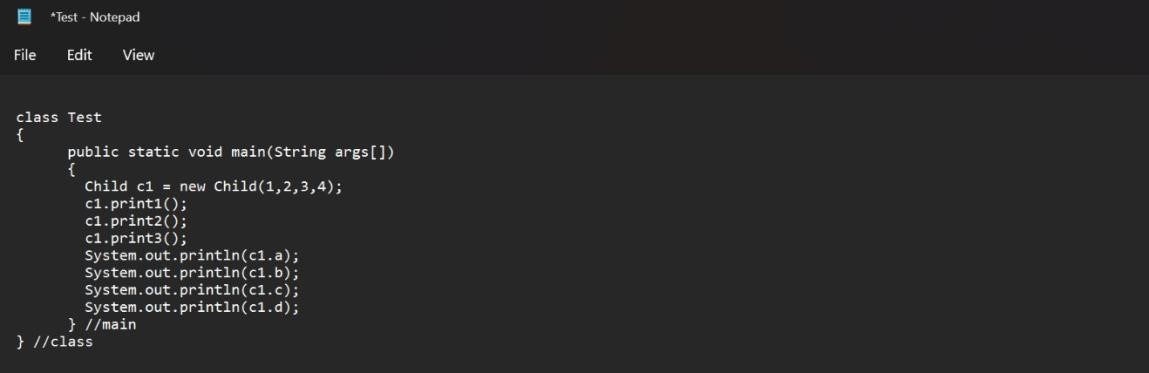
System.out.println(c1.a);

System.out.println(c1.b); System.out.println(c1.c);

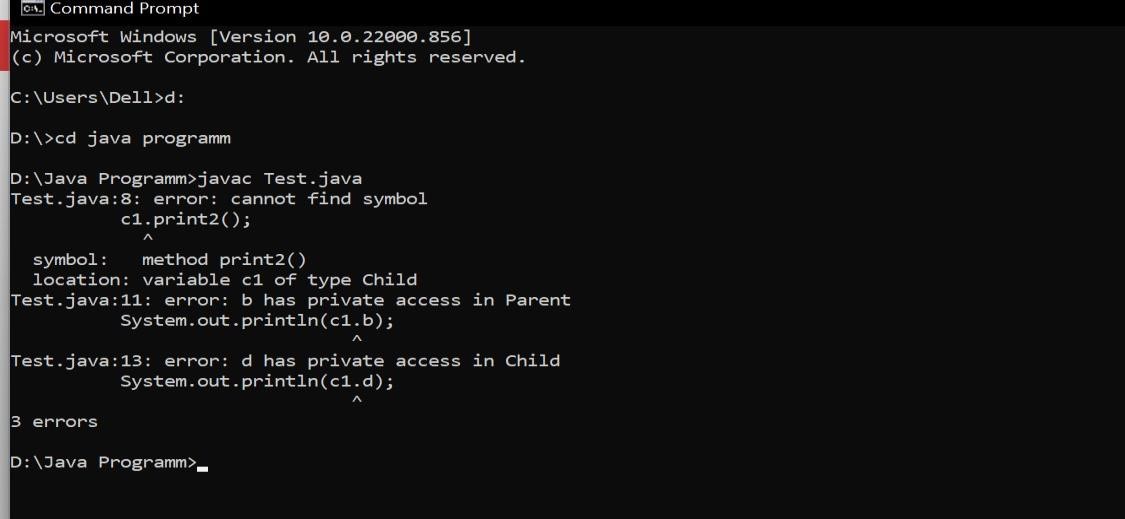
System.out.println(c1.d);

} //main

} //class



**Output:**



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

**Code:**

class Player{

String name; int age; double height;

Player(String name, int age, double height){ this.name= name; this.age= age; this.height= height;

}

void printDetails()

{

System.out.println("Name : " + name +" Age : " + age +" Height : " + height);

}

}

class Cricket\_Player extends Player{

String teamname;

String position;

Cricket\_Player(String name, int age, double height, String teamname, String position){ super(name, age, height);

this.teamname= teamname; this.position= position;

}

void printDetails()

{

System.out.println("Name : " + name +" Age : " + age +" Height : " + height+ "

Team Name :"+teamname+" Position : "+position);

}

}

class Football\_Player extends Player{

String teamname;

String position;

Football\_Player(String name, int age, double height, String teamname, String position){ super(name, age, height);

this.teamname= teamname; this.position= position;

}

void printDetails()

{

System.out.println("Name : " + name +" Age : " + age +" Height : " + height+ " Team Name :"+teamname+" Position : "+position);

}

}

class Hockey\_Player extends Player{

String teamname;

String position;

Hockey\_Player(String name, int age, double height, String teamname, String position){ super(name, age, height);

this.teamname= teamname; this.position= position;

}

void printDetails()

{

System.out.println("Name : " + name +" Age : " + age +" Height : " + height+ " Team Name :"+teamname+" Position : "+position);

}

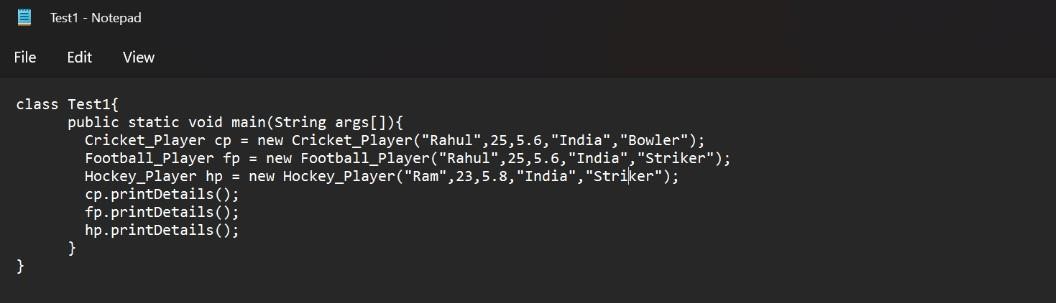
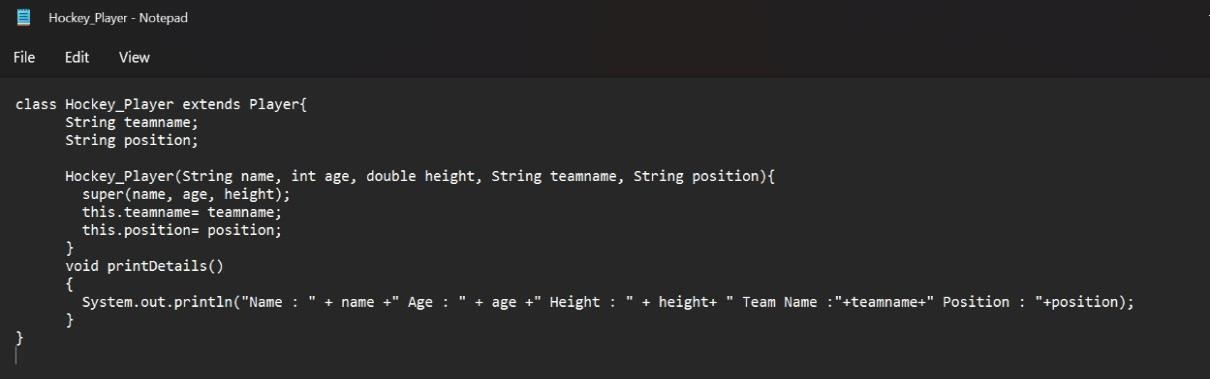
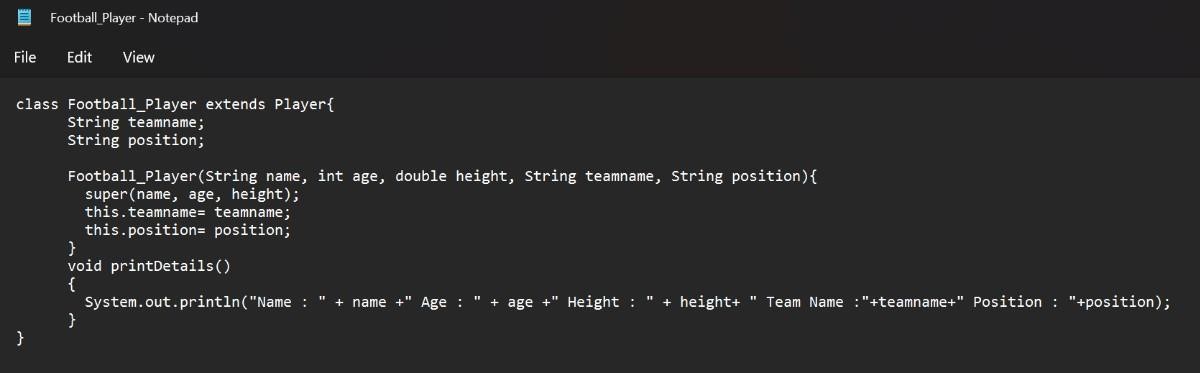
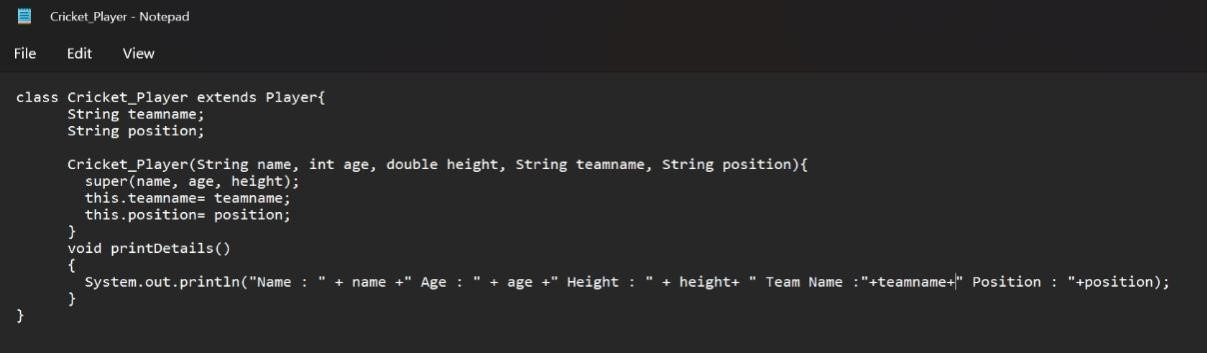
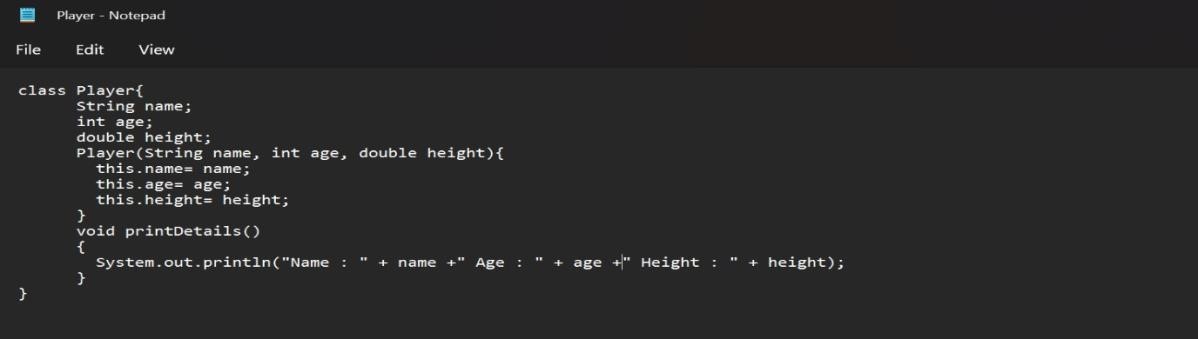
} class Test1{ public static void main(String args[]){

Cricket\_Player cp = new Cricket\_Player("Rahul",25,5.6,"India","Bowler");

Football\_Player fp = new Football\_Player("Rahul",25,5.6,"India","Striker"); Hockey\_Player hp = new Hockey\_Player("Ram",23,5.8,"India","Striker"); cp.printDetails(); fp.printDetails(); hp.printDetails();

}

}



**Output:**



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 day she works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

**Code:**

class Worker{

String name; int salaryrate;

Worker(String name,int salaryrate)

{

this.name = name; this.salaryrate = salaryrate;

}

}

class SalariedWorker extends Worker

{

int hours;

SalariedWorker(String name,int salaryrate,int hours)

{

super(name,salaryrate); this.hours = hours;

}

int ComPay(int hours)

{

int salary = 40\*salaryrate;

System.out.println("Name : "+name+" "+"salary : "+40\*salaryrate); return salary;

}

}

class DailyWorker extends Worker

{

int days;

DailyWorker(String name,int salaryrate,int days)

{

super(name,salaryrate); this.days = days;

}

int ComPay(int days)

{

int salary = days\*salaryrate;

System.out.println("Name : "+name+" "+"salary : "+days\*salaryrate); return salary;

}

}

class Test3

{

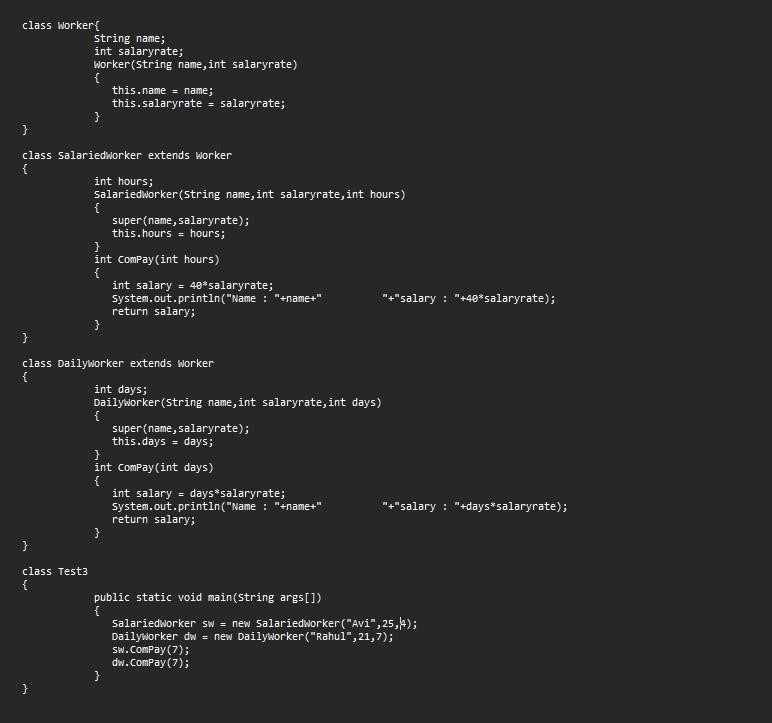
public static void main(String args[])

{

SalariedWorker sw = new SalariedWorker("Avi",25,4); DailyWorker dw = new DailyWorker("Rahul",21,7); sw.ComPay(7); dw.ComPay(7);

}

}



**Output:**



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

**Code:**

import java.util.\*; class Calculation

{

double rate; double trunk(int type , double duration)

{

rate=0.5; //urgent rate return (rate\*duration);

}

//double trunk(int type, double duration)

{

rate=0.3; //lightning rate

// return (rate\*duration);

}

//double trunk(int type , double duration)

{

rate=0.1; //ordinary rate

}

} //calculation

class Telephone extends Calculation

{

public static void main(String args[])

{

Calculation obj=new Calculation();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the duration of call: ");

double duration=sc.nextDouble();

System.out.println("Enter \n 1: Urgent call \n 2: lightning call \n 3: ordinary call"); int type=sc.nextInt(); double ans=0.0; switch(type)

{ case 1:

ans=obj.trunk(type,duration);

break; case 2:

ans=obj.trunk(type,duration);

break; case 3:

ans=obj.trunk(type,duration);

break; default:

System.out.println("Incorrect Input");

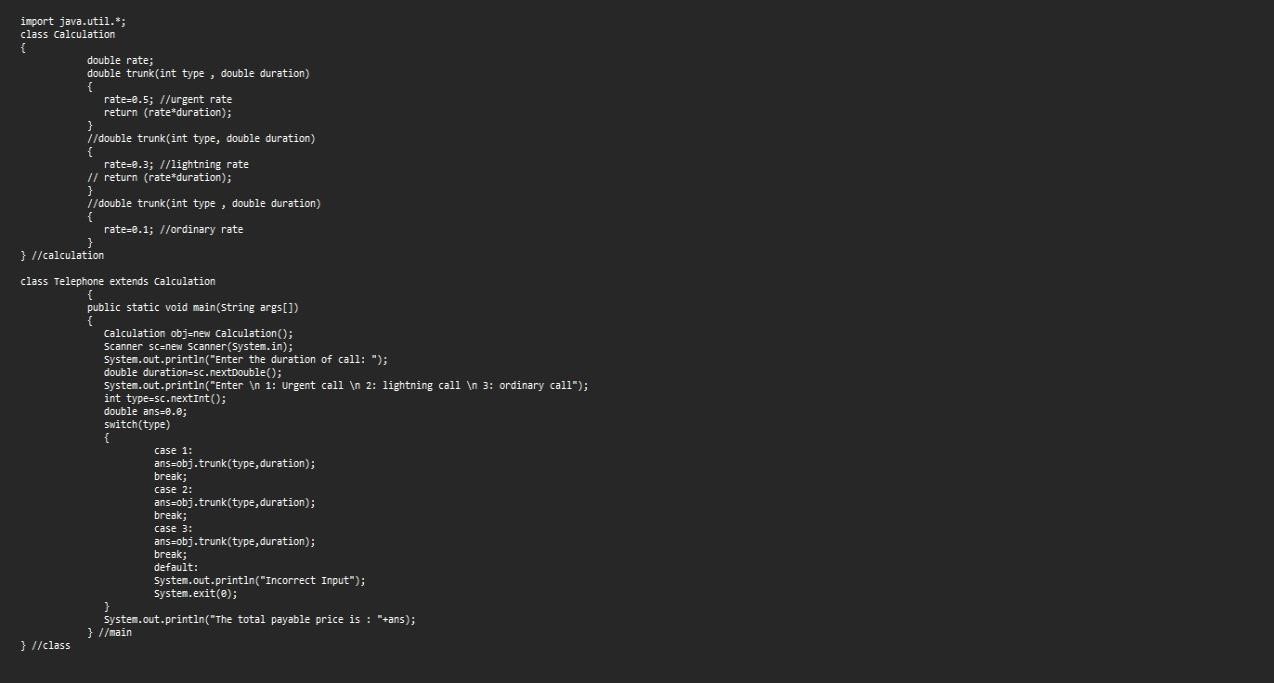
System.exit(0);

}

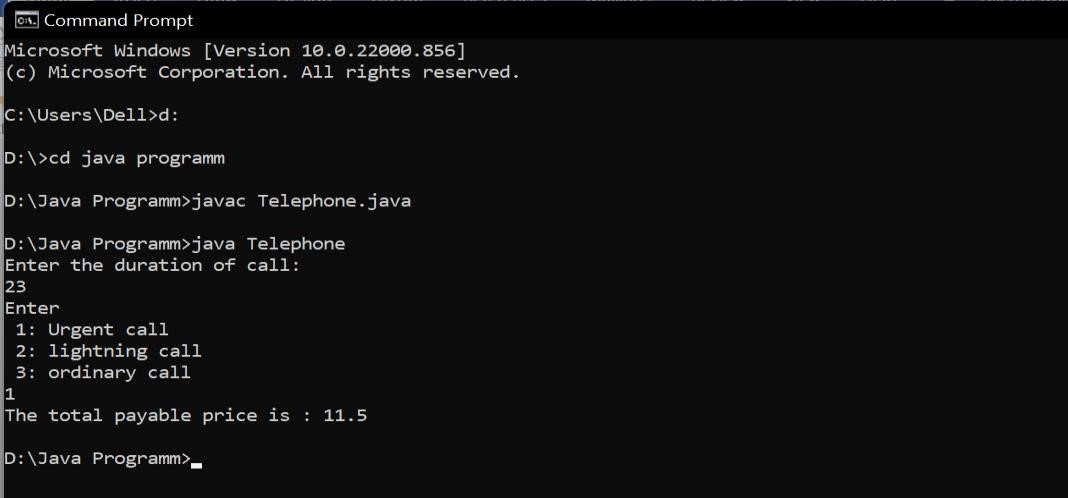
System.out.println("The total payable price is : "+ans);

} //main

} //class



**Output:**



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.

**Code:**

import java.util.\*; class Employee

{

String name,id; double salary;

Employee(String name,String id,double salary)

{

this.name=name;

this.id=id; this.salary=salary;

}

String name(){ return name;

}

double salary(){ return salary;

}

void Increasesalary(double salaryp)

{

this.salary\*=salaryp;

}

void display(){

System.out.print(name+" "+salary+" "+id);

} //display

} //class

class Manager extends Employee

{

String department;

Manager(String name,String id,double salary,String department)

{

super(name,id,salary);

this.department=department;

}

void display()

{

System.out.print(department+" "); super.display();

}//display

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter :- \t Department \t Name\t Salary \t ID \n");

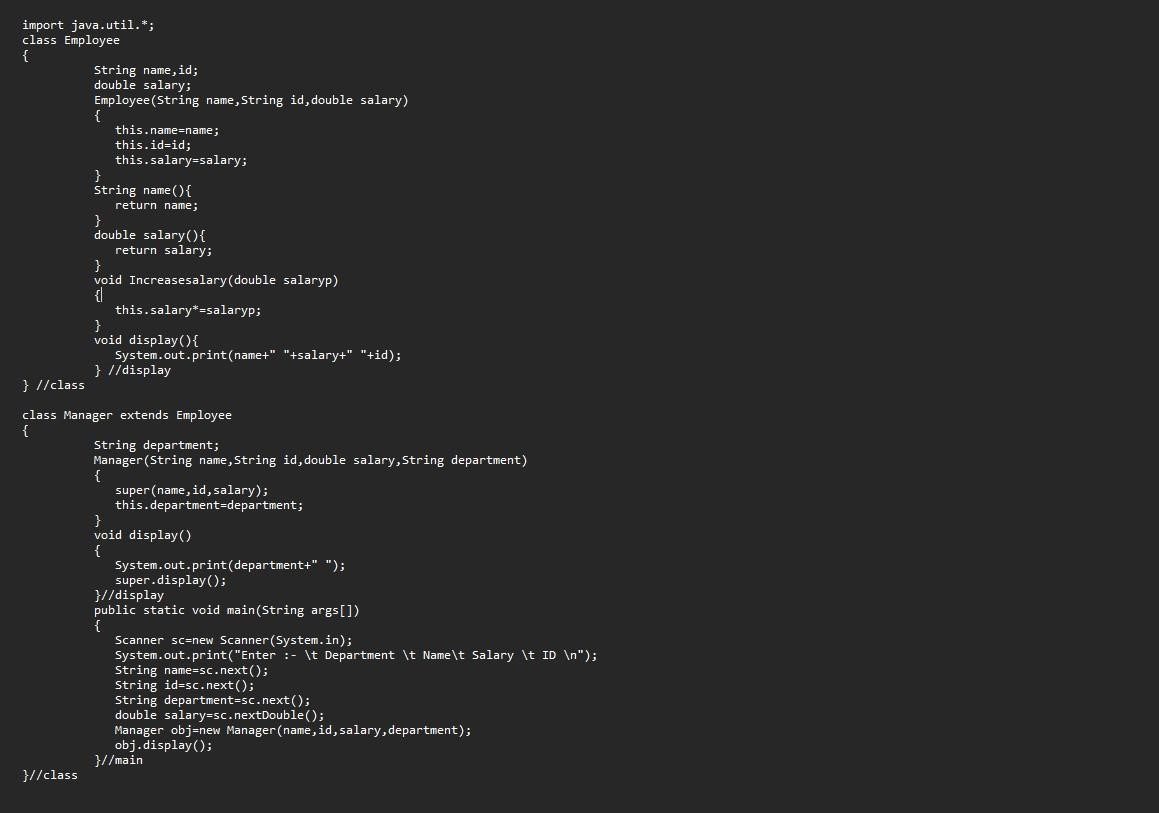
String name=sc.next();

String id=sc.next(); String department=sc.next(); double salary=sc.nextDouble();

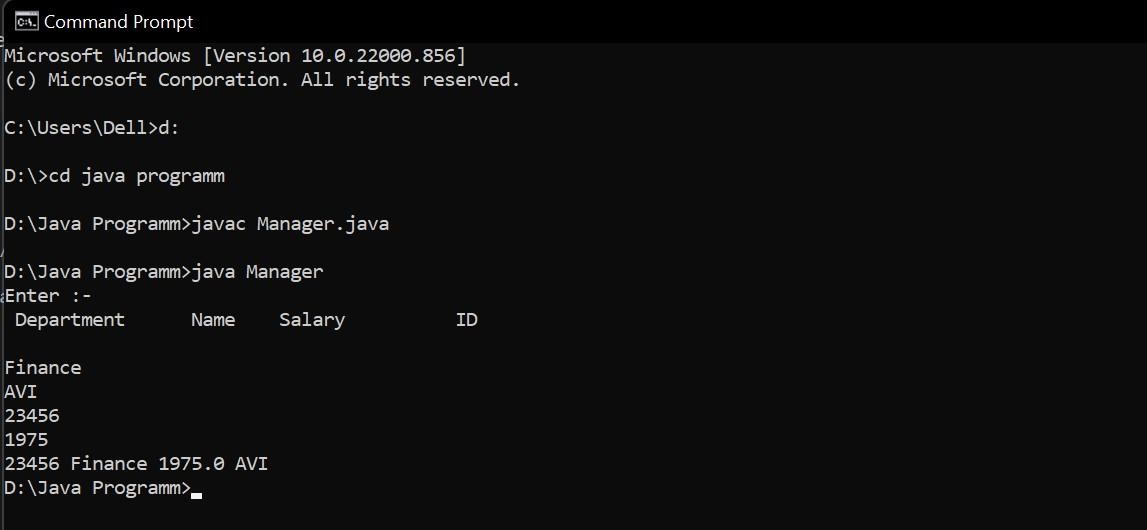
Manager obj=new Manager(name,id,salary,department); obj.display();

}//main

}//class



**Output:**



## Experiment-5(Package)

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

**Code:**  package Mathematics; import

java .lang.\*; public class Arthematics{ public int add(int x, int y)

{

return x + y;

}

}

package Calculator; import Mathematics.\*; public class Test5{

public static void main(String args[])

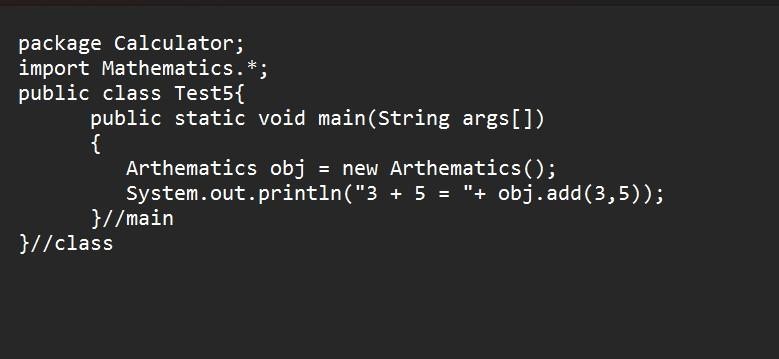
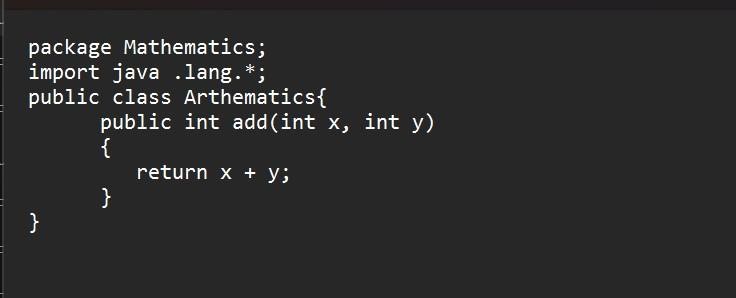
{

Arthematics obj = new Arthematics();

System.out.println("3 + 5 = "+ obj.add(3,5));

}//main

}//class



**Output:**



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.

**Code:**  package balance; public

class Account{

public double display\_Balance(double a, double b)

{

return 100\*a + b;

}//method

}//class

package calculator;

import balance.\*; public class Test6{

public static void main(String args[])

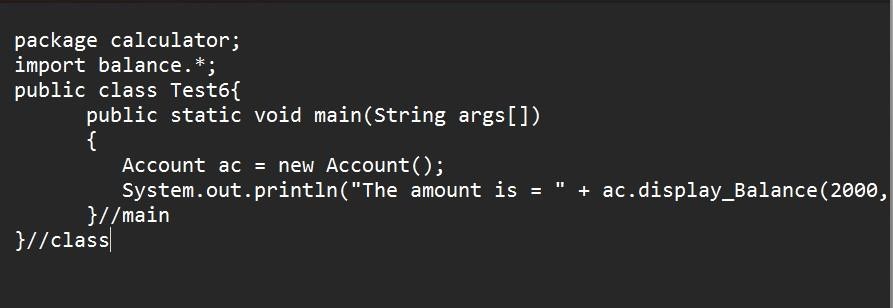
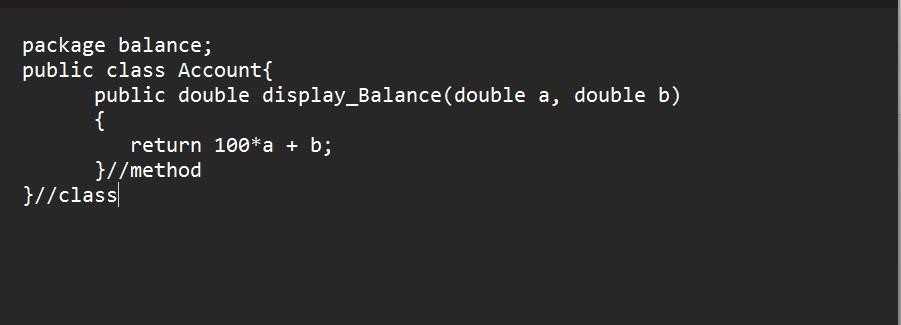
{

Account ac = new Account();

System.out.println("The amount is = " + ac.display\_Balance(2000,5000));

}//main

}//class



**Output:**



3. WAP to create a package p with class A with 4 types of access protected methods. How we will use these methods in different packages class i.e. there is main() in class B in package Q and 4 methods are in Class A in package p.

**Code:** When we call only protected and public methods package p; public class A{

protected void message1()

{

System.out.println("In protected method.");

}

private void message2()

{

System.out.println("In private method.");

}

void message3()

{

System.out.println("In default method.");

}

public void message4()

{

System.out.println("In public method.");

}

} package q; import p.\*;

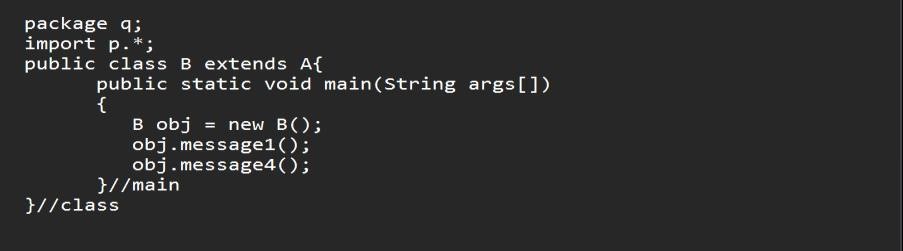
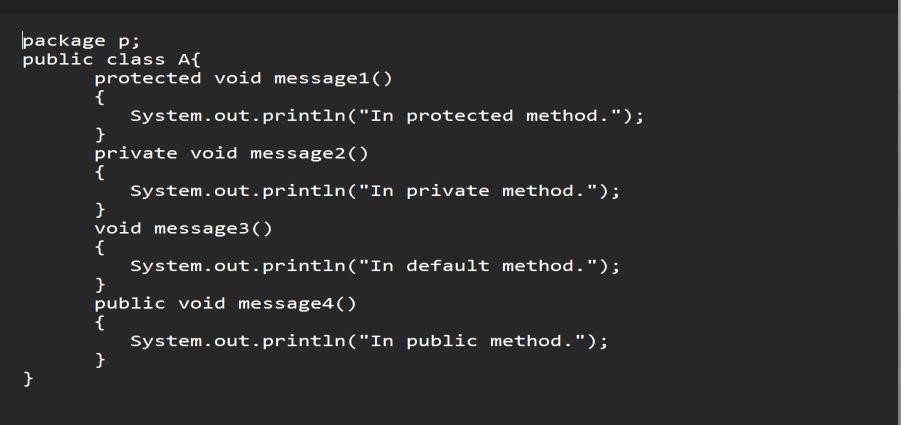
public class B extends A{ public static void main(String args[])

{

B obj = new B(); obj.message1(); obj.message4();

}//main

}//class



**Output:**



**Code:** When we call all type of methods

package p; public

class A{

protected void message1()

{

System.out.println("In protected method.");

}

private void message2()

{

System.out.println("In private method.");

}

void message3()

{

System.out.println("In default method.");

}

public void message4()

{

System.out.println("In public method.");

}

} package q; import p.\*; public class B extends A{ public static void main(String args[])

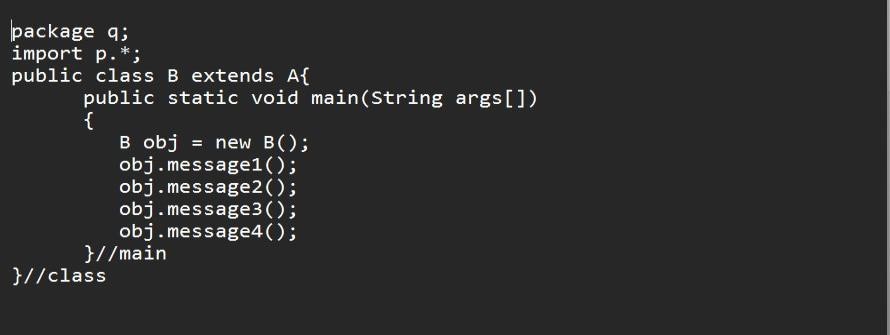
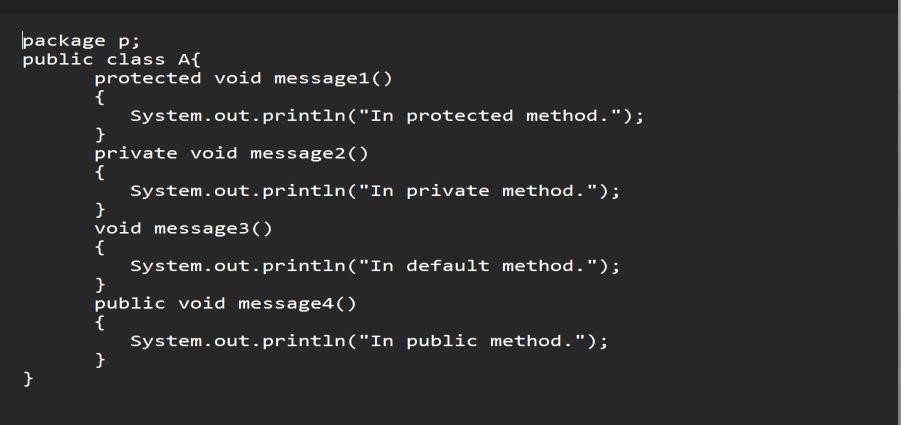
{

B obj = new B(); obj.message1(); obj.message2(); obj.message3();

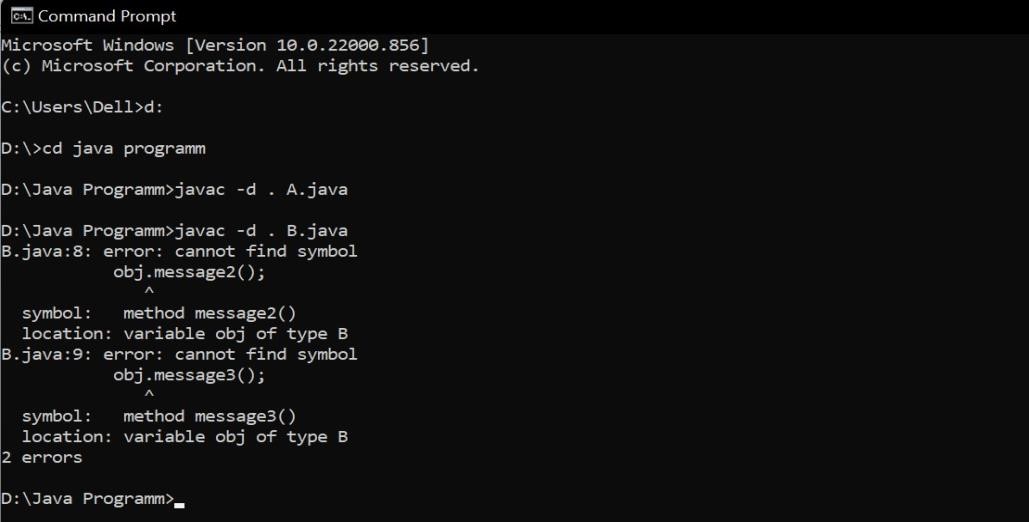
obj.message4();

}//main

}//class



**Output:**



## Experiment-6(Interfaces)

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

**Code:**

interface Test9

{

void square(int a);

}

class Arthematic1 implements Test9

{

public void square(int n)

{

System.out.println("Square of number is :" + (n\*n));

}//method

}//class class ToTestInt

{

public static void main(String args[])

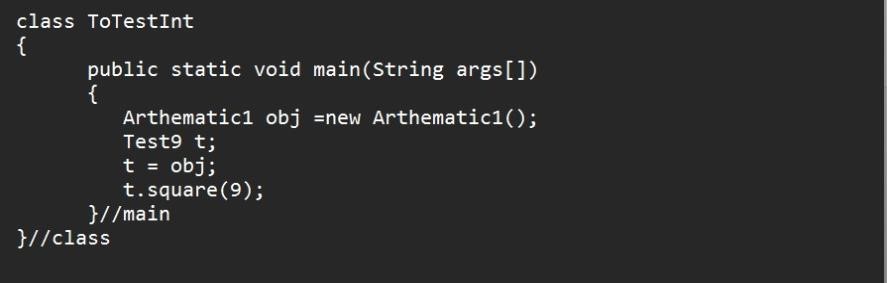
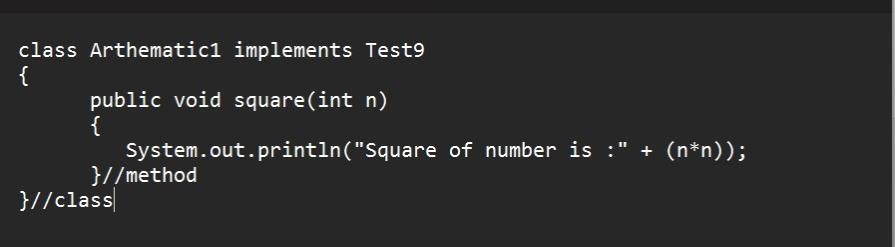
{

Arthematic1 obj =new Arthematic1(); Test9 t; t = obj;

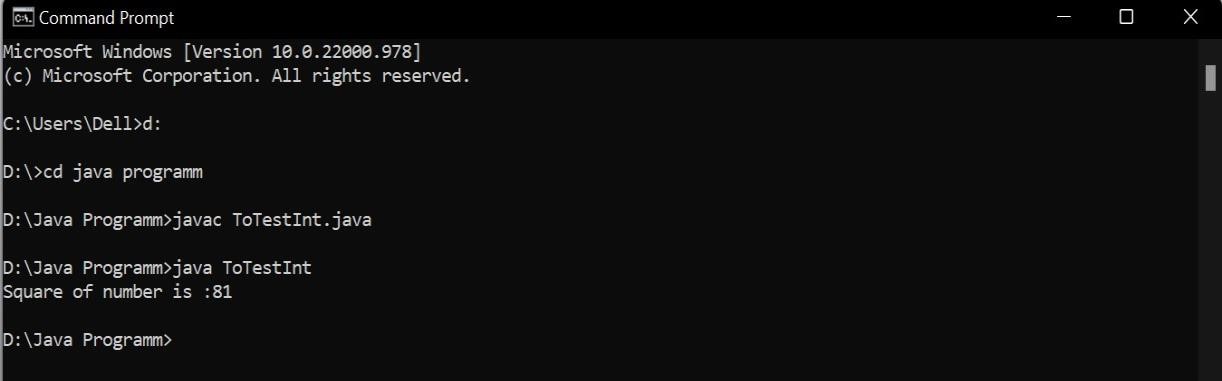
t.square(9);

}//main

}//class



**Output:**



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

**Code:**

interface A1{

void meth1(); void meth2();

}

class MyClass implements A1

{

public void meth1()

{

System.out.println("In method 1");

}//meth1 public void meth2()

{

System.out.println("In method 2");

}//meth2

}//class class

Demo

{

public static void main(String args[])

{

MyClass obj = new MyClass();

A1 a;//referance variable a

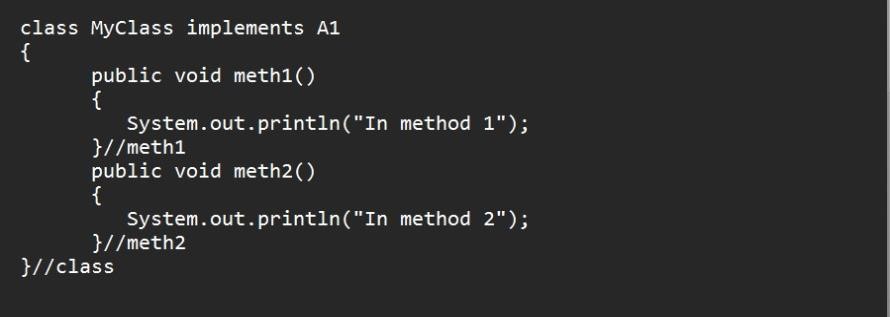
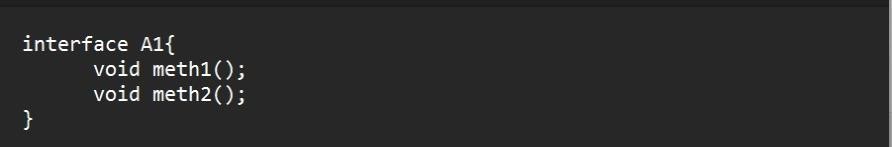
= obj;

a.meth1();

a.meth2();

}//main

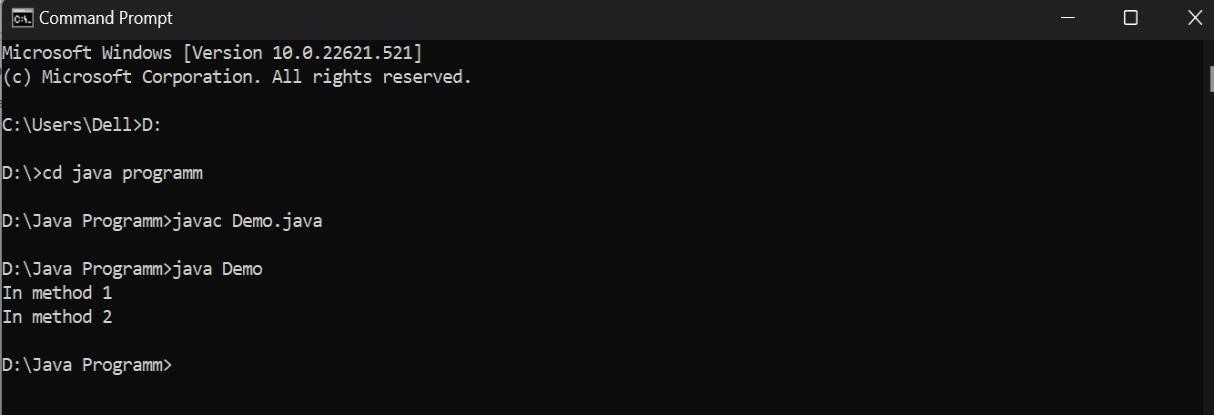
}//class



Graphical user interface, application

Description automatically generated

**Output:**



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

**Code:**

interface Compute

{

public static final float PI= 3.14f; void area();

}

class Circle implements Compute

{

float radius ;

Circle(float radius,float PI)

{

this.radius=radius;

float PI=3.142f;

} public void area()

{ float PI=3.142f;

System.out.println("Area of Circle is : " + (PI\*radius\*radius)); }

} class

Demo1{ public static void main(String args[])

{

Compute r;

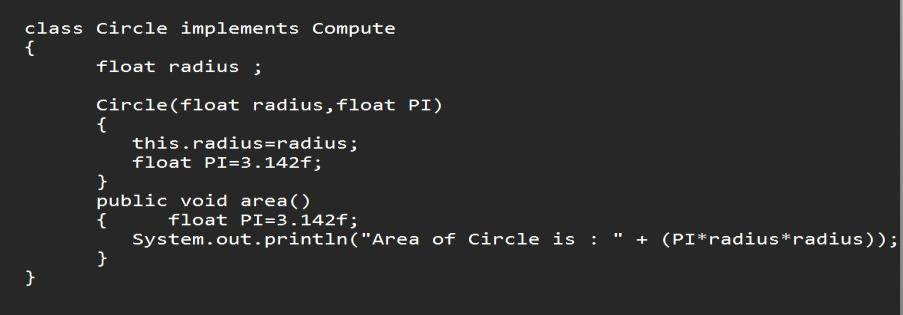
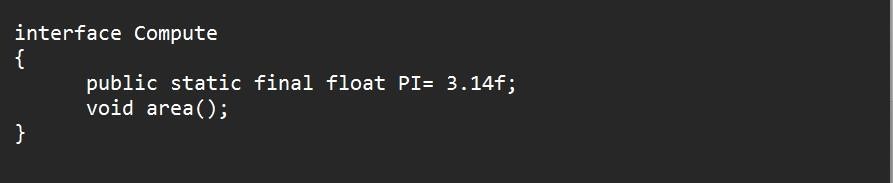
Rectangle obj1 = new Rectangle(10.0f,5.0f); r= obj1; r.area();

Circle obj2 = new Circle(5.0f); r=obj2;

r.area();

}

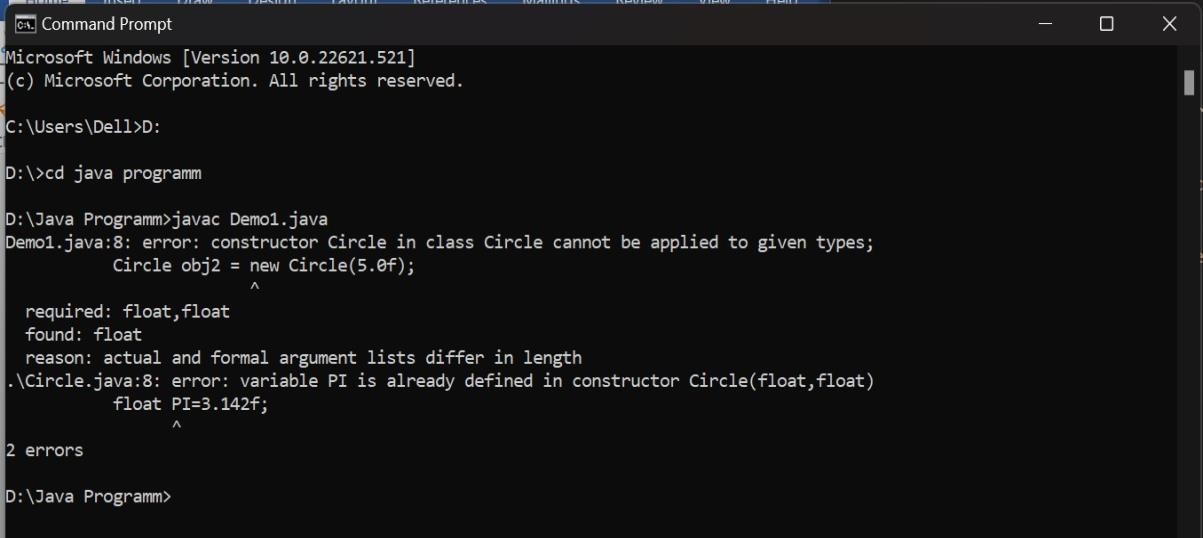
}



Graphical user interface, application

Description automatically generated

**Output:**



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

**Code:**

interface Calc1{ void division();

void modules();

}

class Operation implements Calc1

{

float a,b;

Operation(float a,float b)

{

this.a=a;

this.b=b; }

public void division()

{

System.out.println("Quotient : " + (a/b));

}

public void modules()

{

System.out.println("Remainder : " + (a%b));

}

} class

Demo3{ public static void main (String args[])

{

Calc1 c;

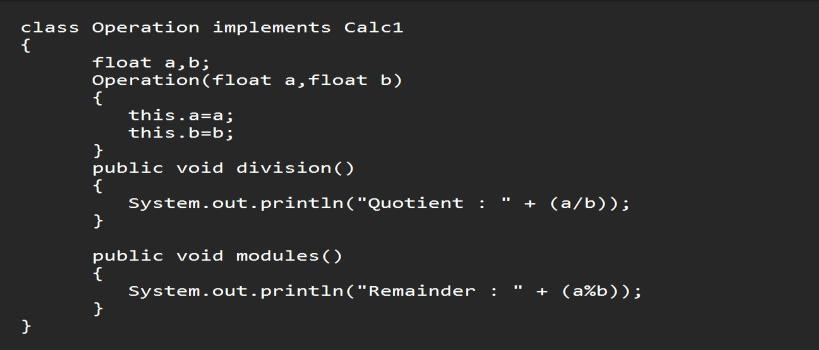
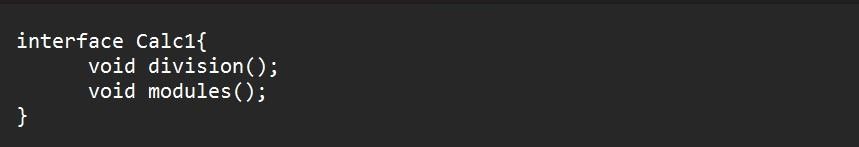
Operation obj= new Operation(155,5); c= obj;

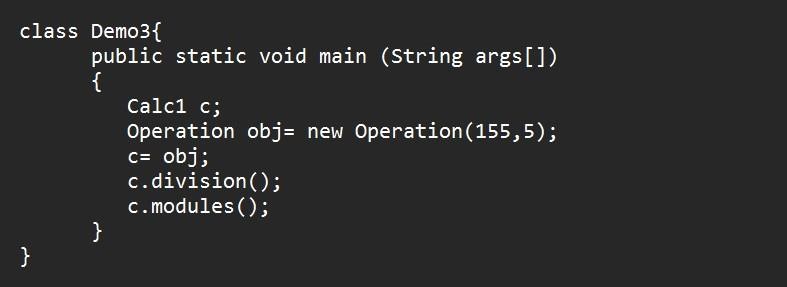
c.division();

c.modules();

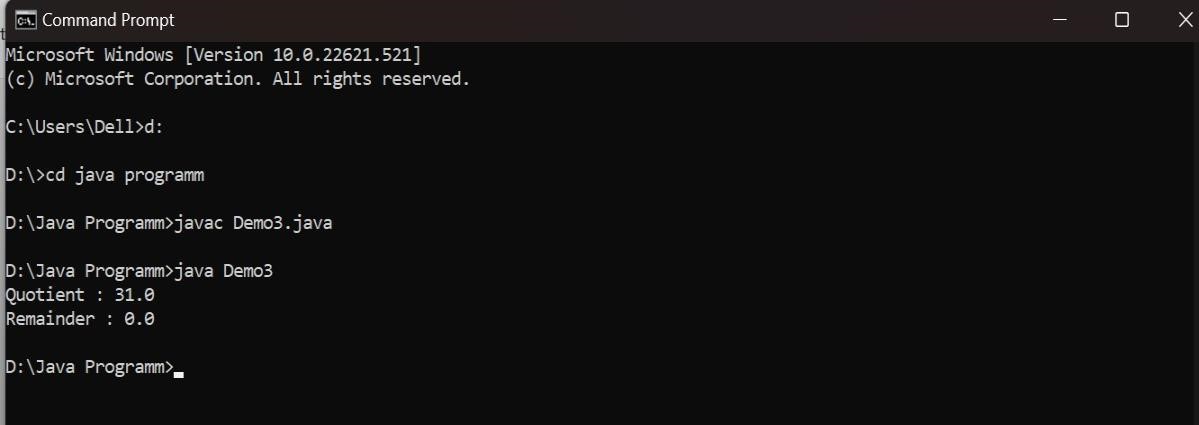
}

}





**Output:**



5. Write program to create an interface StackInterface having methods push (), pop () and display (). StackClass implements StackInterface. Class StackClass contains the main method which is having a switch case for selecting the particular operation of the stack.

**Code:**

interface StackInterface{ void push(int a); void pop(); void display();

} import java.util.\*;

class StackClass implements StackInterface{

int top=-1; int ar[]=new int[10];

public void push(int a){ if(top==ar.length-1){

System.out.println("overflow");

}

else{

top+=1; ar[top]=a;

}

}

public void pop(){ if(top==-1){

System.out.println("underflow");

} else{

System.out.println("element removed is"+ar[top]);

top-=1;

}

}

public void display(){

System.out.println("The elements are :"); for(int i=0;i<=top;i++){

System.out.print(ar[i] + " ");

}

}

} import java.util.Scanner;

class Demo4{

public static void main(String args[]){ int m=0; int n;

Scanner obj=new Scanner(System.in); StackClass obj1=new StackClass(); while(m!=1){

System.out.println("\n1.push\n2.pop\n3.display"); n=obj.nextInt(); switch(n){ case 1

:

System.out.println("enter the number"); int c=obj.nextInt();

obj1.push(c);

break; case 2 :

obj1.pop(); break; case 3 :

obj1.display(); break; default :

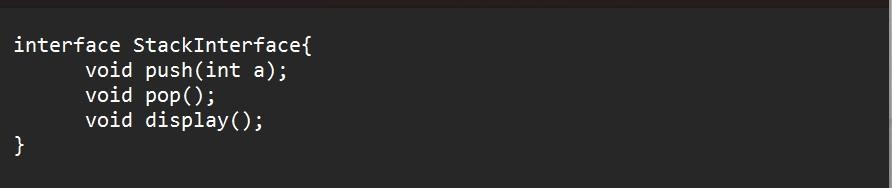
System.out.println("wrong input");

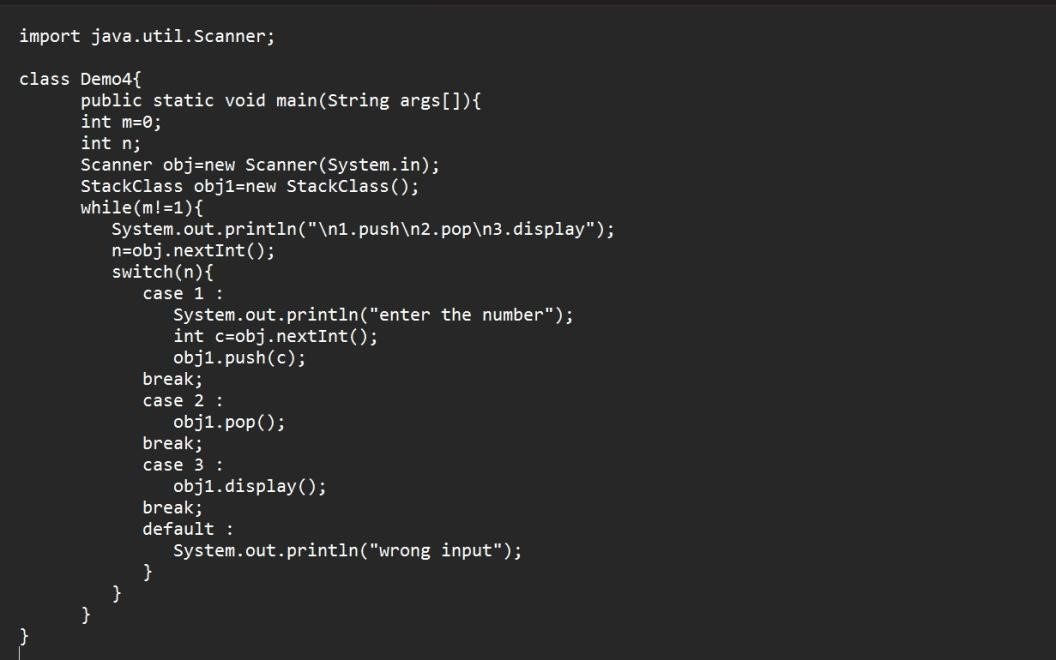
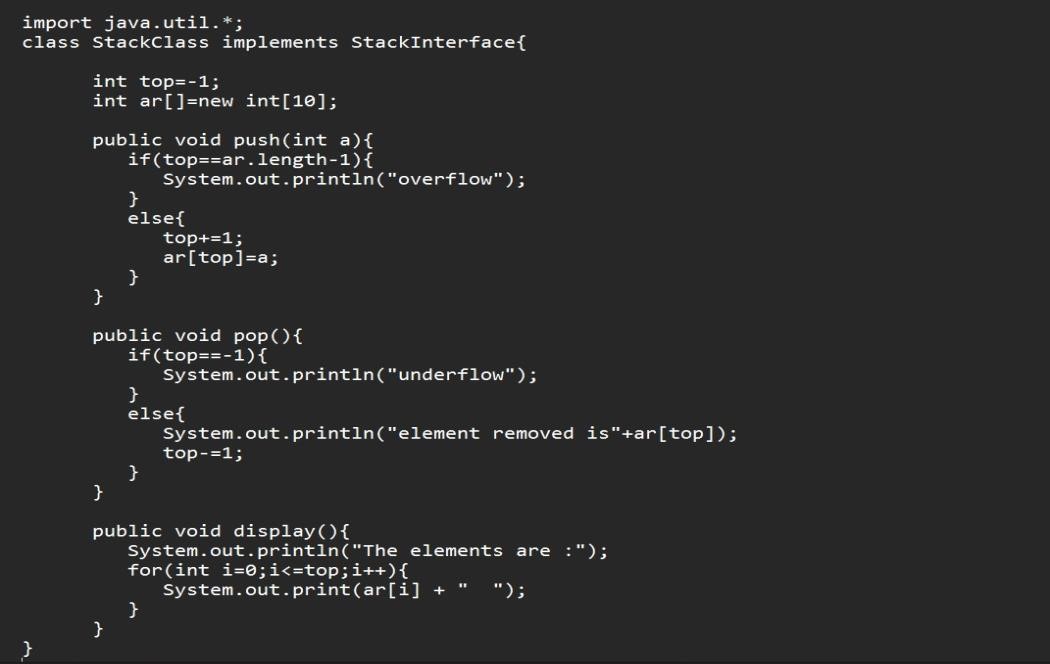
}

}

}

}





**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

## Experiment-7(Exception)

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.

**Code:** class CostomException extends Exception{ CostomException(String msg){ super(msg);

}

} class Main1{

public static void main(String args[]){ StudentRecords[] obj = new StudentRecords[10]; try{

obj[1]=new StudentRecords("A",1); obj[2]=new StudentRecords("B",2); obj[3]=new StudentRecords("C",3); obj[4]=new StudentRecords("D",4); obj[5]=new StudentRecords("E",5); obj[6]=new StudentRecords("F",6); obj[7]=new StudentRecords("G",7); obj[8]=new StudentRecords("H",8); obj[9]=new StudentRecords("I",9); obj[10]=new StudentRecords("J",10);

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println(e);

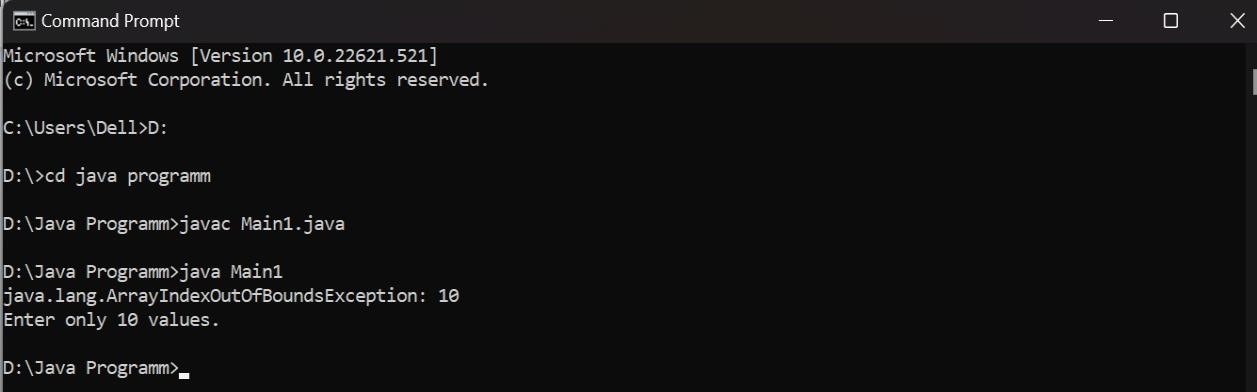
System.out.println("Enter only 10 values.");

}

}

}

**Output:**



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

**Code:**

import java.util.\*;

import java.io.\*; class Error

{

int m(char a,char b)

{

return

1\*(Integer.parseInt(Character.toString(a))%Integer.parseInt(Character.toString(b)));

}

int n(char a, char b)

{

return m(a,b);

}

}

class ErrorThrows extends Error

{

public static void main(String args[])

{

Error obj=new Error(); try{

obj.n(args[0].charAt(0),args[0].charAt(1));

}

catch(NumberFormatException e)

{

System.out.println(e);

}

catch(Exception e)

{

System.out.println(e);

}

finally{

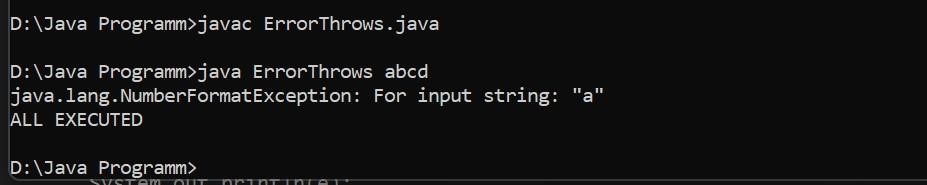
System.out.println("ALL EXECUTED");

}//finally

}//main

}//class

**Output:**



3. Write a code to create your own exception class. Create another class, inside main method prompt user to enter a number if number is less than 500 throw instances of your customexception class.

**Code:** class CostomException extends Exception{ CostomException(String msg){ super(msg);

} }

import java.util.\*; class TestB{ public static void main(String args[]){ Scanner sc =new Scanner(System.in); int n= sc.nextInt(); try{

if(n<500)

throw new CostomException("Number is less than 500");

else

System.out.println("No exception occured");

}

catch(CostomException e){

System.out.println(e);

} finally{

sc.close();

}

}

}

**Output:**



4. You are given two integers, a and b as input, you have to compute a/b: If a and b are not bit signed integers or if is zero, exception will occur and you have to report it. Read sample Input/Output to know what to report in case of exceptions.

**Code:**

import java.util.\*; class Mismatch

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in); try {

int num1 = sc.nextInt(); int num2 = sc.nextInt();

System.out.println(num1/num2);

}

catch(InputMismatchException e)

{

System.out.println(e);

}

catch(ArithmeticException e)

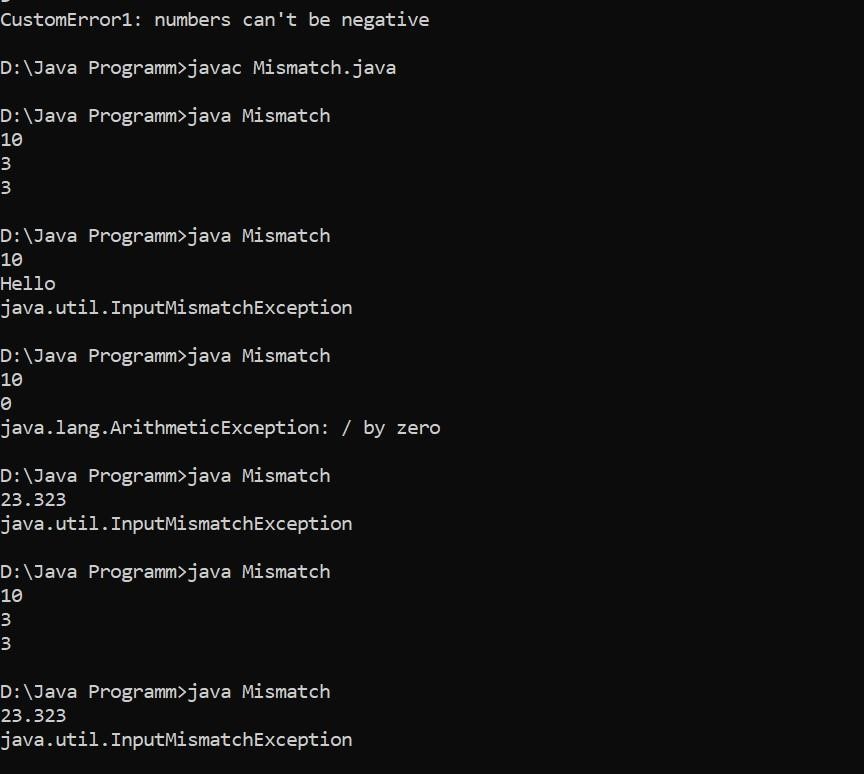
{

System.out.println(e); }

}//main

}//class

**Output:**



5. You are required to compute the power of a number by implementing a calculator. Create a class Calc which consists of a single method long power(int, int). This method takes two integers, a and b, as parameters and finds ab. If either a or b is negative, then the method must throw an exception which says "a and b should not be negative". Also, if both a and b are zero, then the method must throw an exception which says "a and b should not be zero".

**Code:**

import java.util.\*; class CustomError1 extends Exception

{

CustomError1(String message)

{

super(message);

}

} class CalculatorPower

{

static long power(int a,int b) throws CustomError1

{

if(a<0 || b<0){

String message="numbers can't be negative"; throw new CustomError1("numbers can't be negative");

//return 0; } if(a==0){ throw new CustomError1("numbers can't be Zero");

}

return (long)Math.pow(a,b);

}

static void constranits(int a,int b)

{

if(a<-10 || a>10 || b<-10 || b>10){

System.out.println("Follow Constranits");

System.exit(0);

}

}

public static void main(String args[]){ try{

Scanner sc=new Scanner(System.in); System.out.println("Enter a and b:-"); int a=sc.nextInt(); int b=sc.nextInt(); constranits(a,b); long ans=power(a,b);

System.out.println("Answer :- "+ans);

}

catch(CustomError1 E)

{

System.out.println(E);

}

catch(Exception E)

{

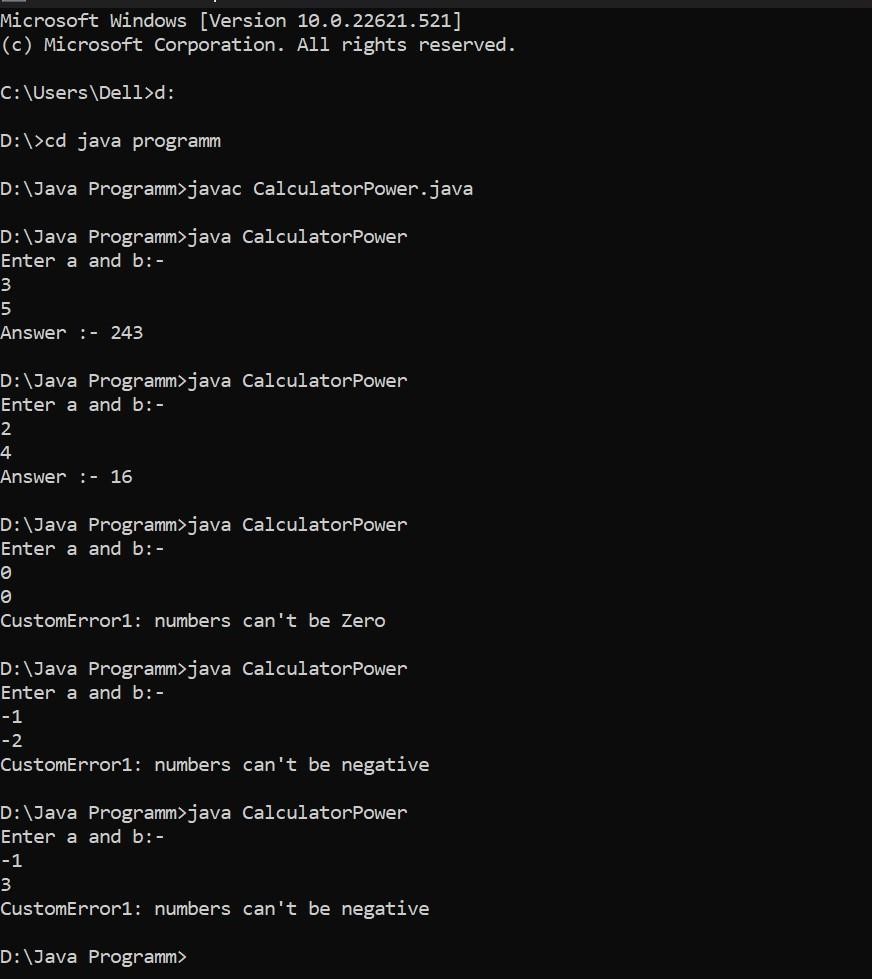
System.out.println(E);

}

}//main

}//class

**Output:**



## Experiment-8 (String and Wrapper classes)

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

**Code:**

import java.util.\*; class Search {

public static void main(String args[])

{

int last=0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string : ");

String s1 = sc.nextLine();

System.out.println("Enter the string to be searched : "); String s2 = sc.nextLine(); int len1 =s1.length(); int len2 =s2.length(); for (int i=0;i<=(len1-len2);i++)

{

if(s1.substring(i,len2+i).equals(s2))

{ if(last==0)

System.out.println("First occurrence is at position : "+(i+1));

last=i+1;

}//if

}//for

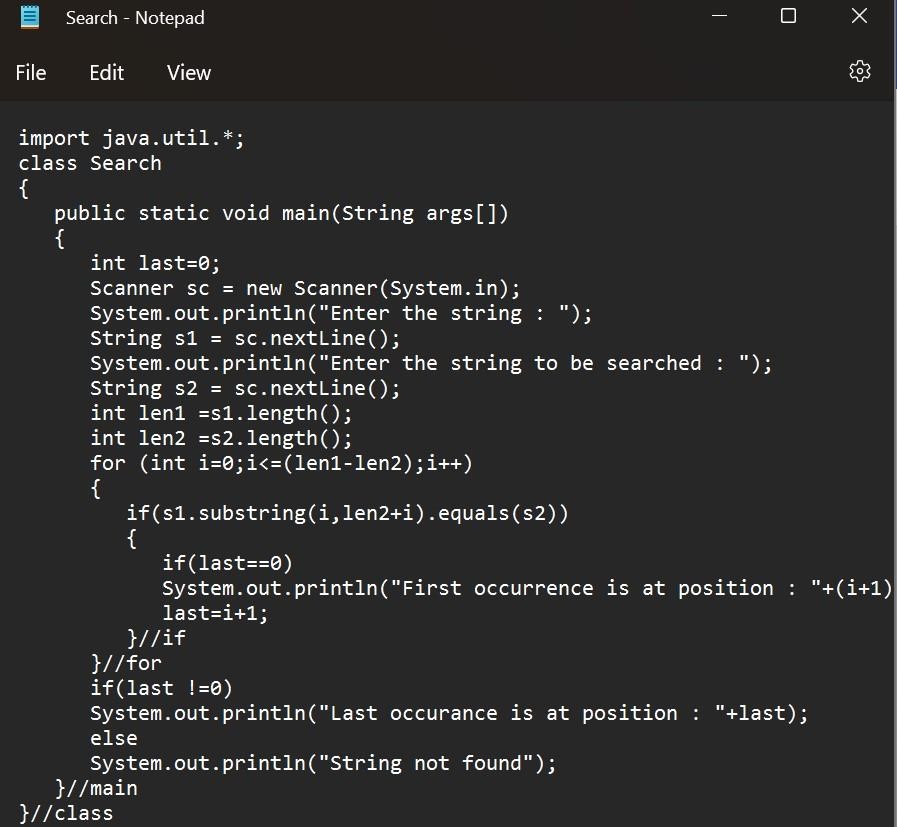
if(last !=0)

System.out.println("Last occurance is at position : "+last); else

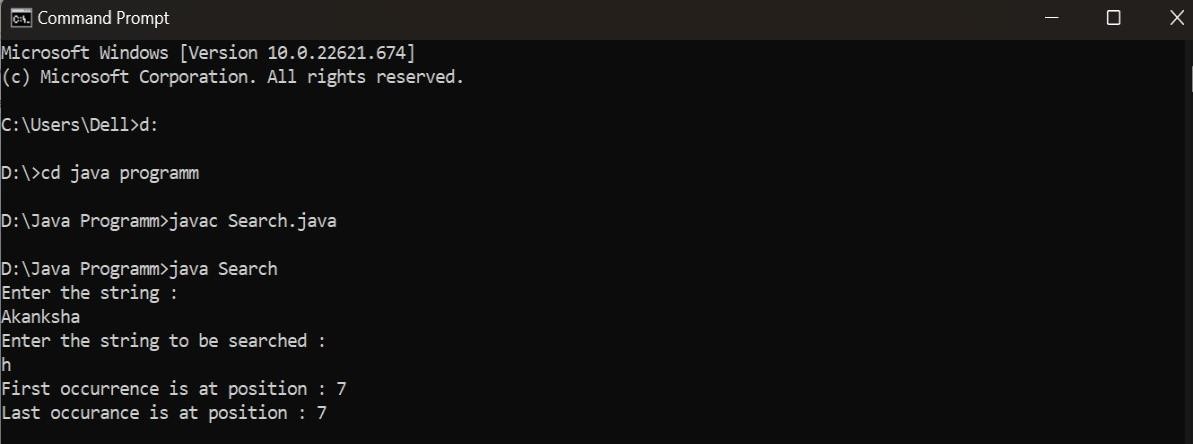
System.out.println("String not found");

}//main

}//class



**Output:**



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

**Code:**

import java.util.\*; class StringName { public static void main(String args[]) {

StringBuffer s = new StringBuffer("");

Scanner sc = new Scanner(System.in); System.out.println("Enter the word : "); s.append(sc.nextLine()); String

w="";

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

{

char ch = s.charAt(i);

int x = ch; x = x-

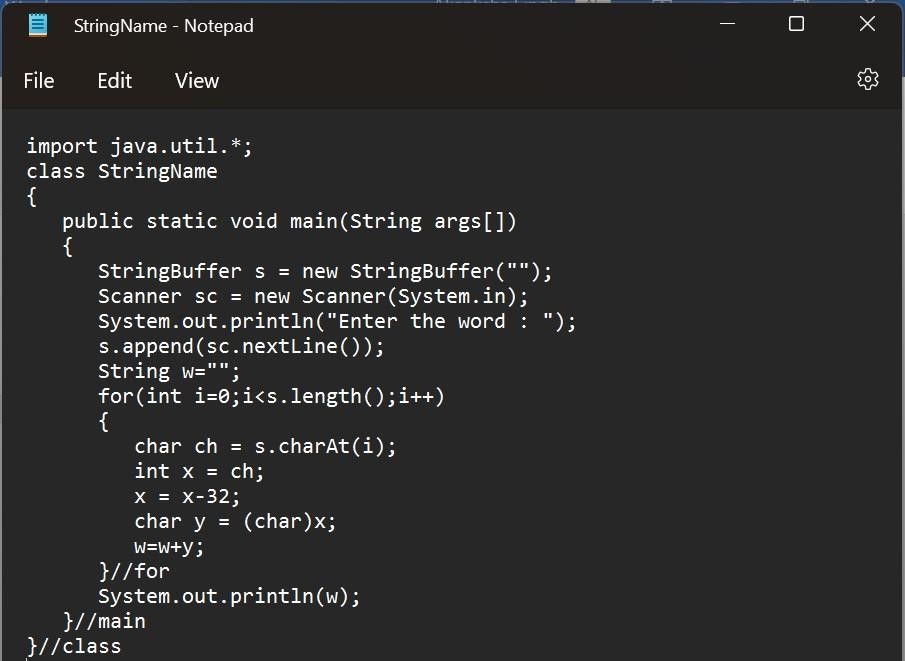
32; char y = (char)x;

w=w+y; }//for

System.out.println(w);

}//main

}//class



**Output:**



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

**Code:**

import java.util.\*; public

class StringName1

{ public static void main(String args[])

{

StringBuffer s = new StringBuffer("");

Scanner sc = new Scanner(System.in);

System.out.println("Enter the name of the string : "); s.append(sc.nextLine());

s.append(" "); String a = ""; String b = ""; for(int

i=0;i<s.length();i++){

char ch =

s.charAt(i); if(ch != ' '){

int o = ch; o = o-32;

char y =(char)o; a = a+y;

}//if else{

b = b+a; b = b+" "; a="";

}//else

}//for

System.out.println(b);

}

//main

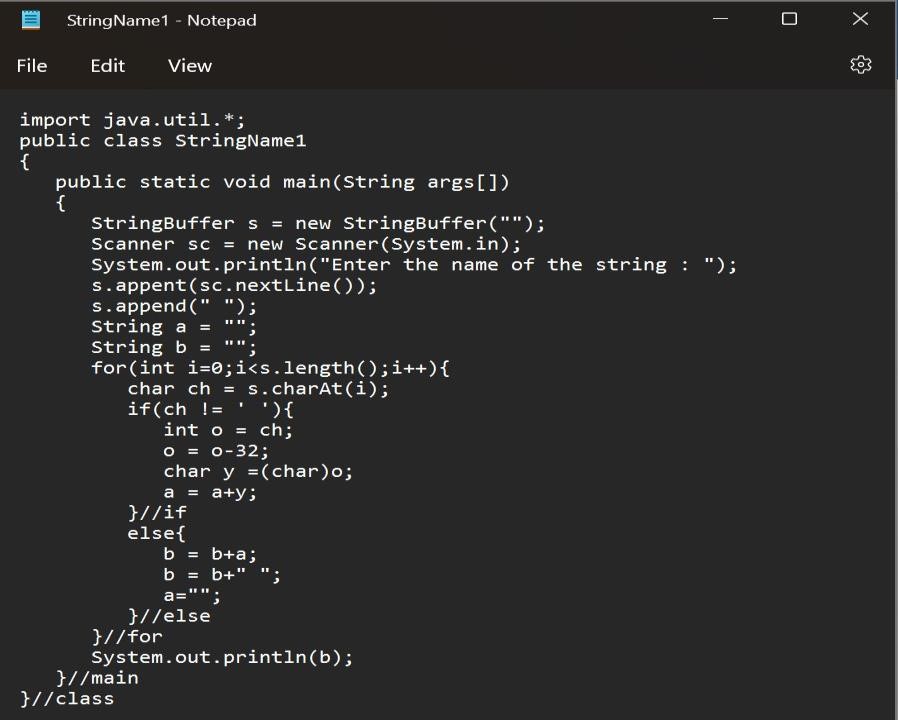
**Output:**



//class



}



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.

**Code:**

import java.util.\*; public

class StringName2

{

public static void main(String args[]){

String str = "akanksha Singh";

System.out.println("Length of the string "+str+" : "+str.length()); char c[] = str.toCharArray(); int count = str.indexOf('a'); if(count == -1)

System.out.println("a is not present."); else{

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

if('a'==c[i])

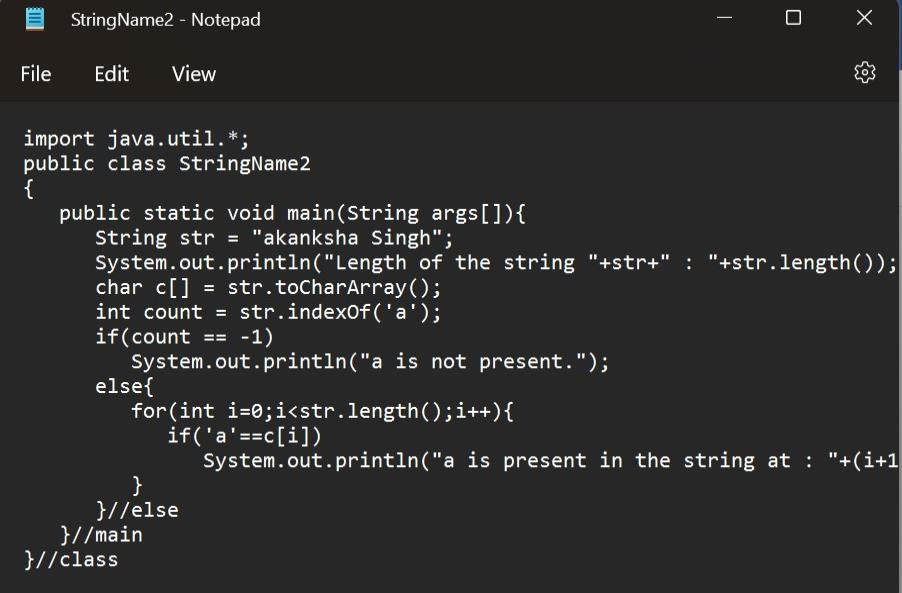
System.out.println("a is present in the string at : "+(i+1));

}

}//else

}//main

}//class



**Output:**



5. 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.

**Code:**  class StringWrapper1

{ public static void main(String args[])

{

int i = 22;

@SuppressWarnings("removal")

Integer intm = new Integer(i);

System.out.println(intm);

String s = Integer.toString(567); System.out.println(s); int o = Integer.parseInt("986");

System.out.println(0);

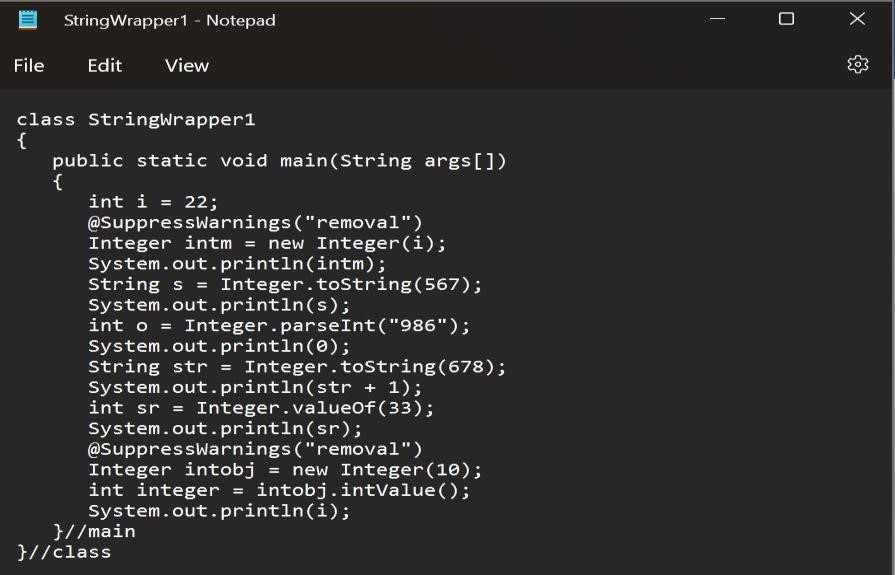
String str = Integer.toString(678); System.out.println(str + 1); int sr = Integer.valueOf(33); System.out.println(sr);

@SuppressWarnings("removal") Integer intobj = new Integer(10); int integer = intobj.intValue();

System.out.println(i);

}//main

}//class



**Output:**



6. 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.

**Code:**  public class StringWrapper2

{ public static void main(String args[])

{

float f = 13.1f;

@SuppressWarnings("removal") float f1 = new Float(f); System.out.println(f1);

String s = Float.toString(12.2F);

System.out.println(s);

String str = String.valueOf(45.2); System.out.println(str); float

flo = Float.parseFloat("23.6");

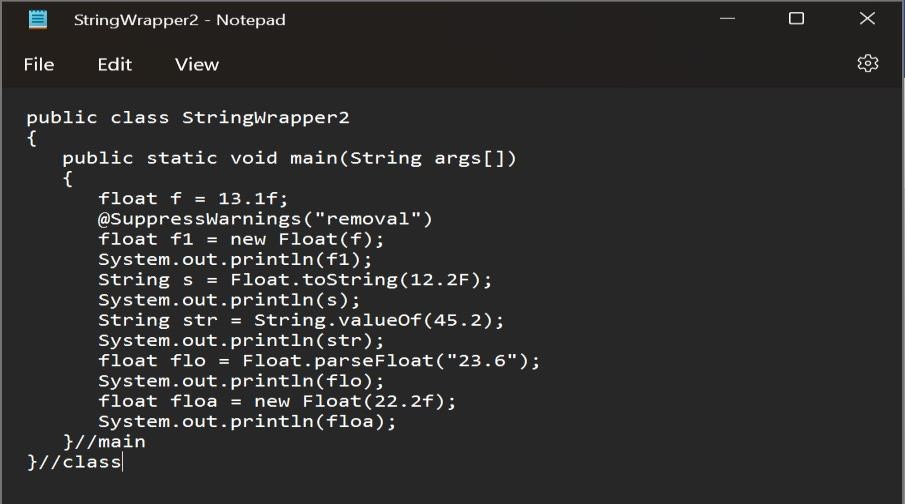
System.out.println(flo); float

floa = new Float(22.2f);

System.out.println(floa);

}//main

}//class



**Output:**



**Experiment-9(Threads and Collections)**

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

Code:

class MyThread1 extends Thread

{

public void run()

{

System.out.println("Thread1 is running.");

}//run

}//class

class MyThread2 implements Runnable

{

public void run()

{

System.out.println("Thread2 is running.");

}//run

}//class

class DemoA

{

public static void main(String args[])

{

MyThread1 obj1 = new MyThread1();

obj1.start();

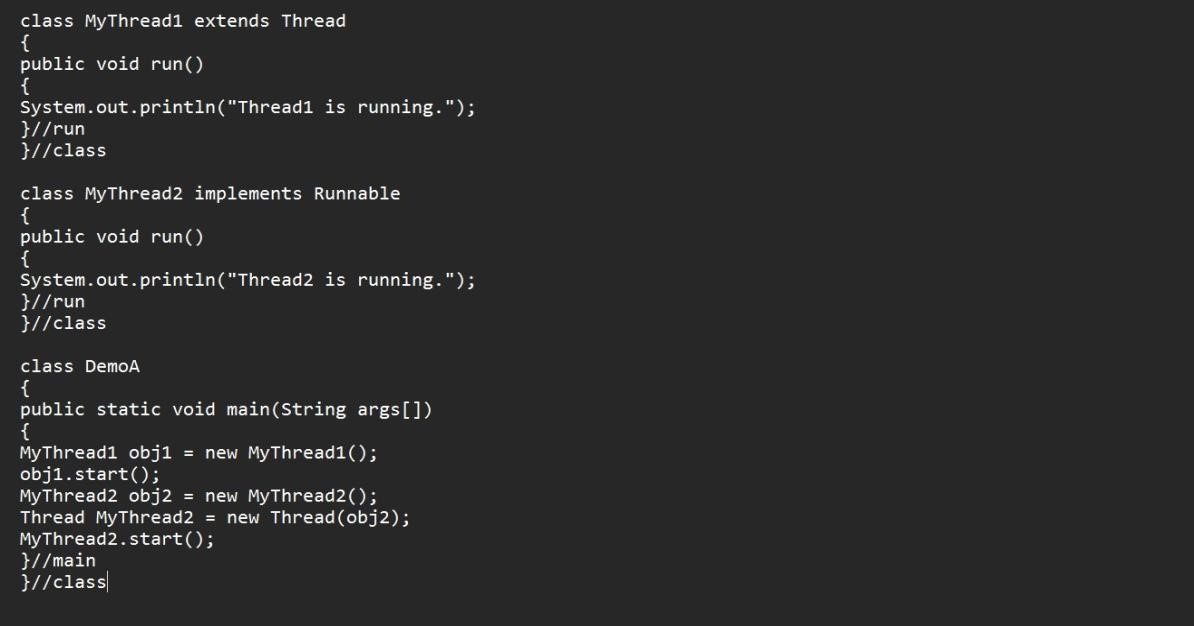
MyThread2 obj2 = new MyThread2();

Thread MyThread2 = new Thread(obj2);

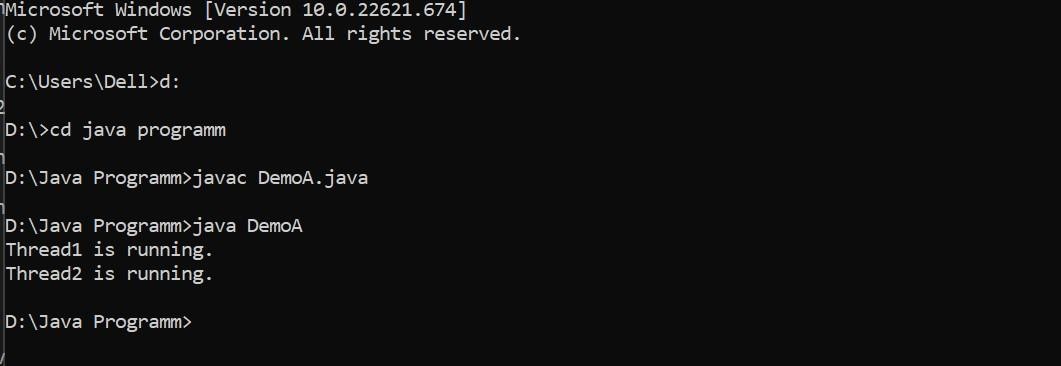
MyThread2.start();

}//main

}//class



Output:



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

Code:

import java.io.\*; class

ThreadEven extends Thread

{

public void run()

{

for(int i=0; i<=20;i++)

{

if(i%2==0){

System.out.println(i);

}//if

}//for

}//run

}//class

class ThreadOdd extends Thread

{

public void run()

{

for(int i=0; i<=20;i++)

{

if(i%2==0){

System.out.println(i+1);

}//if

}//for

}//run

}//class

class DemoB

{

public static void main(String args[])

{

ThreadEven obj1 = new ThreadEven(); obj1.start(); try{

Thread.sleep(500);

}

catch(Exception e)

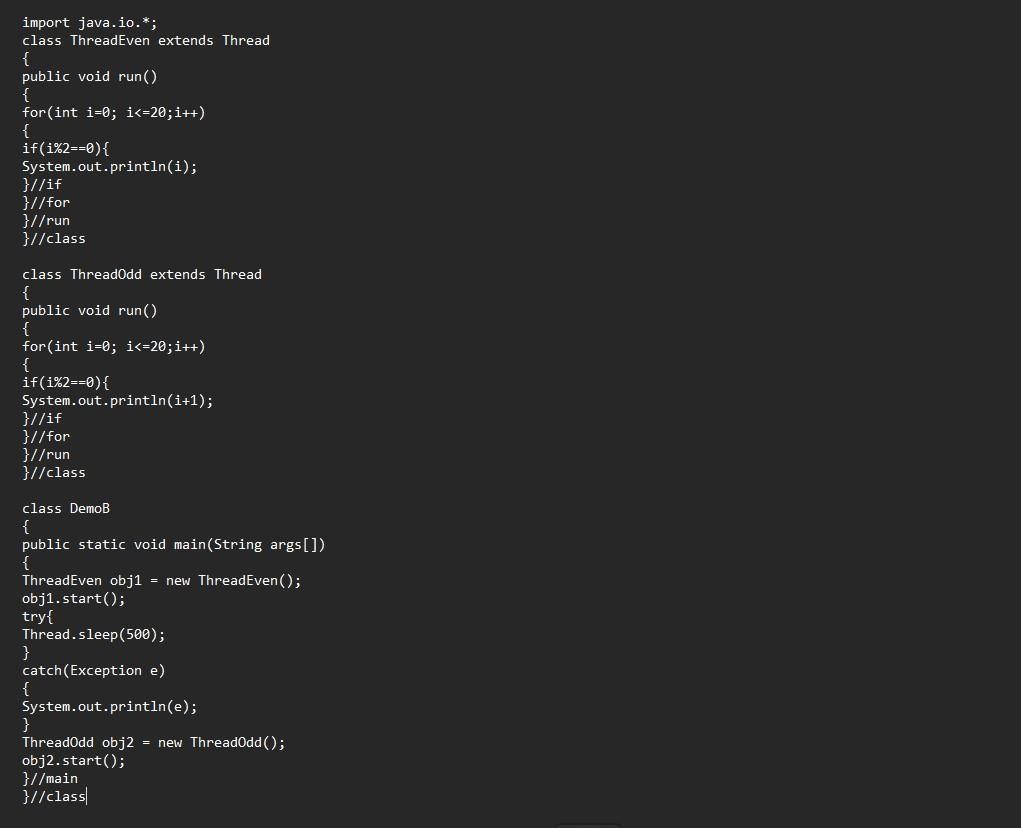
{

System.out.println(e);

}

ThreadOdd obj2 = new ThreadOdd(); obj2.start(); }//main

}//class



Output:



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

Code: class ThreatA extends

Thread

{

static int counter=0; int val=0; public void run()

{

if (currentThread().getPriority()==Thread.MIN\_PRIORITY)

{ test();

System.out.println("The count variable is ="+counter);

System.exit(0);

}//if try{

Thread.sleep(0);

}

catch(Exception e)

{System.out.println(e);

} test();

val++; }//run synchronized int test()

{

counter++; return counter; }//test

}//class

class Threadex2{ public static void main(String args[])

{

for(int i=0;i<10;i++){

ThreatA t1=new ThreatA();t1.setPriority(10-i); t1.start();

} try{

Thread.sleep(1000);

}

catch(Exception e)

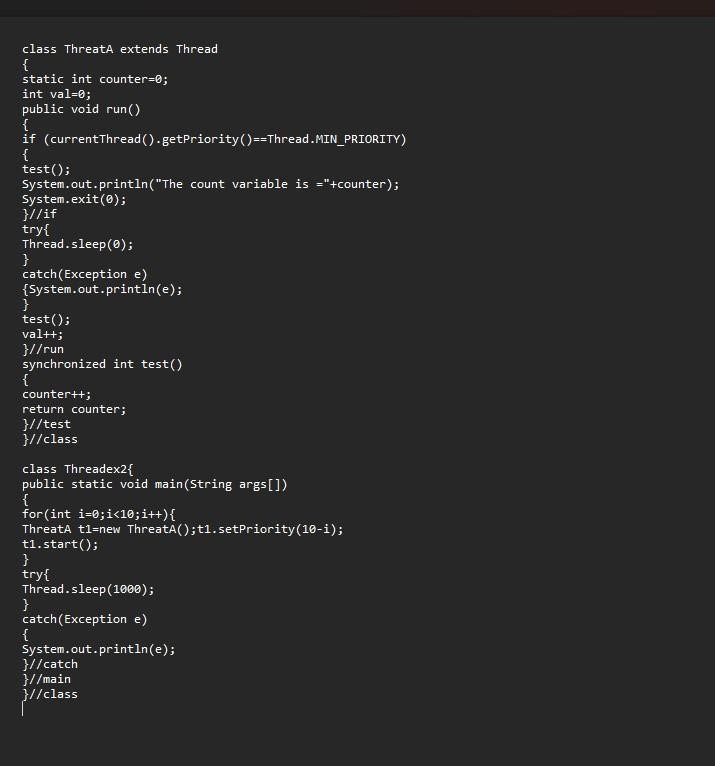
{

System.out.println(e);

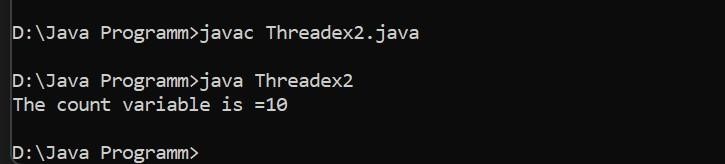
}//catch

}//main

}//class



Output:



4. Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting.

Code: import java.lang.Thread; import java.lang.\*; public class

DemoC extends Thread

{

public static void main(String args[]) throws InterruptedException

{

Thread T1 = new Thread();

Thread T2 = new Thread();

Thread T3 = new Thread();

Thread T4 = new Thread();

Thread T5 = new Thread();

T1.setPriority(6);

T2.setPriority(1);

T3.setPriority(9);

T4.setPriority(10);

T5.setPriority(4); T1.sleep(1500); if(T1.isAlive())

System.out.println("Thread 1 is alive"); else

System.out.println("Thread 1 is not alive");

T2.start(); if(T2.isAlive())

System.out.println("Thread 2 is alive"); else

System.out.println("Thread 2 is not alive"); T3.sleep(1500); if(T3.isAlive())

System.out.println("Thread 3 is alive"); else

System.out.println("Thread 3 is not alive");

T4.start(); if (T4.isAlive())

System.out.println("Thread 4 is alive"); else

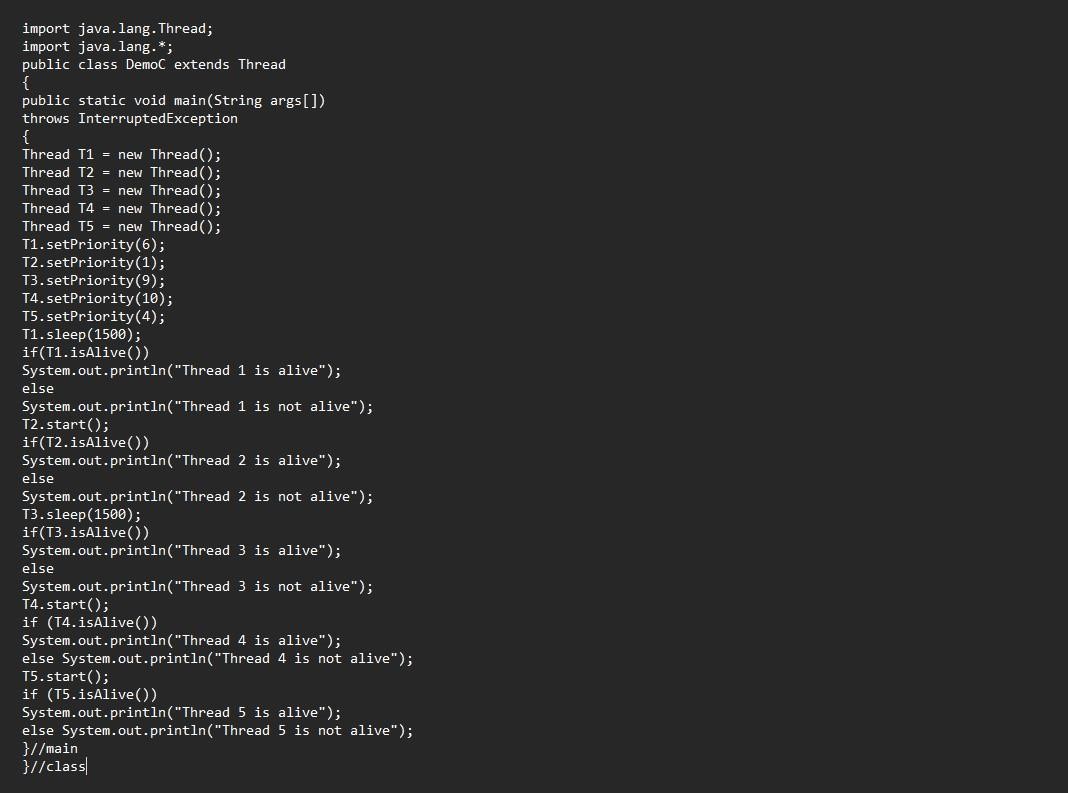
System.out.println("Thread 4 is not alive"); T5.start(); if (T5.isAlive())

System.out.println("Thread 5 is alive"); else

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

}//main

}//class



Output:



## Experiment-10 (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.

Code: import java.util.\*;

@SuppressWarnings("unchecked") class

Collections1

{

public static void main(String args[])

{

ArrayList<Integer> list=new ArrayList<Integer>();//Creating arraylist list.add(2); list.add(3); list.add(4); list.add(5);

ArrayList<Integer> listd=(ArrayList<Integer>)list.clone();//duplicate list

Iterator itr = listd.iterator();//Traversing listd through Iteration

System.out.println("ORDER:-"); while(itr.hasNext())

{

System.out.println(itr.next()); //read the list

}

System.out.println("Reverse = "); for

(int i= list.size()-1;i>=0;i--)

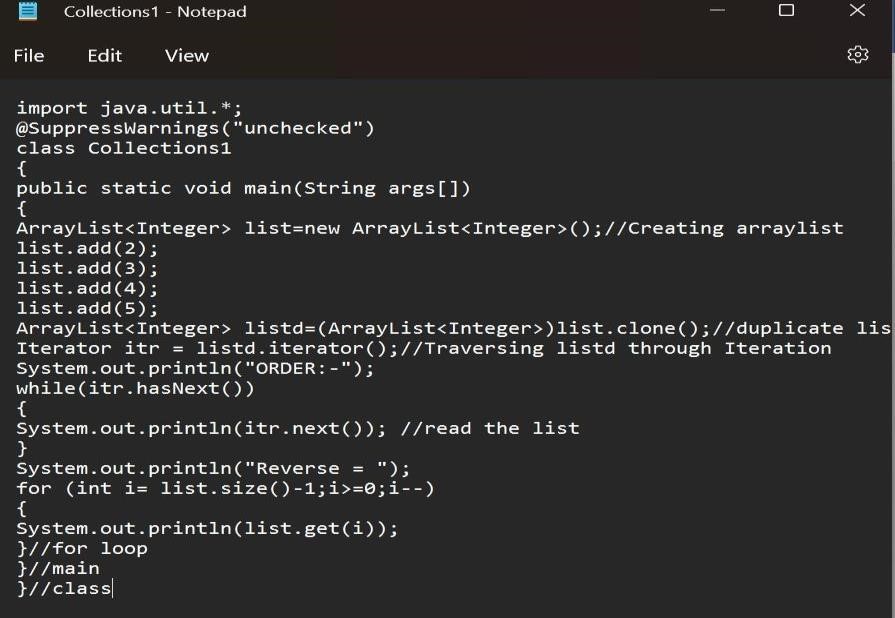
{

System.out.println(list.get(i));

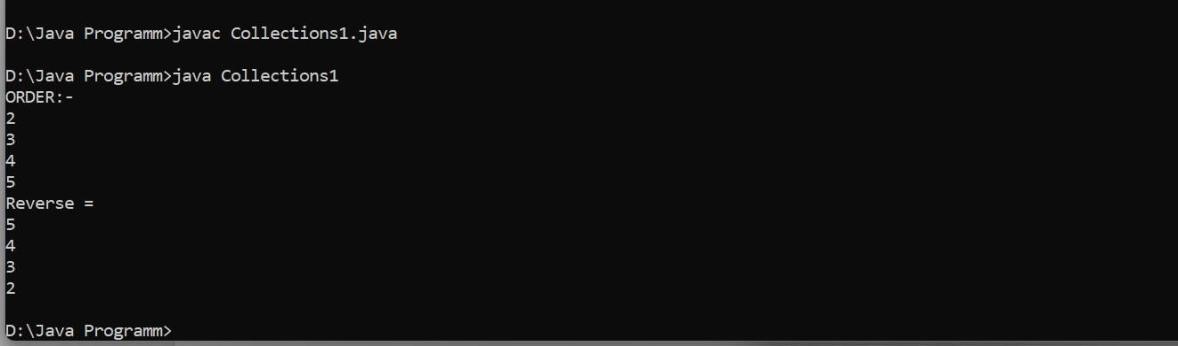
}//for loop

}//main

}//class



Output:



2. 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 pairs as entry objects

Code: import java.util.\*;

@SuppressWarnings("unchecked")

class Collections2

{

public static void main(String args[])

{

HashMap<Integer, String> hm= new HashMap<Integer, String>();//<key,value> hm.put(10,"A"); hm.put(11,"B"); hm.put(12,"C");

System.out.println(hm);

System.out.println("Key Check 1 = ");

Scanner sc = new Scanner(System.in);

System.out.println("Enter key"); int n = sc.nextInt(); if (hm.containsKey(n))

System.out.println("Present"); else

System.out.println("Not Present");

System.out.println("Key Check 2 = "); System.out.println("Enter value");

String s =sc.next(); if(hm.containsValue(s))

System.out.println("Present"); else

System.out.println("Not Present");

System.out.println("Iteration hashmap......"); for(Integer m:hm.keySet())

System.out.println(m);

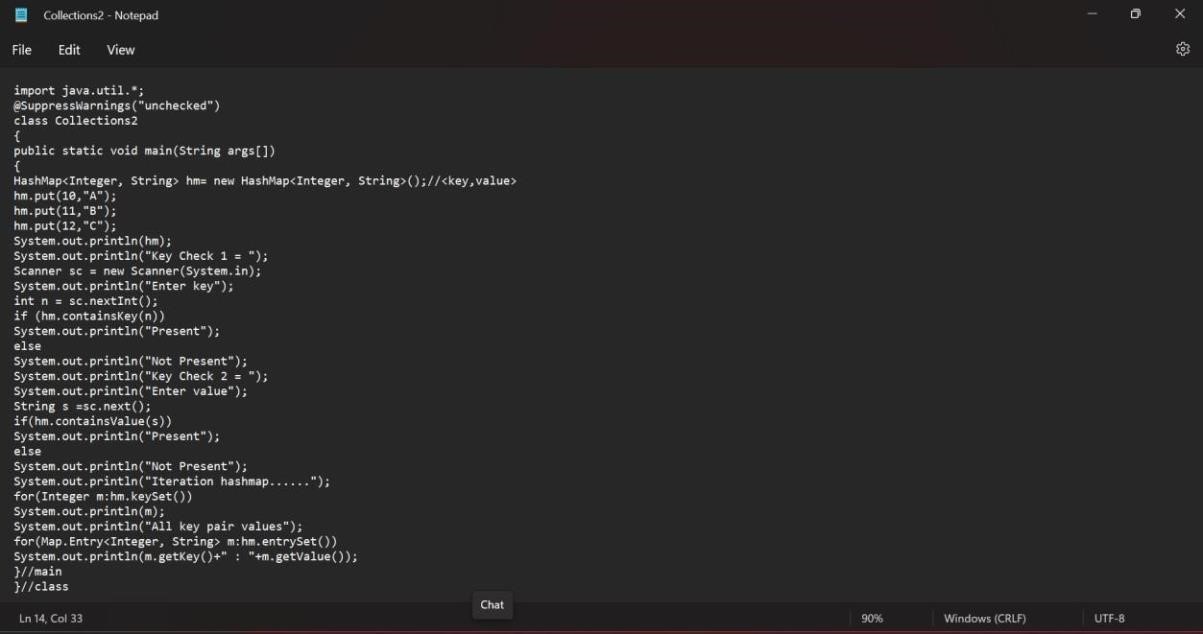
System.out.println("All key pair values"); for(Map.Entry<Integer,

String> m:hm.entrySet())

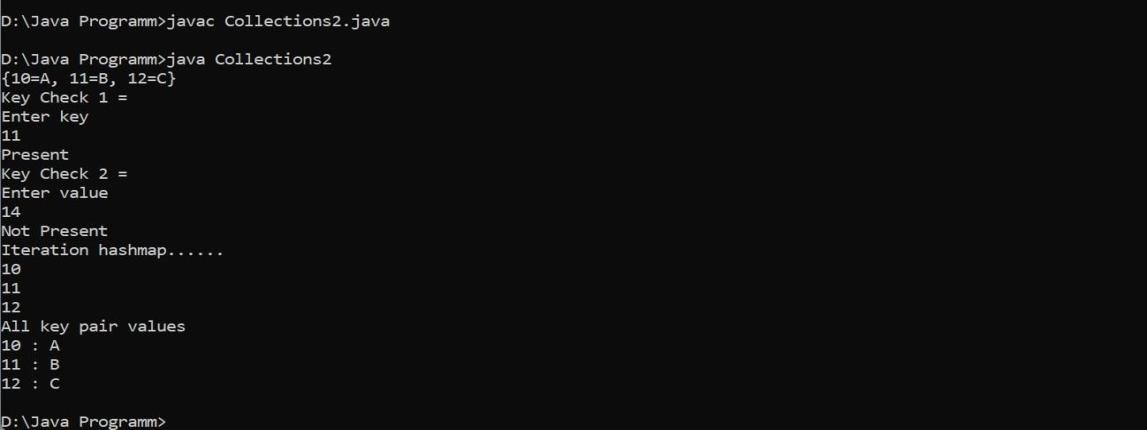
System.out.println(m.getKey()+" : "+m.getValue());

}//main

}//class



Output:



3. Write a program for the following HashSet

* copy another collection object to HashSet object.
* delete all entities at one call from HashSet
* search user defined objects from HashSet Code:

import java.util.\*;

@SuppressWarnings("unchecked") class

Collections3

{

public static void main(String args[])

{

HashSet<Integer> hs=new HashSet<Integer>(); hs.add(1); hs.add(2); hs.add(3); hs.add(4); System.out.println("Original HS = "); hs.add(1);//not added as its a duplicate System.out.println(hs);

System.out.println("Copied HS = "); HashSet<Integer> hs1

= new HashSet<Integer>(); hs1.addAll(hs);

System.out.println(hs1);

Scanner sc = new Scanner(System.in); System.out.println("Enter element ="); int n = sc.nextInt(); if(hs.contains(n)) System.out.println("Present");

else

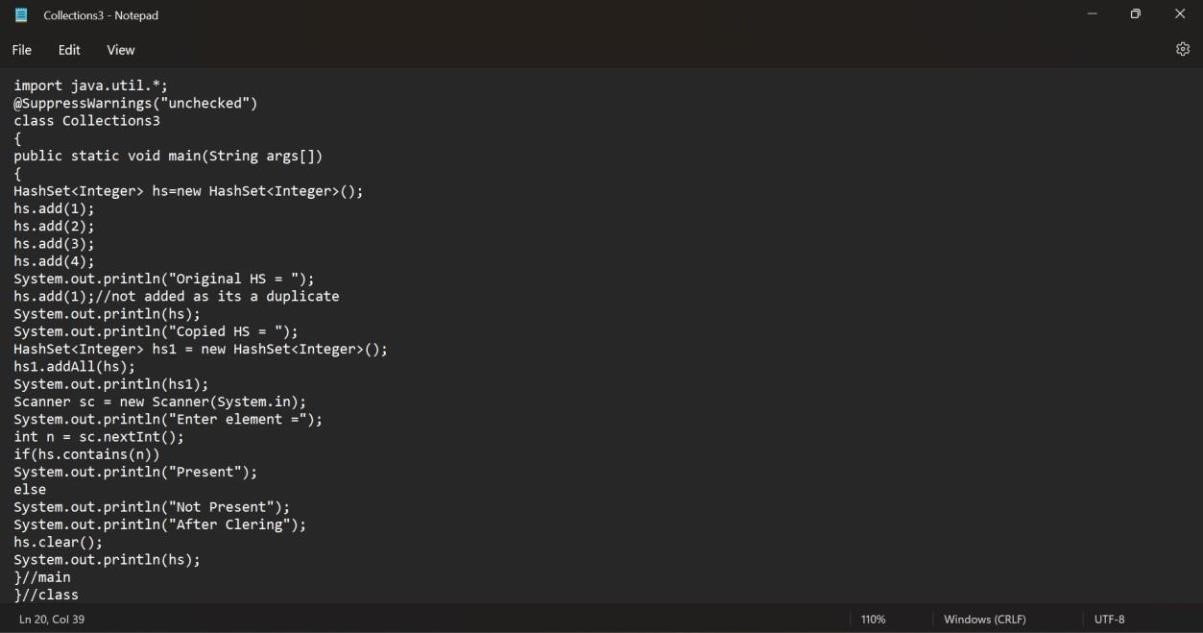
System.out.println("Not Present");

System.out.println("After Clering"); hs.clear();

System.out.println(hs);

}//main

}//class



Output:



Experiment-11(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 tocreate a table. Code:

package lex; import

java.sql.\*; public class Test {

public static void main(String[] args) { try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection conn =

DriverManager.getConnection("jdbc:mysql://localhost:3306/company","LEX","152001"); Statement stmt = conn.createStatement(); int n=stmt.executeUpdate("create table employee(name varchar(255),id int,email

varchar(255),number long)");

String Data="insert into employee(" + "name,id,email,number) values"+

"('adi',001,'lec@gmail.com',123456789),"+"('karma',002,'karma2gmail.com',4568791);"; int d=stmt.executeUpdate(Data);

System.out.println("no of rows updateed"+d); conn.close();

}

catch(ClassNotFoundException e) {

System.out.print(e);

}

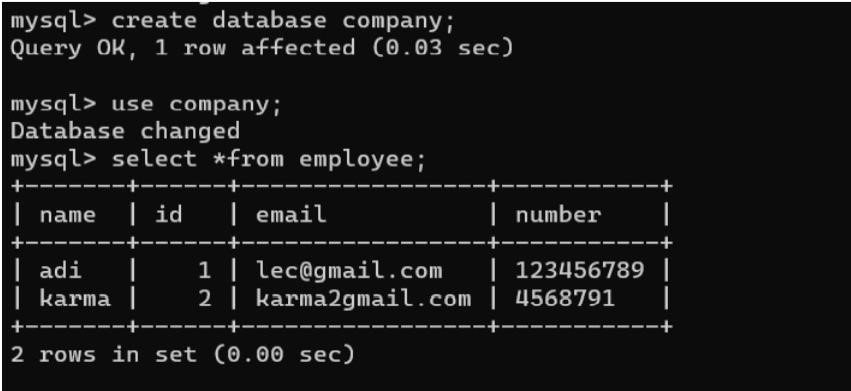
catch(SQLException e) {

System.out.println(e);

}

}

}



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

Code:

package lex; import

java.sql.\*; public class Testsym {

public static void main(String[] args) { try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection conn = DriverManager.getConnection

("jdbc:mysql://localhost:3306/sym","LEX","152001");

Statement stmt = conn.createStatement();

String Data="insert into employee(id,name,salary) values"+

"(001,'adi',20000),"+"(002,'karma',10000);"; int d=stmt.executeUpdate(Data);

System.out.println("no of rows updateed"+d); conn.close();

}

catch(ClassNotFoundException e) {

System.out.print(e);

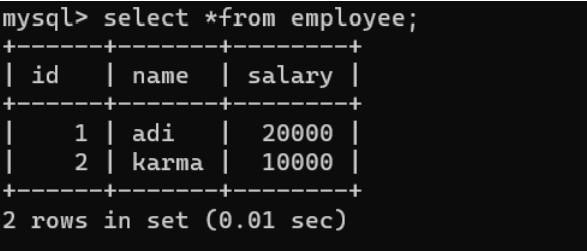
}

catch(SQLException e) { System.out.println(e);

}

}

}



3. Create a table Meeting having columns (NameOfParticipant, MeetingID,

ScheduledTime,Email, Mobile), Populate with random data and perform following operations. READCOMPLETE QUESTION TO POPULATE DATABASE.

* Using JaveCode determine the names of the columns in the MEETING table

anddisplay the column names on the console. o Write a query to find the names of all participants for the meeting with ID

1144.Display the names on the console, preceded by the message, “Names of participantsin meeting 1144”.

* Insert a new participant with using only the participant’s name. Don’t specify

theparticipant ID - let the ID be determined by the database. Determine the ID that wasassigned to the new participant and display it on the console (the statement thatdisplays the ID should display the name of the new participant as well).

* Count the number of people participating in the meeting with ID 1105. Display

amessage on the console that gives both the meeting name and the number ofparticipants. o Determine the names of all people who participate in meetings that meet on Tuesdays.Display the names, preceded by the message, “Participants attending Tuesdaymeetings”.

Code:

package lex; import

java.sql.\*;

public class Employee { public static void main(String[] args) { try { Class.forName("com.mysql.cj.jdbc.Driver");

Connection conn = DriverManager.getConnection

("jdbc:mysql://localhost:3306/test","LEX","152001");

Statement stmt = conn.createStatement();

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

ResultSetMetaData rsmd = rs.getMetaData(); int count = rsmd.getColumnCount(); System.out.println("List of column names in the current table: "); for (int i =

1;i<=rsmd.getColumnCount(); i++) {

System.out.println(rsmd.getColumnName(i)+"\t");

}

while (rs.next()) { if(rs.getInt(2)==1144) System.out.println("Names of participantsin meeting with 1144 :"+rs.getString(1));

}

String Data="insert into meeting(NameOfParticipant,ScheduledTime,email,mobile) values"+

"('teddy','2022-11-15','srgd468@gmail.com',165685);"; int d=stmt.executeUpdate(Data); ResultSet rs1=stmt.executeQuery("select\*from meeting"); while

(rs1.next()) { if(rs1.getInt(5)==165685) System.out.println("meeting id and name :"+rs1.getString(1)+" "+rs1.getInt(2));

}

ResultSet rs2=stmt.executeQuery("select\*from meeting"); while

(rs2.next()) { if(rs2.getInt(2)==1105) System.out.println("no and name of people with id 1105"+rs2.getString(1)+" "+rs2.getInt(2));

}

ResultSet rs3=stmt.executeQuery("select\*from meeting"); while

(rs3.next()) { if(rs3.getString(3)=="2022-11-15");

System.out.println("no and name of people with id 1105"+rs3.getString(1));

} conn.close();

}

catch(ClassNotFoundException e) {

System.out.print(e);

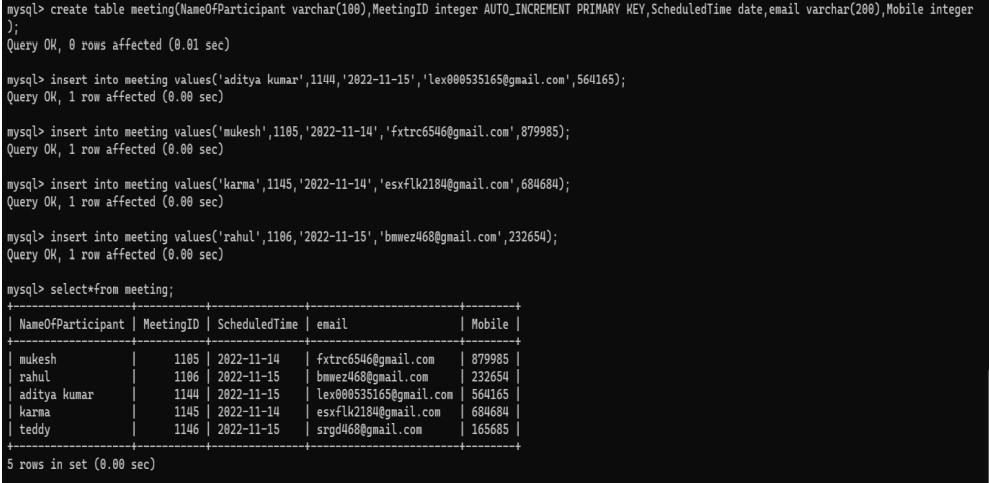
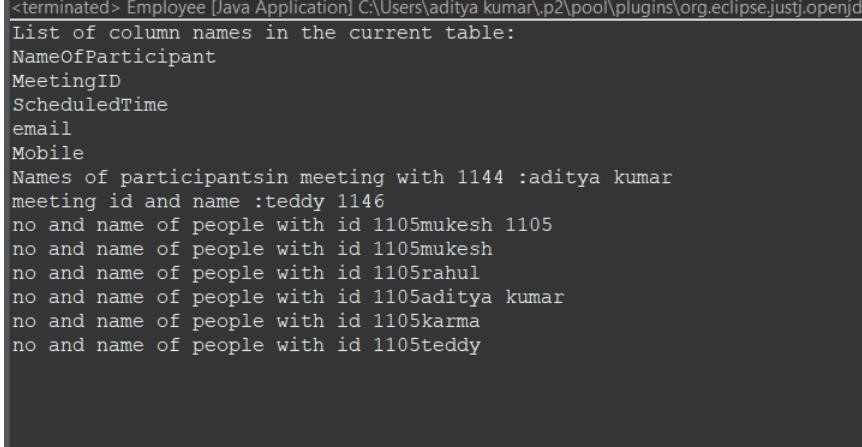
}

catch(SQLException e) { System.out.println(e);

}

}

}



## Experiment-12(Servlet)

1) Servlet:

a) ServletContext interface Code:

// Using Servlet Inteface import java.io.\*; import javax.servlet.\*;

import javax.servlet.http.HttpServlet; public class demoservlet extends HttpServlet implements Servlet { private static final long serialVersionUID = -2399671137476004288L; ServletConfig config=null;

public void init(ServletConfig config)throws ServletException

{

this.config=config;

}

public void service (ServletRequest request,ServletResponse response)throws ServletException,IOException

{

response.setContentType("text/html"); PrintWriter out=response.getWriter(); out.print("Welcome here[Inteface Servlet]");

out.close(); } //service

public void destroy()

{

System.out.println("The servlet is destroyed");

}

public ServletConfig getServletConfig()

{

return (ServletConfig)config;

} public String getServletInfo()

{

return "" ; } }//class



b)getParameterValues( ) of Servlet Request Code:

package practice\_servlet;, import java.io.IOException; import

java.io.PrintWriter; import javax.servlet.ServletException; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; public class add extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) { int a=Integer.parseInt(request.getParameter("no1")); int b=Integer.parseInt(request.getParameter("no2")); out.println("Addition is: "+(a+b));

}

}

@Override protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

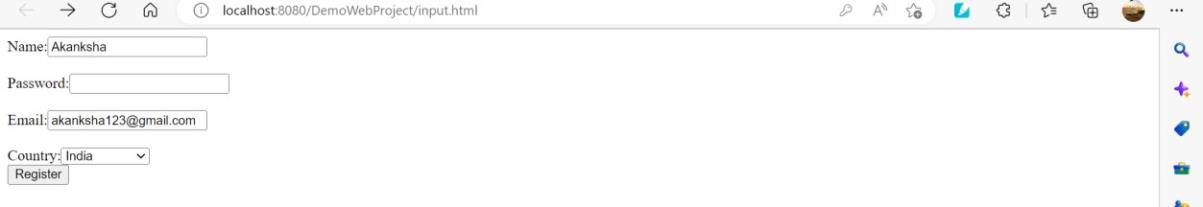
{



processRequest(request, response);



}



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

Code:

//printing the current date of server import javax.servlet.ServletException; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import java.io.IOException; import java.io.PrintWriter; import java.time.LocalDateTime; import java.time.format.DateTimeFormatter;

public class timecheck extends HttpServlet

{

LocalDateTime time=LocalDateTime.now();

DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss"); private static final long serialVersionUID = 1L;

public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

response.setContentType("text/html");

PrintWriter out=response.getWriter(); out.print("Welcome Here Today, the time is :- "); out.print(" "+time+" "+"\tOR\t"+" "); out.print(dtf.format(time)); out.print(" "); out.close(); }//doget

}//class



3) Write a Servlet page to which include the two other Servlet page through of include directivesfeature provided in Servlet.

Code: package new1;, import java.io.IOException;

import java.io.PrintWriter; import javax.servlet.RequestDispatcher; import javax.servlet.ServletException; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;

public class add extends HttpServlet { protected void processRequest(HttpServletRequest

request, HttpServletResponse response) throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8"); try (PrintWriter out = response.getWriter())

{

int a=Integer.parseInt(request.getParameter("no1"));

int b=Integer.parseInt(request.getParameter("no2")); int c=(a+b); if(c%2==0){ out.println("Addition is: "+(a+b)); } else{ RequestDispatcher

rd=request.getRequestDispatcher("servlet2"); rd.forward(request, response);

}

}

} Servlet2:

public class NewServlet extends HttpServlet { protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8"); PrintWriter out = response.getWriter(); out.println("Addition is odd");

}

HTML file

<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>



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

Code:

Package Test;, 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 calculator extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

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((Integer.parseInt(n1) + Integer.parseInt(n2)));

}

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

{ out.println(Integer.parseInt(n1) - Integer.parseInt(n2));

} else if(op.equals("multiplication"))

{ out.println(Integer.parseInt(n1) \* Integer.parseInt(n2));

} else {

out.println(Integer.parseInt(n1) / Integer.parseInt(n2));

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

processRequest(request, response);

}

@Override public String getServletInfo()

{ return "Short description";

}

// </editor-fold>

}

HTML FILE

<html>

<head>

<title>calculator</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h1 style="border:2px solid Tomato;">4 function calculator</h1>

<form action="calculator" method="get" name="frm">

Enter first number: <input name="txt1" type="text" /><br> <br>

Enter second number: <input name="txt2" type="text" /><br> <br>

Operator <select name="op"> <option value="Addition">Addition</option>

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

<option value="multiplication">multiplication</option>

<option value="division">division</option>

</select>

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

</form>

</body>

</html>

