NAME-TASMIYA FATHIMA

USN-1BM19CS172

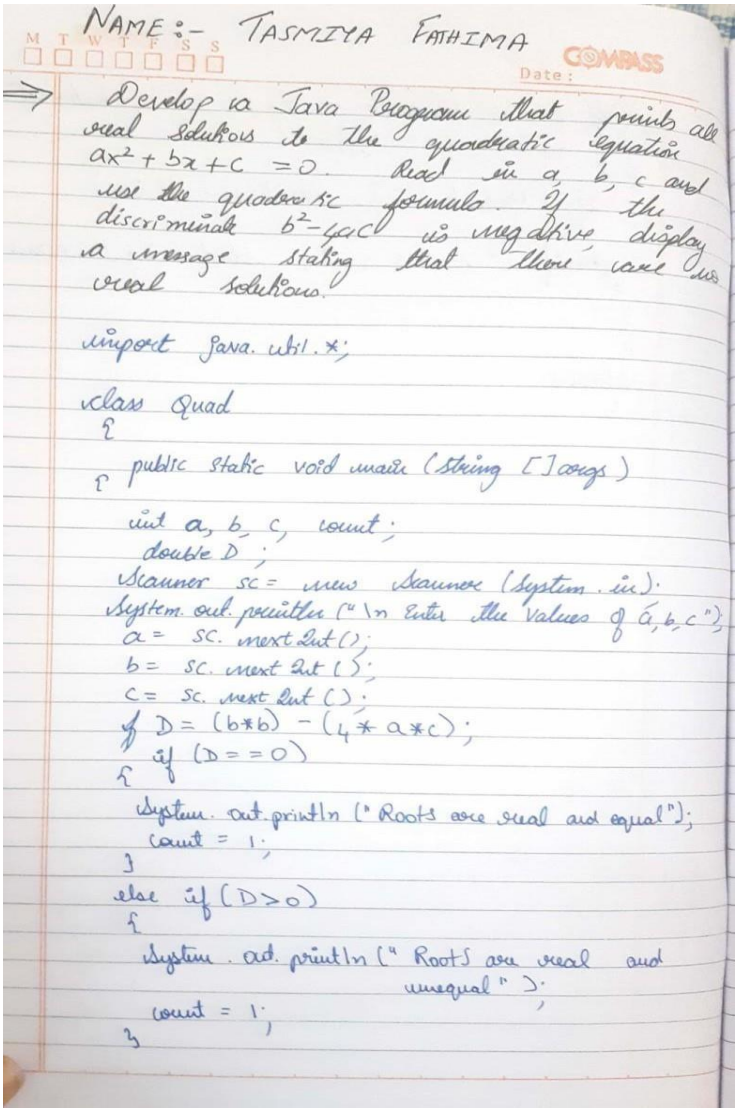
Lab Program: 1

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0.

Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

import java.util.*;

WRITE UP :-

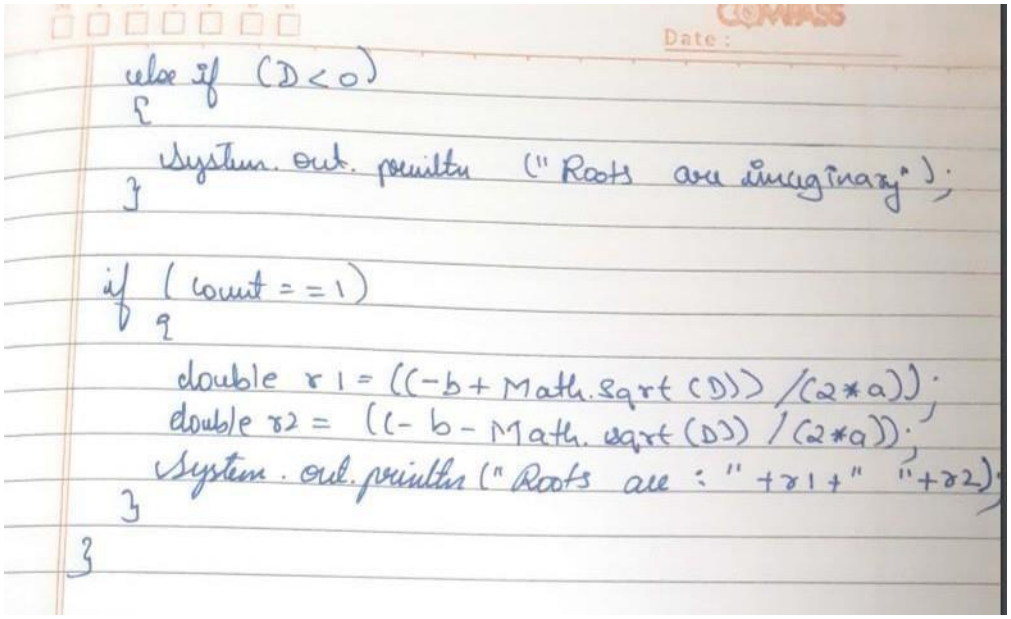


NAME :- TASMIYA FATHIMA

Date :

Develop a Java Program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.*;
class Quad
{
public static void main (String [] args)
  int a, b, c, count;
   double D;
  Scanner sc = new Scanner (System.in);
  System.out.println("\n Enter the values of a,b,c");
  a = sc.nextInt();
  b = sc.nextInt();
  c = sc.nextInt();
  D = (b*b) - (4*a*c);
    if (D==0)
  {
   System.out.println("Roots are real and equal");
    count = 1;
  }
  else if (D>0)
  {
   System.out.println(" Roots are real and
                        unequal");
    count = 1;
  }
```



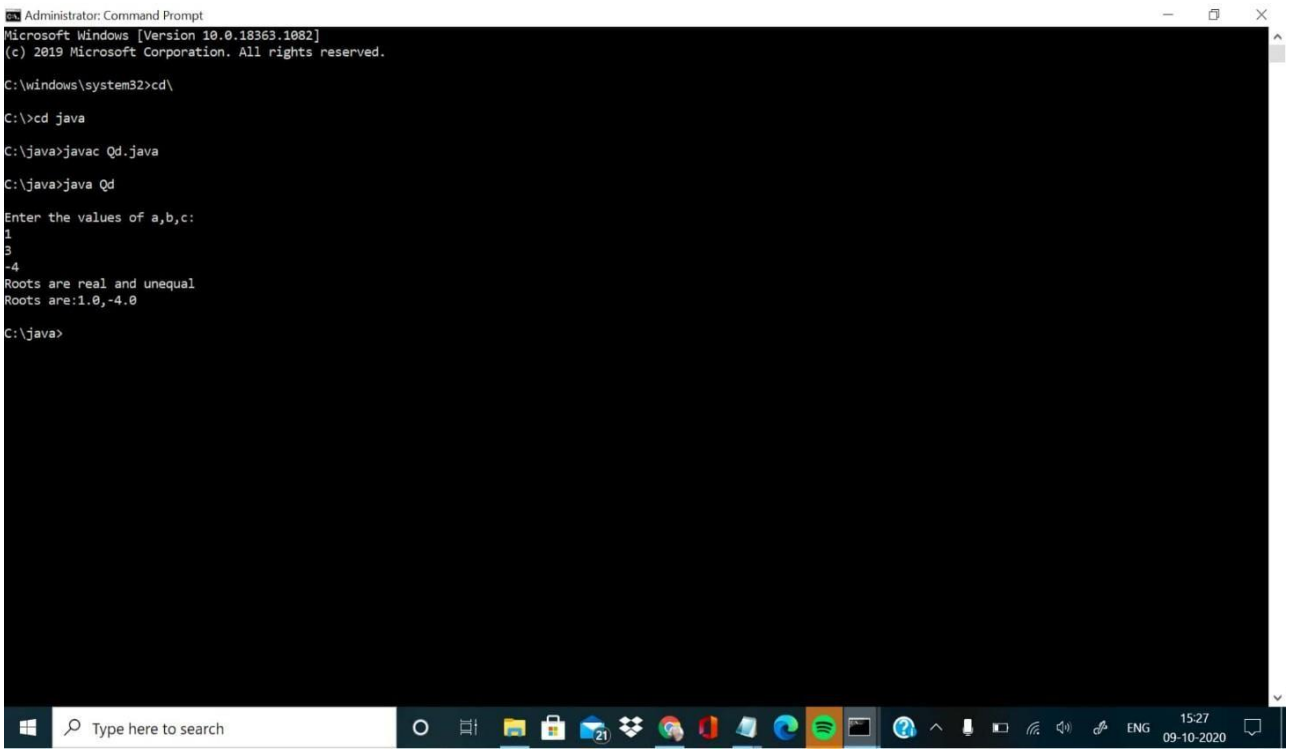
```
else if (D<0)
{
System.out.println("Roots are imaginary");
}

if (count==1)
{
double r1=((-b+Math.sqrt(D))/(2*a));
double r2=((-b-Math.sqrt(D))/(2*a));
System.out.println("Roots are :"+r1+" "+r2);
}
}
```

PROGRAM:-

```
class Qd
{


  public static void main(String[] args)
{ int
  a,b,c,f=0;
  double D;
  Scanner sc=new Scanner(System.in);
  System.out.println("\nEnter the values of
  a,b,c:"); a=sc.nextInt(); b=sc.nextInt();
  c=sc.nextInt(); D=(b*b)-(4*a*c); if(D==0)
 {
  System.out.println("Roots are real and equal"); f=1;
 } else
 if(D>0)
 {
```

```
 System.out.println("Roots are real and
unequal"); f=1; } else if(D<0)
 {
  System.out.println("Roots are imaginary");
 } if(f==1)
{

  double          r1=((-b+Math.sqrt(D))/(2*a));
  double           r2=((-b-Math.sqrt(D))/(2*a));
  System.out.println("Roots are:"+r1+","+r2);
}
}
}
```

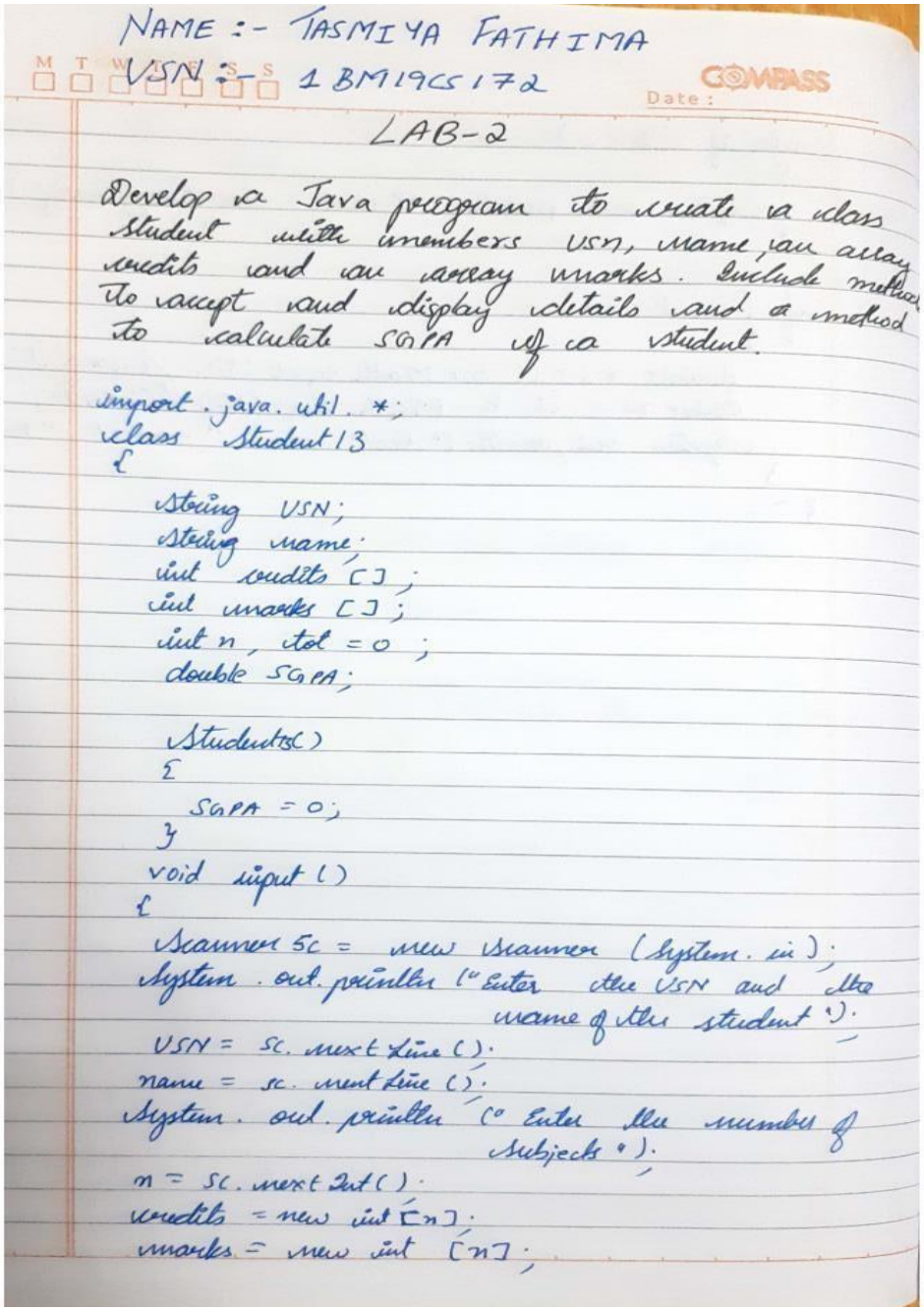
OUTPUT:-



Lab Program 2:

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

WRITE UP:-



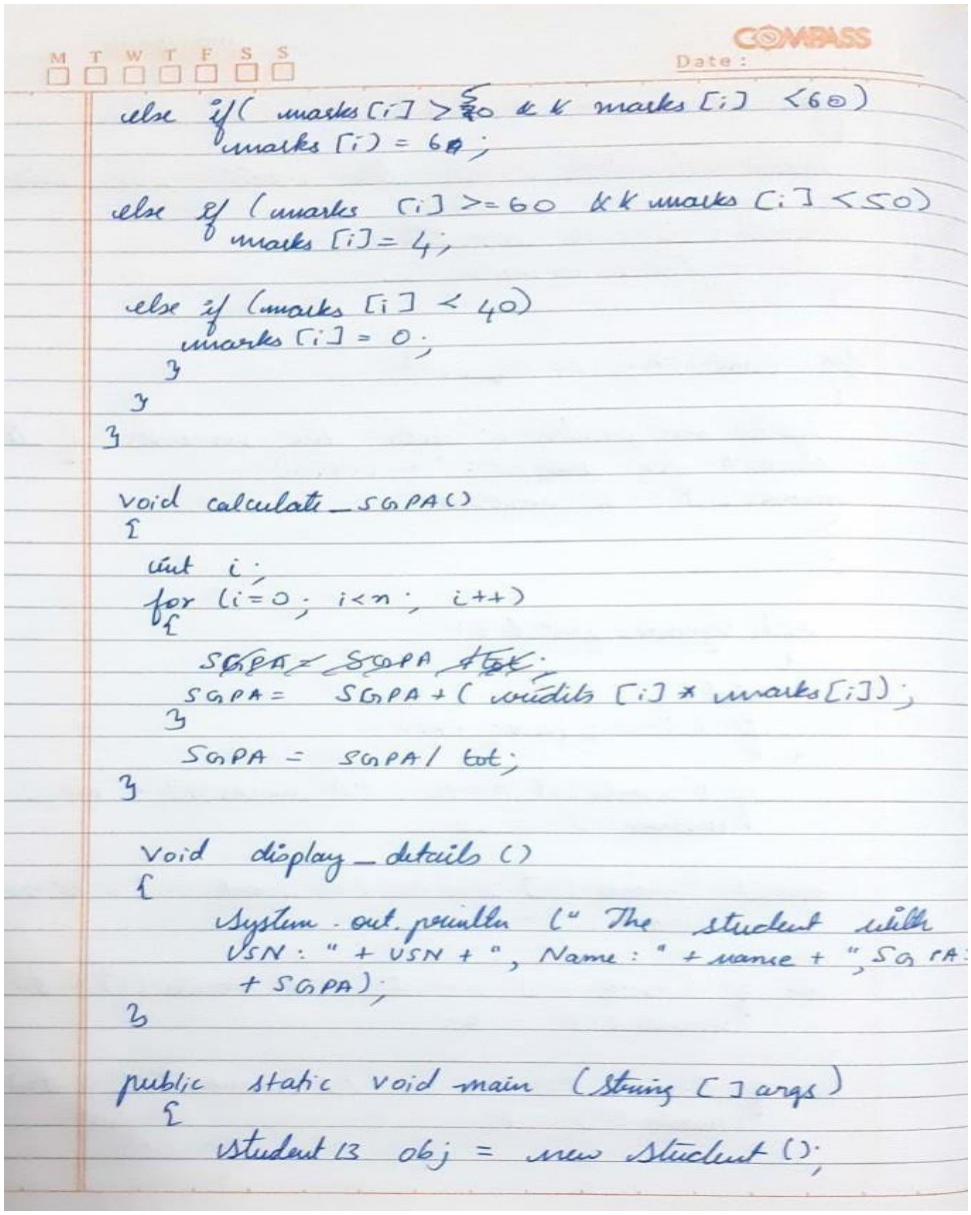
NAME :- TASMIYA FATHIMA
USN :- 1BM19CS172

LAB-2

Develop a Java program to create a class student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

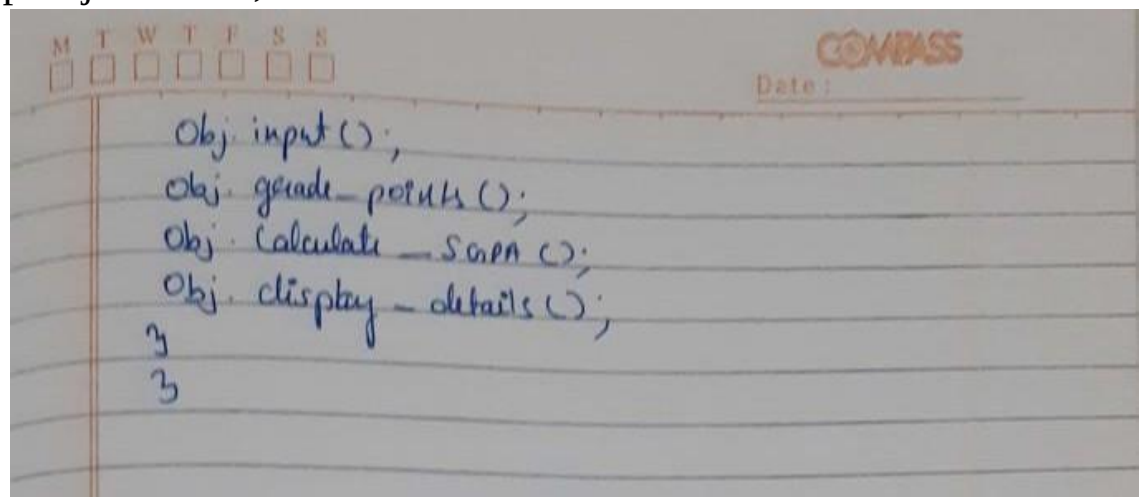
```
import java.util.*;
class Student13
{
    String USN;
    String name;
    int credits[];
    int marks[];
    int n, tot = 0;
    double SGPA;

    Student13()
    {
      SGPA = 0;
    }
    void input()
    {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the USN and the
                        name of the student:");
    USN = sc.nextLine();
    name = sc.nextLine();
    System.out.println("Enter the number of
                        subjects");
    n = sc.nextInt();
    credits = new int[n];
    marks = new int[n];
```

```
for(int i=0; i<n; i++)
{
System.out.println("Enter the credits for subject"
                    + (i+1));
credits[i] = sc.nextInt();
  tot = tot + credits[i];
}

for (int i=0; i<n; i++)
{
System.out.println("Enter the marks of the
student for subject:" + (i+1));
marks[i] = sc.nextInt();
}
}

Void grade_points()
  {
    int i;
    for(i=0; i<n; i++)
  {
    if (marks[i] >=90 && marks[i] < 100)
      marks[i] = 10;

else if (marks[i] >= 80 && marks[i] < 90)
      marks[i] = 9;

else if (marks[i] >= 70 && marks[i] < 80)
      marks[i] = 8;

else if (marks[i] >= 60 && marks[i] < 70)
      marks[i] = 7;
```



```
else if( marks[i] >=50 && marks[i] <60)
        marks[i] = 6;

else if (marks [i] >=60 && marks [i] <50)
        marks[i]= 4;

else if (marks [i] < 40)
        marks [i] = 0;
    }
  }
}

void calculate_SGPA()
{
  int i;
  for (i=0; i<n; i++)
  {
     SGPA = SGPA + ( credits [i] * marks[i]);
  }
     SGPA = SGPA / tot;
}

 Void display_details ()
 {
      System.out.println (" The student with
      USN : " + USN + ", Name : " + name + ", SGPA:
       + SGPA);
 }

public static void main (String [] args)
  {
      Student13 obj = new Student ();
```

import java.util.*;



```
Obj.input();
Obj.grade_points();
Obj.calculate_SGPA();
Obj.display_details();
}
}
```

class Student13

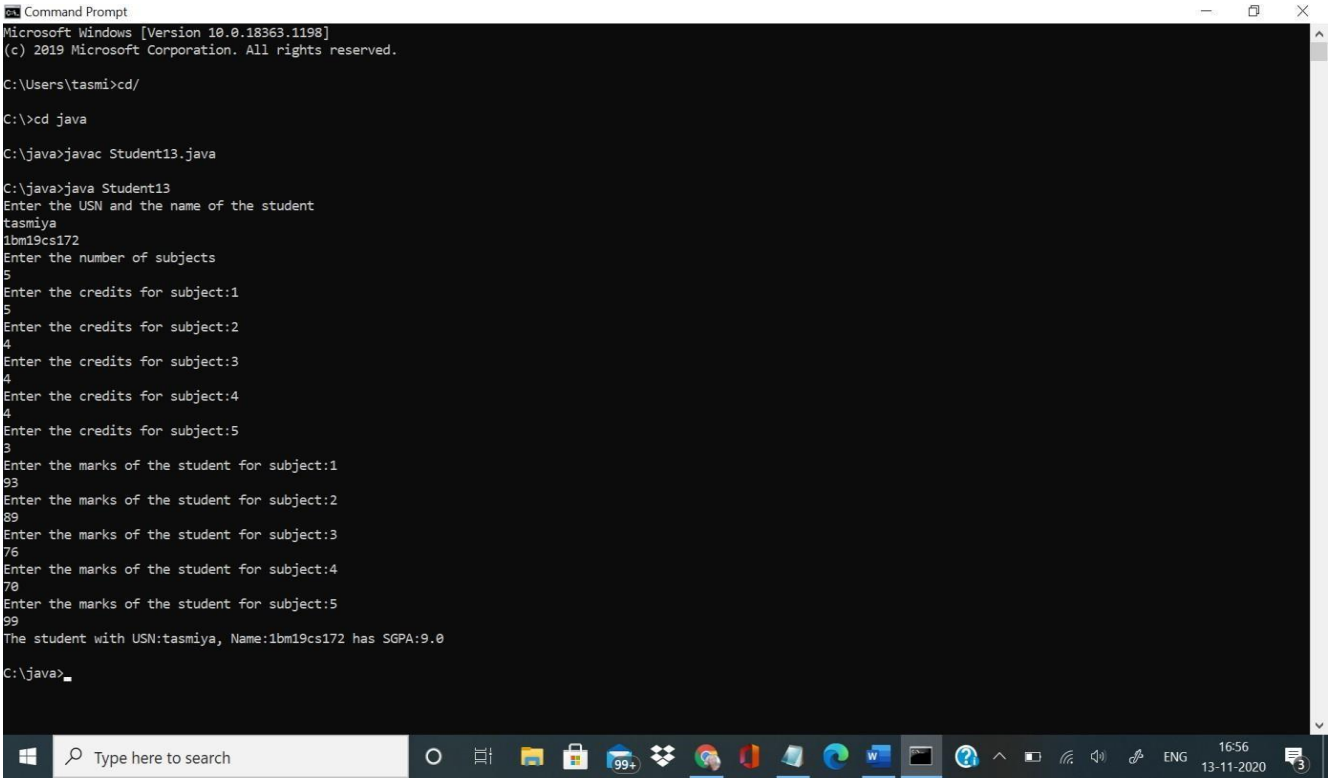
```
{
 String USN;
 String name;
 int credits[];
 int marks[]; int
 n,tot=0;
 double SGPA;

Student13()
{
 SGPA=0;
} void
input()
{
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the USN and the name of the
 student"); USN=sc.nextLine(); name=sc.nextLine();
 System.out.println("Enter the number of subjects");
 n=sc.nextInt(); credits=new int[n]; marks =new
 int[n]; for(int i=0;i<n;i++)
{
 System.out.println("Enter the credits for
 subject:"+(i+1)); credits[i]=sc.nextInt();
 tot=tot+credits[i];
} for(int
i=0;i<n;i++)
{
```

```
 System.out.println("Enter the marks of the student for
subject:"+(i+1));
 marks[i]=sc.nextInt();
} } void
grade_points()
{

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

        if(marks[i]>=90 && marks[i]<100)
        {

            marks[i]=10;
        }

        else if(marks[i]>=80 && marks[i]<90)
        {

            marks[i]=9;
        }

        else if(marks[i]>=70 && marks[i]<80)
        { marks[i]=8;
        }
```

```
        else if(marks[i]>=60 && marks[i]<70)
        {

           marks[i]=7;
        }

        else if(marks[i]>=50 && marks[i]<60)
        {

           marks[i]=6;
        }

        else if(marks[i]>=40 && marks[i]<50)
        {

           marks[i]=4;
        }

        else if(marks[i]<40)
        {

           marks[i]=0;
        }
    }
}
```

```
void calculate_SGPA()
{ int
 i;
 for(i=0;i<n;i++)
 {
  SGPA=SGPA+(credits[i]*marks[i]);
 }
  SGPA=SGPA/tot;

} void
display_details()
{

   System.out.println("The student with USN:"+USN+",
Name:"+name+" has SGPA:"+SGPA);
} public static void main(String[]
args)
{

 Student13 obj=new
 Student13(); obj.input();
 obj.grade_points();
 obj.calculate_SGPA();
 obj.display_details();
}
}
```

OUTPUT:-



LAB PROGRAM -3

Lab program 3: Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

Write Up:-

TASMIYA FATHIMA

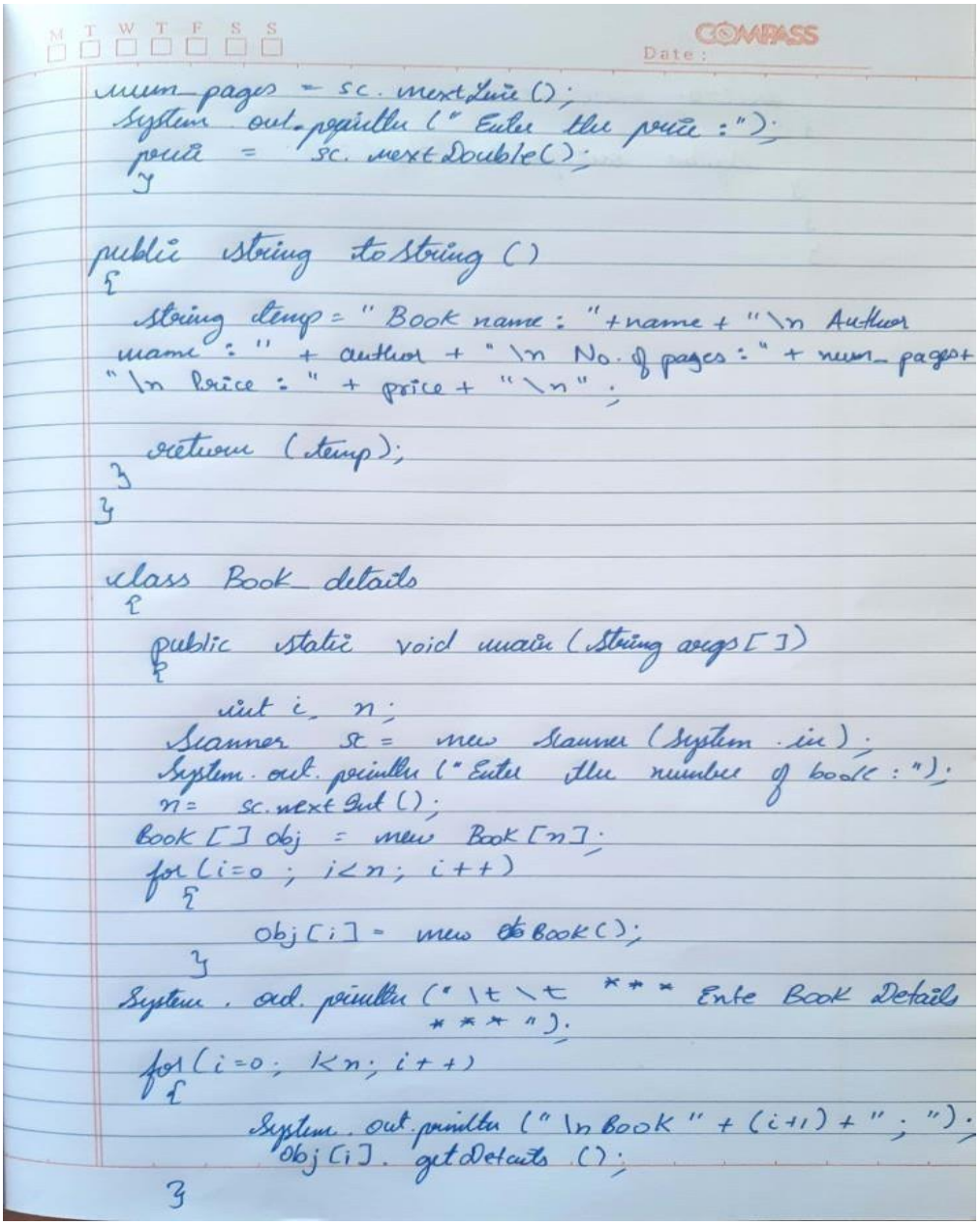
USN :- 1BM19CS172

Date :

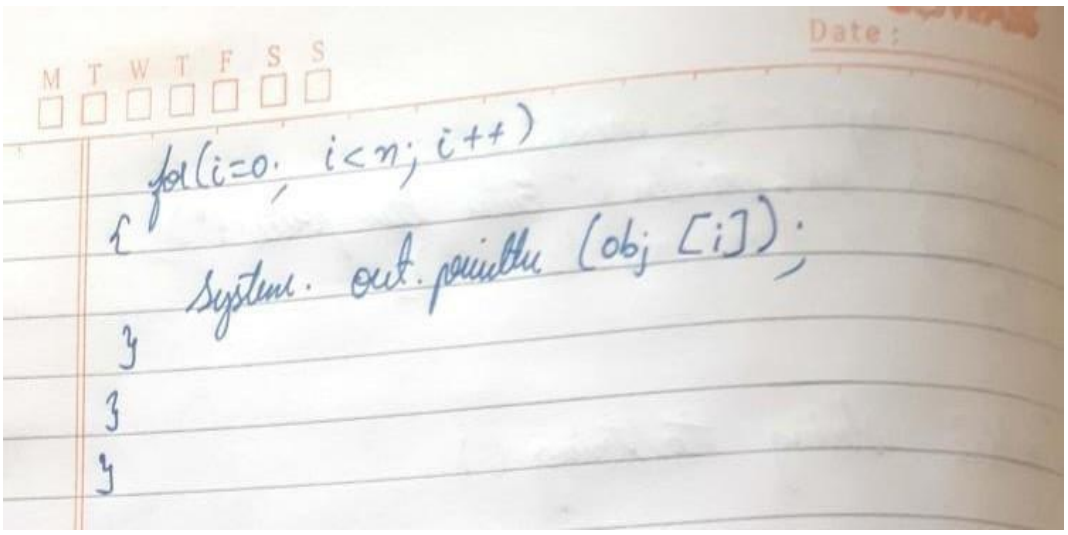
LAB-3

Create a class Book which contains four members : name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
class Book
{
    private String name, author;
    private double price;
    private int num_pages;

    Book()
    {
        name = "The Secret Key";
        author = "Rajendra";
        price = 499.00;
        num_pages = 500;
    }

    Void getDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("\n Enter the book name:");
        name = sc.nextLine();
        System.out.println("\n Enter the author name:");
        author = sc.nextLine();
        System.out.println("\n Enter the price no. of page:");
```



```
num_pages = sc.nextLine();
System.out.println("Enter the price:");
price = sc.nextDouble();
}

public String toString()
{
String temp = "Book name: "+name+"\n Author name: " + author + "\n No. of pages: " + num_pages+ "\n Price: " + price + "\n";

return (temp);
}
}

class Book_details
{
public static void main(String args[])
{
int i, n;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number of books:");
n= sc.nextInt();
Book [] obj = new Book[n];
for(i=0; i<n; i++)
{
obj[i] = new Book();
}
System.out.println("\t\t *** Enter Book Details ***");
for(i=0; i<n; i++)
{
System.out.println("\n Book " + (i+1) + ";");
obj[i].getDetails();
}
```



```
for(i=0; i<n; i++)
{
System.out.println(obj[i]);
}
}
}
```

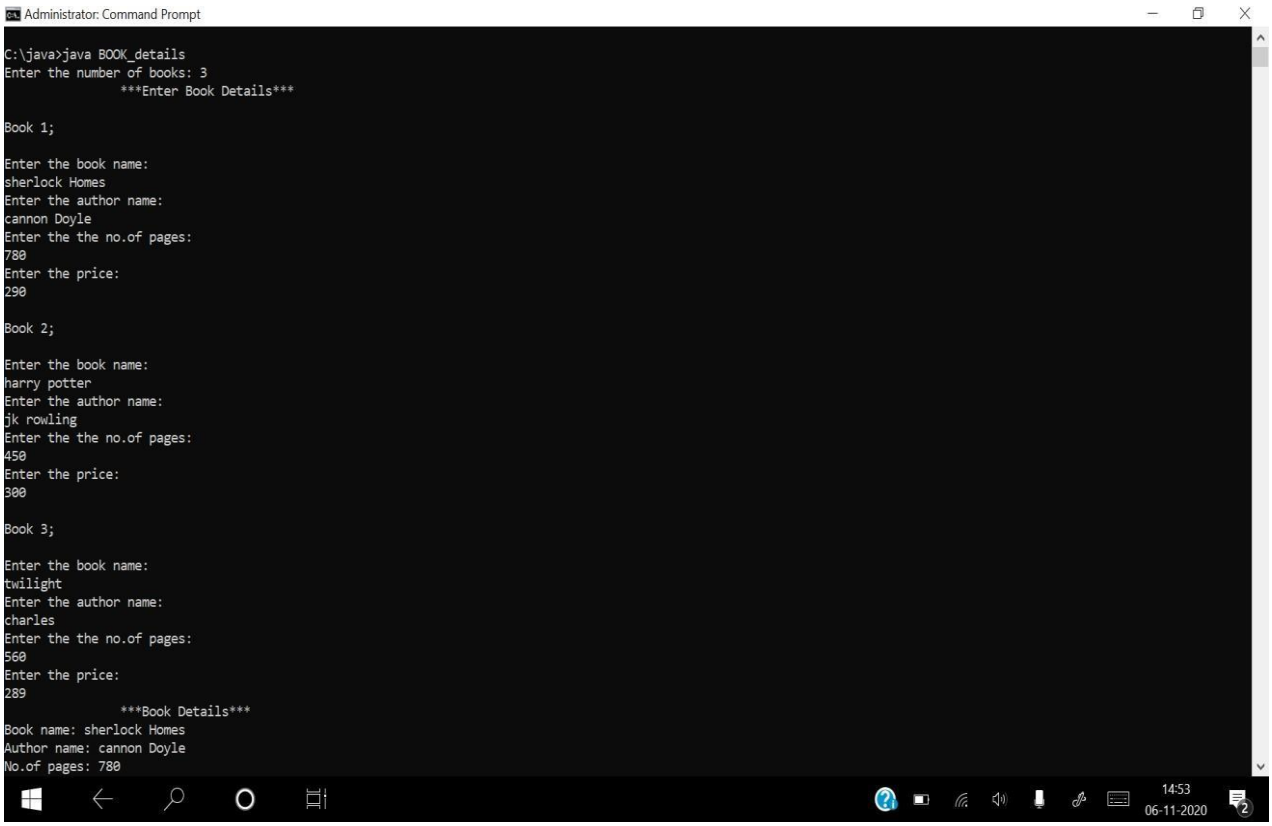
```
import java.util.Scanner;
```

```
class Book
{

        private String name,author;
        private      double      price;
        private int num_pages;

       Book()
        {

                name="The Secret key";
                author="Rajendra";
                price=399.00;
                num_pages=500;
        }



        void getDetails()
        {

                Scanner in=new Scanner(System.in);
                System.out.println("\nEnter the book name: ");
                name=in.nextLine();
                System.out.println("Enter the author name: ");
                author=in.nextLine();
```

```
            System.out.println("Enter the the no.of pages: ");
            num_pages=in.nextInt(); System.out.println("Enter the
            price: "); price=in.nextDouble();
        }



        public String toString()
        {

            String temp="Book name: "+name+"\nAuthor name:
"+author+"\nNo.of pages: "+num_pages+"\nPrice: "+price+"\n";
            return(temp);


        }


} class
BOOK_details
{

  public static void main(String args[]) {


        int i,n;
        Scanner in=new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        n=in.nextInt();
        Book[] obj=new Book[n]; for(i=0;i<n;i++)
```

```
        {

                obj[i]=new Book();
        }
        System.out.println("\t\t***Enter Book Details***");
       for(i=0;i<n;i++)
        {

                System.out.println("\nBook "+(i+1)+";");
                obj[i].getDetails();
        }

        System.out.println("\t\t***Book Details***");
       for(i=0;i<n;i++)
        {

                System.out.println(obj[i]);
        }
  }
}
```

OUTPUT:



LAB PROGRAM-4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given Shape.

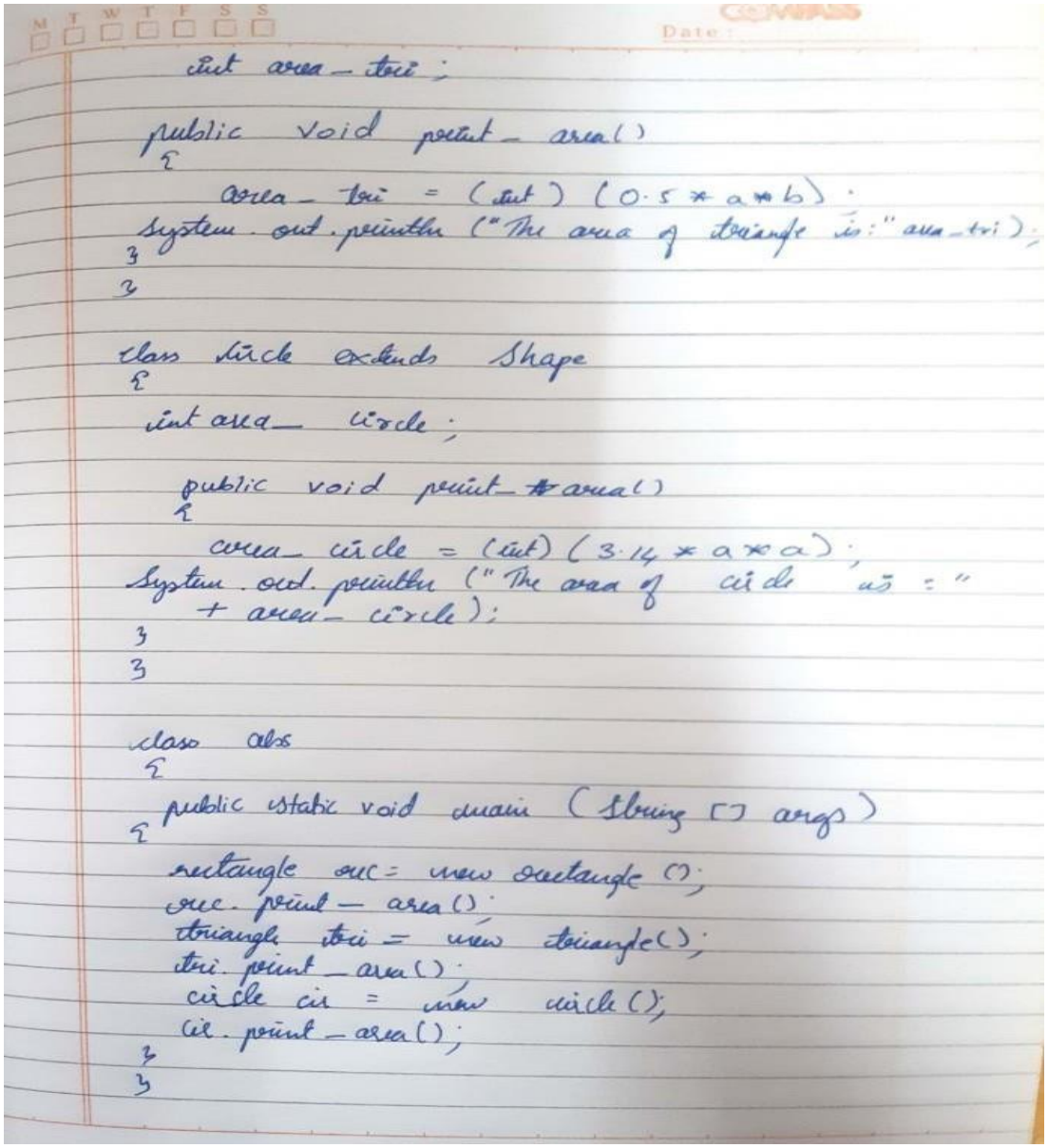
WRITE UP:-

NAME:- TASMIYA FATHIMA

USN: 1BM19CS172 LAB-4

Develop a Java program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
abstract class Shape
{
    int a = 3;
    int b = 4;

    abstract public void printArea();
}

class Rectangle extends Shape
{
    public int area_rect;

    public void printArea()
    {
        area_rect = a*b;
        System.out.println("The area of rectangle
            is: " + area_rect);
    }
}

class Triangle extends Shape
{
```



```
int area_tri;

public void print_area()
{
    area_tri = (int) (0.5 * a * b);
System.out.println("The area of triangle is:" area_tri);
}
}

class circle extends Shape
{
 int area_ circle;

   public void print_area()
   {
     area_ circle = (int) (3.14 * a * a);
System.out.println("The area of circle is :"
   + area_ circle);
}
}

class abs
{
  public static void main (String [] args)
{
  rectangle rec = new rectangle ();
  rec.print_area();
  triangle tri = new triangle();
  tri.print_area();
  circle cir = new circle();
  cir.print_area();
}
}
```

abstract class Shape

{ int a=3; int b=4; abstract

public void print_area();

} class rectangle extends

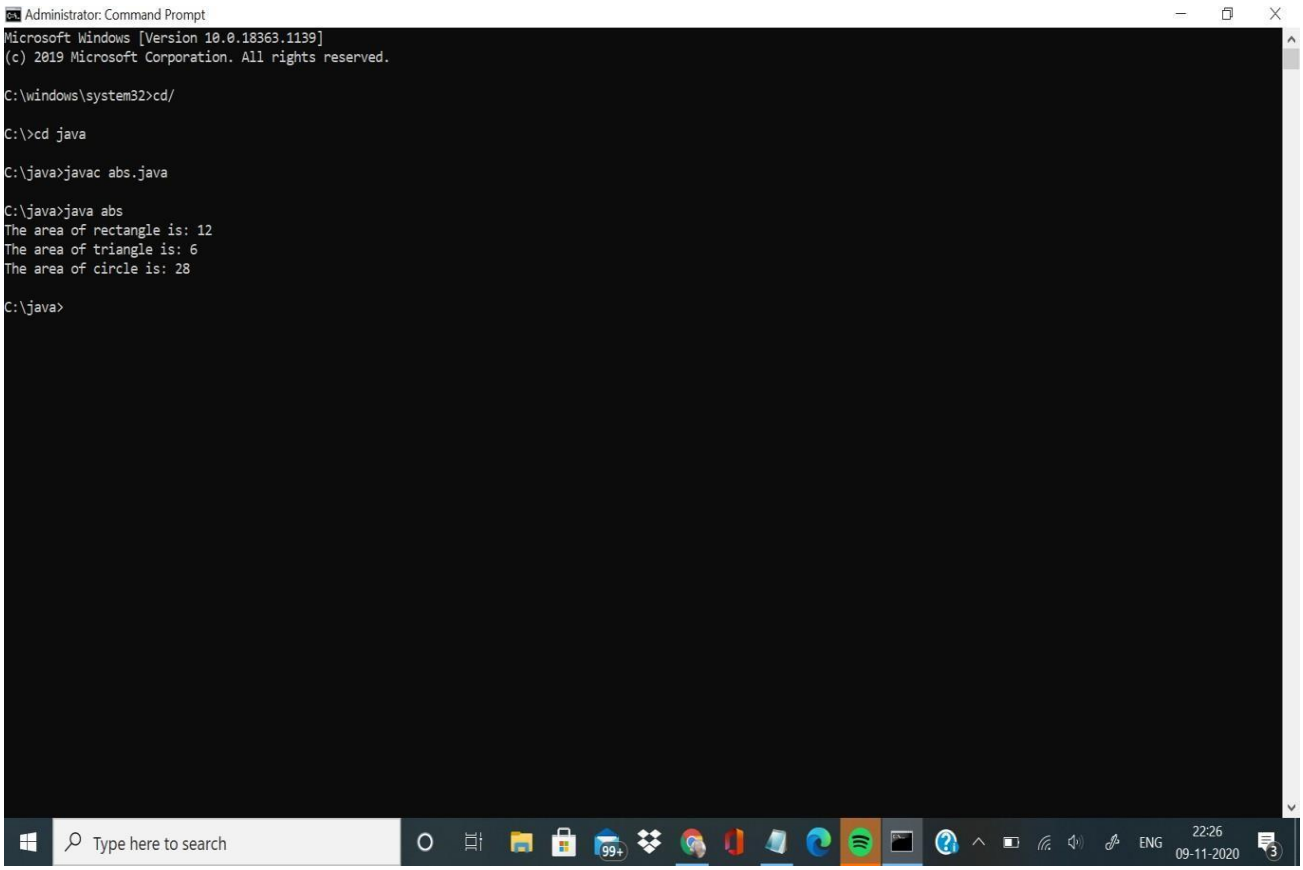
Shape {

public int area_rect;

```
public void print_area()
{ area_rect=a*b;
System.out.println("The area of rectangle is: "+area_rect);
} } class triangle extends
Shape
{
int area_tri;


public void print_area()
{
area_tri=(int) (0.5*a*b);
System.out.println("The area of triangle is: "+area_tri);
} } class circle extends
Shape
{
int area_circle;


public void print_area()
{
area_circle=(int) (3.14*a*a);
System.out.println("The area of circle is: "+area_circle);
} } class abs{ public static void
main(String[] args){ rectangle rec =
new rectangle(); rec.print_area();
triangle tri = new triangle();
```

```
tri.print_area(); circle cir = new
circle(); cir.print_area();
}
}
```

OUTPUT:-



LAB PROGRAM-5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific

to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

WRITE UP:-

TASMIYA FATHIMA

USN:- 1BM19CS172

Date :

LAB-5:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes cur_acc and sav_acc to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

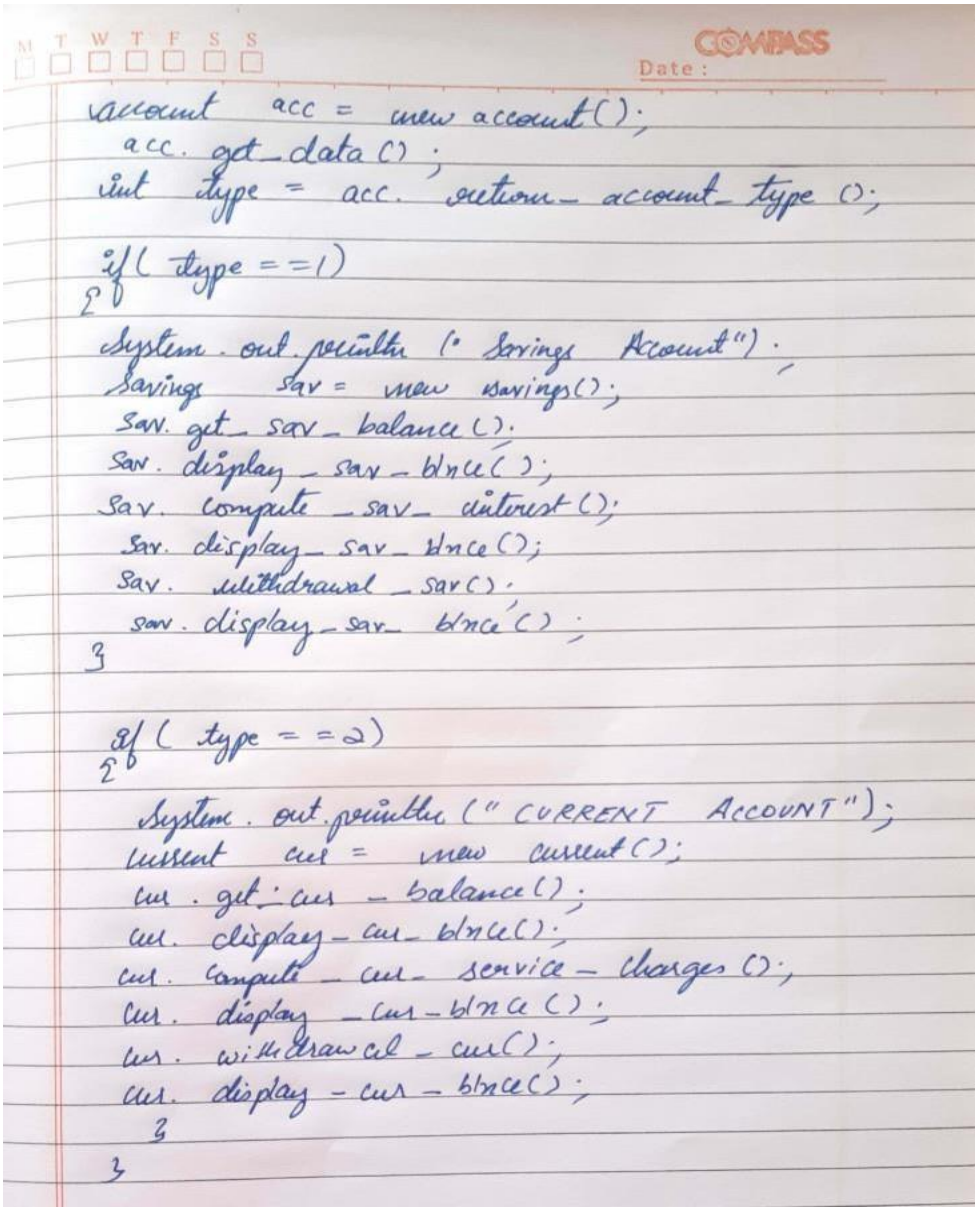
a) Accept deposit from customer and update the balance
b) Display the balance
c) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

class account
{
    private String name;
    private long account_number;
    private int account_type;
```

```
    double balance;

    void get_data ()
    {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter your name");
      name = sc.next();
      System.out.println("Enter the account_number");
      account_number = sc.nextLong();
      System.out.println("Choose the account type");
      System.out.println("1. Savings account");
      System.out.println("2. Current account");
      account_type = sc.nextInt();
    }

    int return_account_type()
    {
      return account_type;
    }
}

class savings extends account
{
    Scanner sc = new Scanner(System.in);
    double amount;

    void get_sav_balance()
    {
      System.out.println("enter the amount to be
      placed in your savings account");
      amount sc.nextDouble();
      balance += amount;
    }
```

```
void display_sav_bln()
{
  System.out.println("balance =" + balance);
}

void compute_sav_interest()
{
 System.out.println("Interest @ 5% shall be
added to your balance");
 balance = balance + (.05 * balance);
}

Void withdrawl_sav()
{
System.out.println("Enter the amount to be
withdrawn");
 amount = sc.nextDouble();
  balance = balance - amount;
}
}

Class current extends account
{
    Scanner sc = new Scanner(System.in);
     double amount;
     final double min_balance = 5000;

    Void get_cur_balance()
    {
    System.out.println("Enter the amount to
be placed in your account");
    amount = sc.nextDouble();
    balance += amount;
}
```

```
void display_cur_balance()
{
System.out.println("balance = " +balance);
}

Void compute_cur_service_charges()
{
  if (balance < min_balance)
  {
    System.out.println("service tax of Rs. 500
    shall be levied");
    balance = balance - 500;
  }
else
  {
    System.out.println("minimum balance is
    maintained");
  }
}

Void withdrawl_cur()
{
System.out.println("Enter the amount to
be withdrawn");
amount = sc.nextDouble();
balance = balance - amount;
  }
}

class bankmain
{
public static void main(String args[])
{
System.out.println("Enter the bank details");
```



```
account acc = new account();
acc.get_data();
int type = acc.return_account_type();

if(type==1)
{
System.out.println("Savings Account");
Savings sav = new savings();
sav.get_sav_balance();
sav.display_sav_bnce();
sav.compute_sav_interest();
sav.display_sav_bnce();
sav.withdrawal_sav();
sav.display_sav_bnce();
}

if(type==2)
{
System.out.println("CURRENT ACCOUNT");
current cur = new current();
cur.get_cur_balance();
cur.display_cur_bnce();
cur.compute_cur_service_charges();
cur.display_cur_bnce();
cur.withdrawal_cur();
cur.display_cur_bnce();
}
}
```

```
import java.util.Scanner;

class account
{
```

```
    private String name; private
    long account_number; private
    int account_type; double
    balance;


    void get_data()
    {
     Scanner ss=new Scanner(System.in);
     System.out.println("enter your name");
     name=ss.next();
     System.out.println("enter the account_number");
    account_number=ss.nextLong();
    System.out.println("choose the account type ");
    System.out.println("1.savings account");
    System.out.println("2.current account");
    account_type=ss.nextInt();
    }



    int return_account_type()
    { return
     account_type;
    }
}
```

```
class savings extends account
{

        Scanner ss=new Scanner(System.in); double
        amount;

        void get_sav_balance()
        {
         System.out.println("enter the amount to be placed in your
savings account"); amount=ss.nextDouble(); balance+=amount;
        }


        void display_sav_blnce()
        {
         System.out.println("balance="+balance);
        }


        void compute_sav_interest()
        {
         System.out.println("interest of 5% shall be added to your
balance"); balance=balance+(.05*balance);
        }


        void withdrawl_sav()
        {
```

```
        System.out.println("enter the amount to be withdrawn");
        amount=ss.nextDouble();
        balance=balance-amount;
       }


}

class current extends account
{

       Scanner ss=new Scanner(System.in); double
       amount;
       final double min_balance=5000;


       void get_cur_balance()
       {
        System.out.println("enter the amount to be placed in your
current account"); amount=ss.nextDouble(); balance+=amount;
       }


       void display_cur_blnce()
       {
        System.out.println("balance="+balance);
       }
```

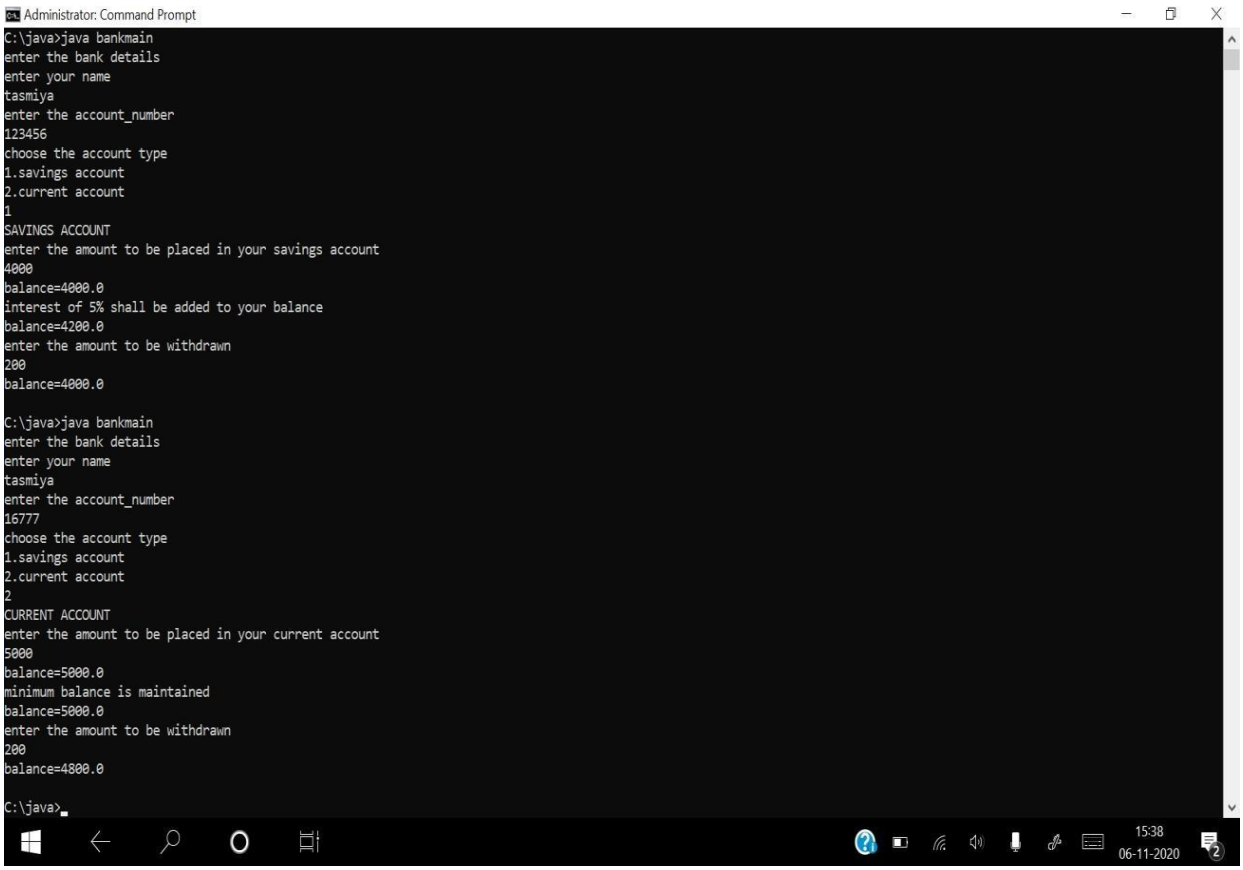
```
    void compute_cur_service_charges()
    { if(balance<min_balance)
            {
             System.out.println("service tax of rs.500 shall be levied");
             balance=balance-500;
            }

      else
            {
             System.out.println("minimum balance is maintained");
            }
    }


    void withdrawl_cur()
    {

      System.out.println("enter the amount to be
      withdrawn"); amount=ss.nextDouble();
      balance=balance-amount;
    }


}

class bankmain
```

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

        System.out.println("enter the bank
        details"); account acc=new account();
        acc.get_data(); int
        type=acc.return_account_type(); if
        (type==1)
       {
        System.out.println("SAVINGS
        ACCOUNT"); savings sav=new savings();
        sav.get_sav_balance();
        sav.display_sav_blnce();
        sav.compute_sav_interest();
        sav.display_sav_blnce();
        sav.withdrawl_sav();
        sav.display_sav_blnce();
       } if(type==2)
       {
        System.out.println("CURRENT
         ACCOUNT"); current cur=new current();
         cur.get_cur_balance();
         cur.display_cur_blnce();
         cur.compute_cur_service_charges();
         cur.display_cur_blnce(); cur.withdrawl_cur();
         cur.display_cur_blnce();
```

```
        }
}
```

```
}
```

OUTPUT:-

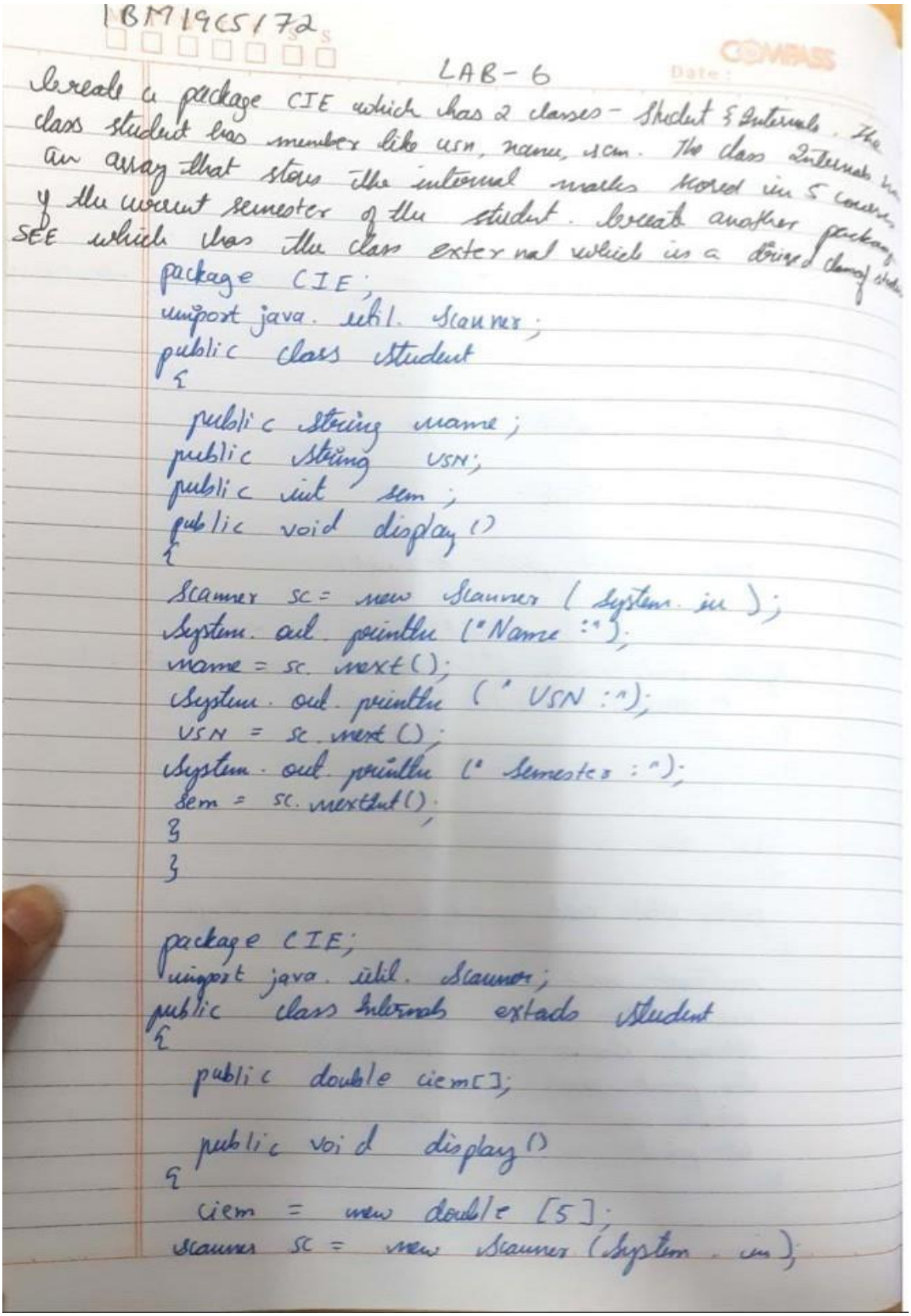


Lab Program: 6

/*Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses. Below is

two programs of class Student and class Internals. Both belong to a package named CIE. Class Internals extends from class Student. */

WRITE UP :-



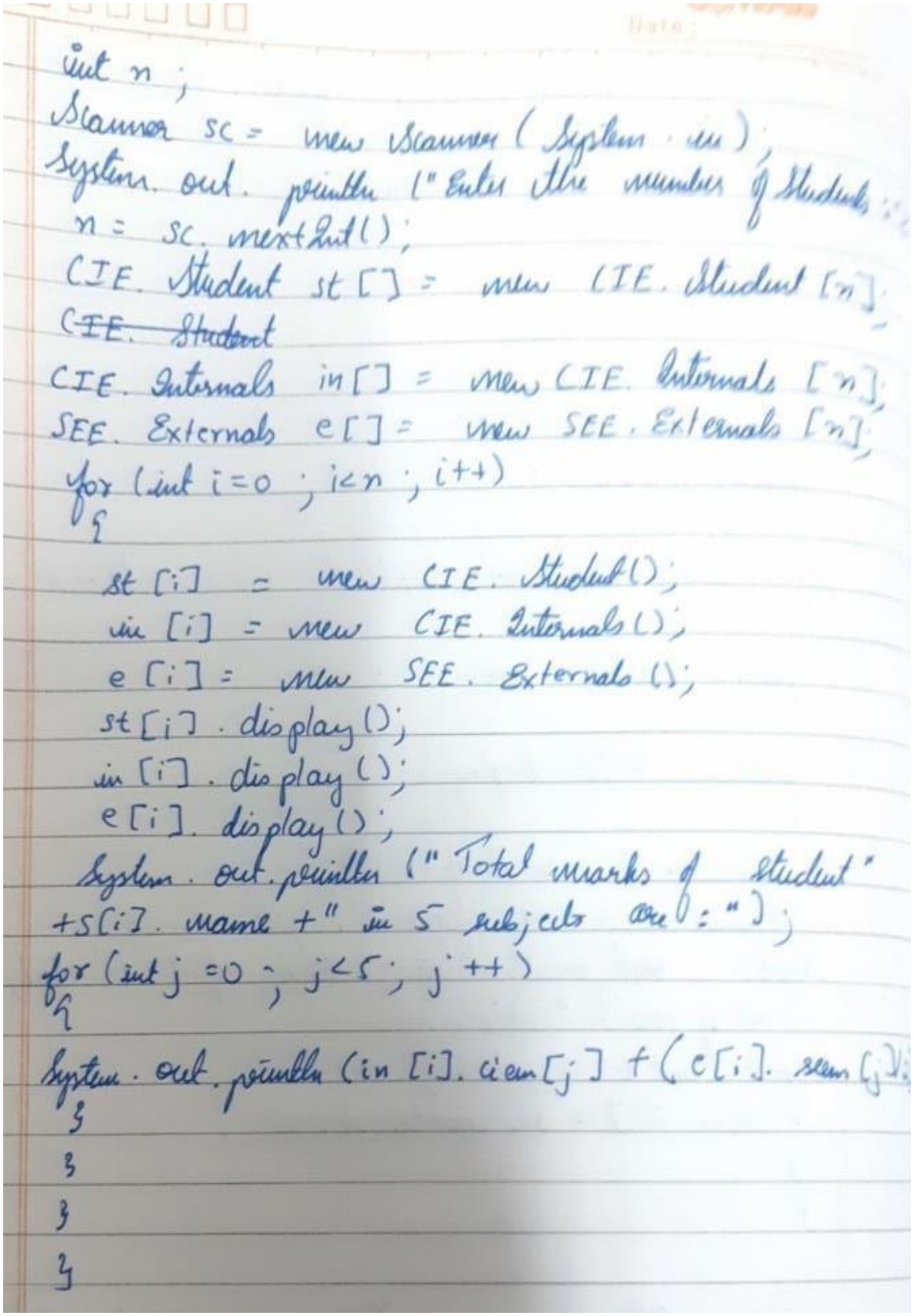
1BM19CS172

LAB-6

Write a package CIE which has 2 classes - Student & Internals. The class student has member like usn, name, sem. The class Internals has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class external which is a derived class of Student

```
package CIE;
import java.util.Scanner;
public class Student
{
  public String name;
  public String USN;
  public int sem;
  public void display()
  {
  Scanner sc = new Scanner (System.in);
  System.out.println("Name :");
  name = sc.next();
  System.out.println("USN :");
  USN = sc.next();
  System.out.println("Semester :");
  sem = sc.nextInt();
  }
  }

package CIE;
import java.util.Scanner;
public class Internals extends Student
{
  public double iem[];

  public void display()
  {
  iem = new double [5];
  Scanner sc = new Scanner (System.in);
```

```
System.out.println("CIE Marks for 5 subjects");
for(int i=0; i<5; i++)
 iem[i] = sc.nextDouble();
}
}

package SEE;
import java.util.*;
import CIE.*;

public class Extemals extends CIE Student
{
 public double seem[];

 public void display()
 {
   seem = new double[5];
 Scanner sc = new Scanner(System.in);
System.out.println("SEE Marks for 5 subjects
 (out of 100)." );
 for(int i=0; i<5; i++)
  seem[i] = sc.nextDouble();
 }
}

File with name Main.java (driver class)

import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main
{
 public static void main(String args[])
 {
```



```
int n ;
Scanner sc = new Scanner ( System . in );
System. out. println ("Enter the number of Students ");
n = sc. nextInt();
CIE. Student st [] = new CIE. Student [n];
CIE. Student
CIE. Internals in[] = new CIE. Internals [n];
SEE. Externals e[] = new SEE. Externals [n];
for (int i=0 ; i<n ; i++)
{
   st [i]  = new CIE. Student();
   in [i] = new CIE. Internals();
   e [i] = new SEE. Externals ();
   st[i] . display();
   in [i] . display ();
   e [i]. display ();
   System . out. println (" Total marks of student"
+s[i]. name +" in 5 subjects are :" );
for (int j =0 ; j<5; j++)
{
System . out. println (in [i]. item[j] + ( e[i]. item[j]);
}
}
}
}
```

File with name Student.java

```
package CIE; import
java.util.Scanner;
 public class Student
```

```
 { public String
name; public String
usn; public int sem;
public void display()
{
Scanner s=new Scanner(System.in);
System.out.println("Name:");
name=s.next();
System.out.println("USN:"); usn=s.next();
System.out.println("Semester:");
sem=s.nextInt(); }
}
```

File with name Internals.java

```
package CIE;
import java.util.Scanner;


public class Internals extends Student
{ public double
ciem[]; public void
display()
{
ciem=new double[5];
Scanner t=new Scanner(System.in);
System.out.println("CIE Marks for 5 subjects(out of 50):");
for(int i=0;i<5;i++) ciem[i]=t.nextDouble();
```

```
}
}
```

File with name Externals.java

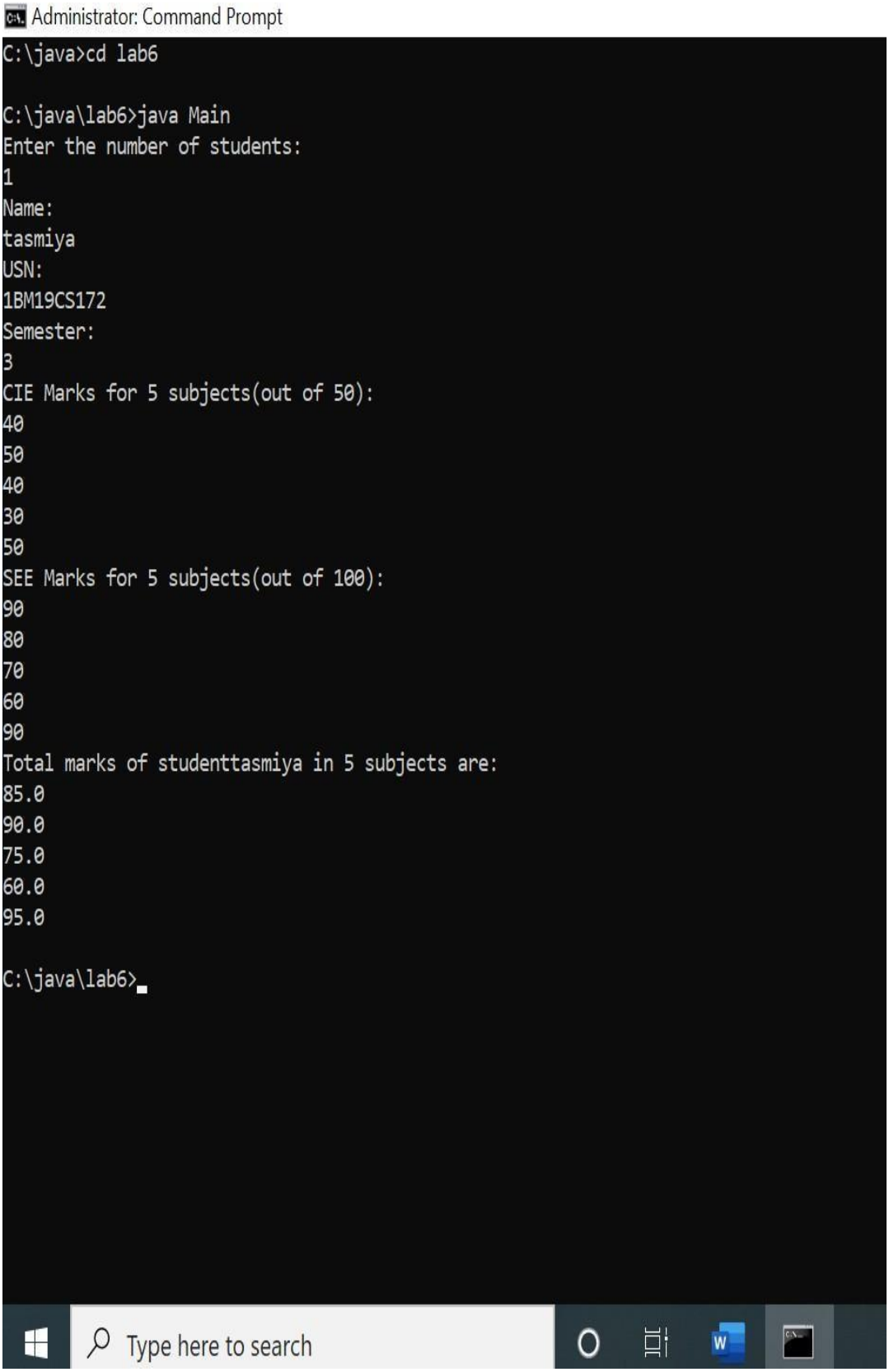
```
package SEE;
import CIE.*;
import java.util.*;
public class Externals extends CIE.Student
{ public double
seem[]; public void
display()
{
seem=new double[5];
Scanner s=new Scanner(System.in);
System.out.println("SEE Marks for 5 subjects(out of 100):"); for(int
i=0;i<5;i++) seem[i]=s.nextDouble();
}
}
```

File with name Main.java

```
import CIE.*; import
SEE.*; import
java.util.Scanner; public
class Main
```

```
{ public static void main(String
args[])
{ int
n;
Scanner s=new Scanner(System.in);
System.out.println("Enter the number of students:"); n=s.nextInt();
CIE.Student st[]=new CIE.Student[n];
CIE.Internals in[]=new CIE.Internals[n];
SEE.Externals e[]=new SEE.Externals[n]; for(int
i=0;i<n;i++)
{ st[i]=new
CIE.Student(); in[i]=new
CIE.Internals(); e[i]=new
SEE.Externals();
st[i].display();
in[i].display();
e[i].display();
System.out.println("Total marks of student"+st[i].name+" in 5
subjects are:"); for(int j=0;j<5;j++)
{
System.out.println(in[i].ciem[j]+(e[i].seem[j]/2));
}
}
}
}
```

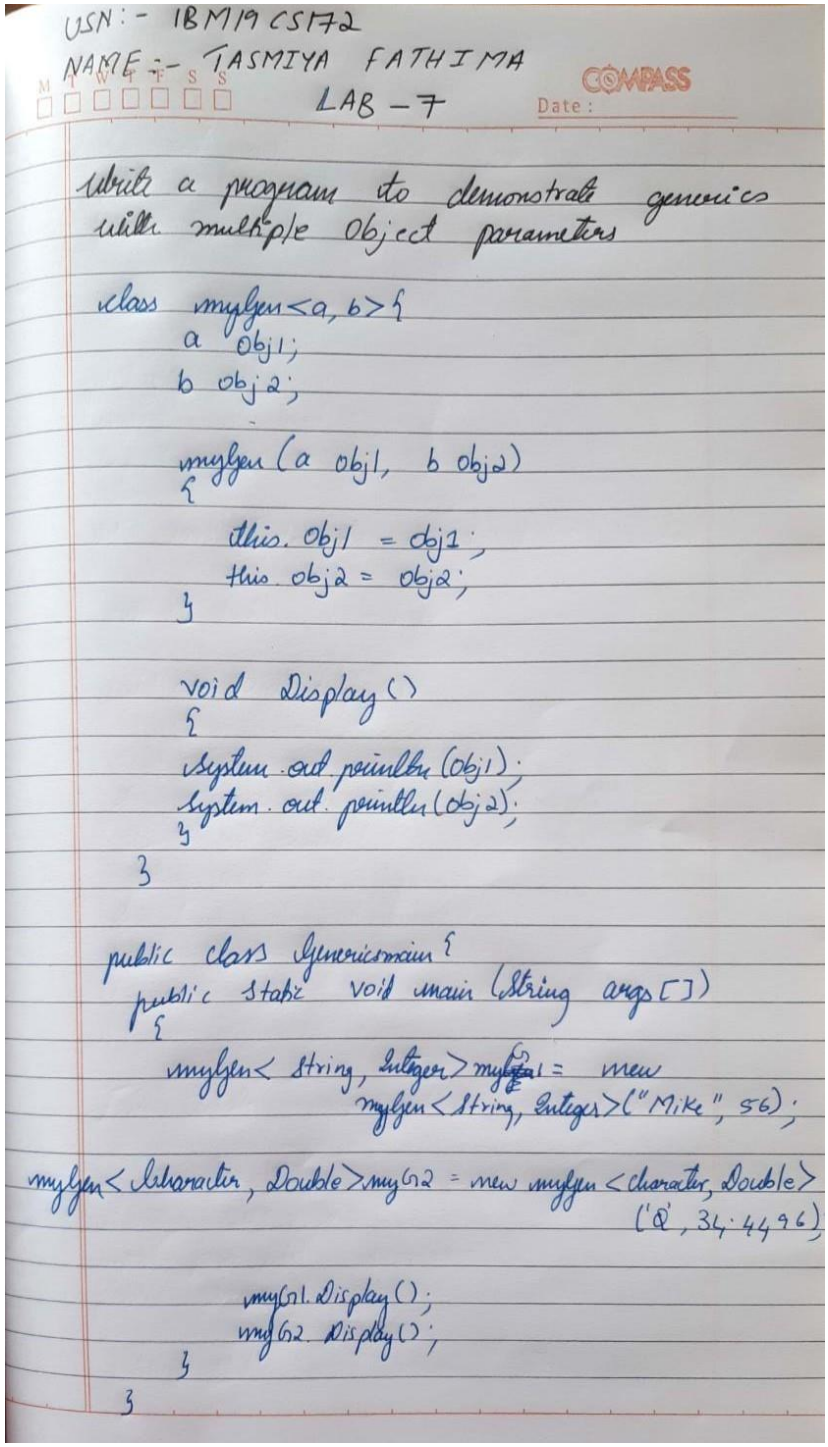
OUTPUT:-


Administrator: Command Prompt
C:\java>cd lab6
C:\java\lab6>java Main
Enter the number of students:
1
Name:
tasmiya
USN:
1BM19CS172
Semester:
3
CIE Marks for 5 subjects(out of 50):
40
50
40
30
50
SEE Marks for 5 subjects(out of 100):
90
80
70
60
90
Total marks of studenttasmiya in 5 subjects are:
85.0
90.0
75.0
60.0
95.0
C:\java\lab6>
Type here to search

Lab Program: 7

Write a program to demonstrate generics with multiple object parameters.

WRITE UP :-



USN :- 1BM19CS172
NAME :- TASMIYA FATHIMA
LAB - 7
Date :

Write a program to demonstrate generics with multiple object parameters

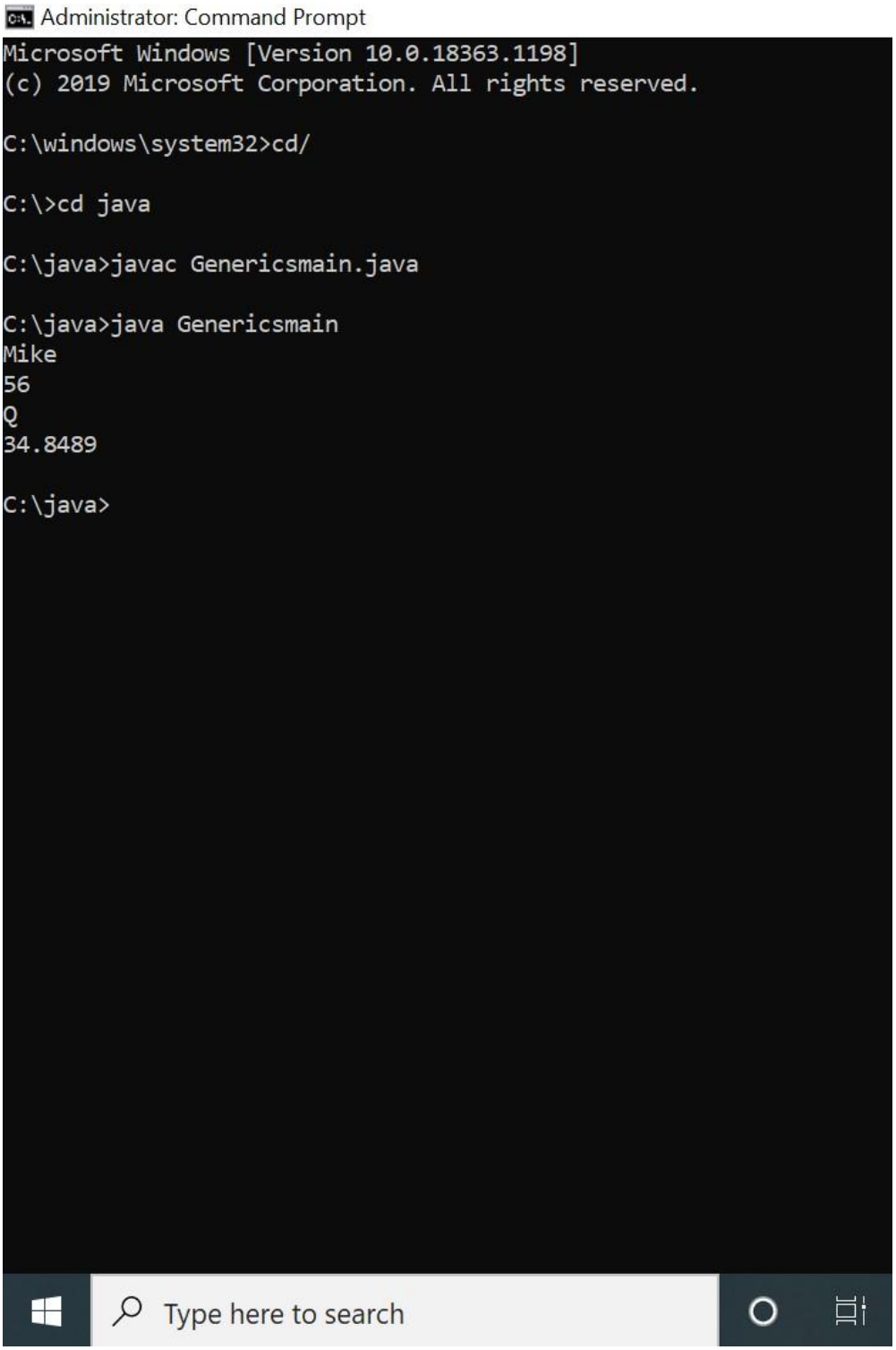
```
class mygen<a, b>{
    a obj1;
    b obj2;

    mygen(a obj1, b obj2)
    {
        this.obj1 = obj1;
        this.obj2 = obj2;
    }

    void Display()
    {
    System.out.println(obj1);
    System.out.println(obj2);
    }
}

public class Genericsmain {
    public static void main(String args[])
    {
        mygen<String, Integer> myg1 = new
                    mygen<String, Integer>("Mike", 56);

mygen<Character, Double> myG2 = new mygen<Character, Double>
                    ('Q', 34.4496);

        myG1.Display();
        myG2.Display();
    }
}
```

class myGen<a,b>{ a

obj1;

```
    b obj2;


  myGen(a obj1, b obj2){
      this.obj1 = obj1; this.obj2
      = obj2;
  } void
  Display(){
      System.out.println(obj1);
      System.out.println(obj2);
  }
}

public class Genericsmain{
    public static void main(String args[]){
        myGen<String,Integer>myG1 = new
myGen<String,Integer>("Mike",56);
        myGen<Character,Double>myG2 = new
myGen<Character,Double>('Q',34.8489);
        myG1.Display(); myG2.Display();
    }
}
```

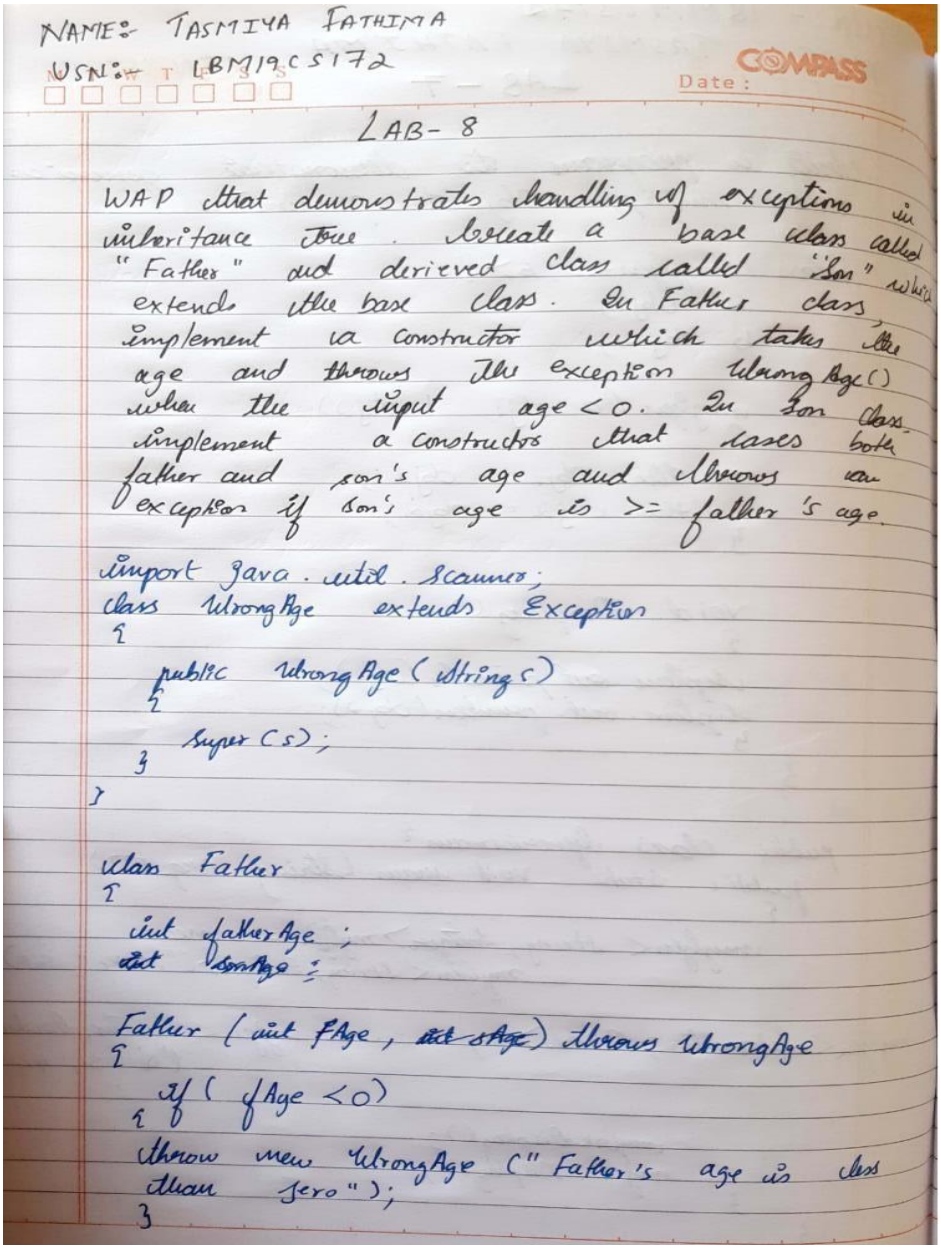
OUTPUT:-



Lab Program: 8

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

WRITE UP :-



NAME: TASMIYA FATHIMA
USN: 1BM19CS172
Date:

LAB- 8

WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that takes both father and son's age and throws an exception if son's age is >= father's age.

```
import java.util.Scanner;
class WrongAge extends Exception
{
    public WrongAge(String s)
    {
        super(s);
    }
}

class Father
{
    int fatherAge;
    int sonAge;

    Father(int fAge, int sAge) throws WrongAge
    {
        if(fAge<0)
        throw new WrongAge("Father's age is less than zero");
    }
```

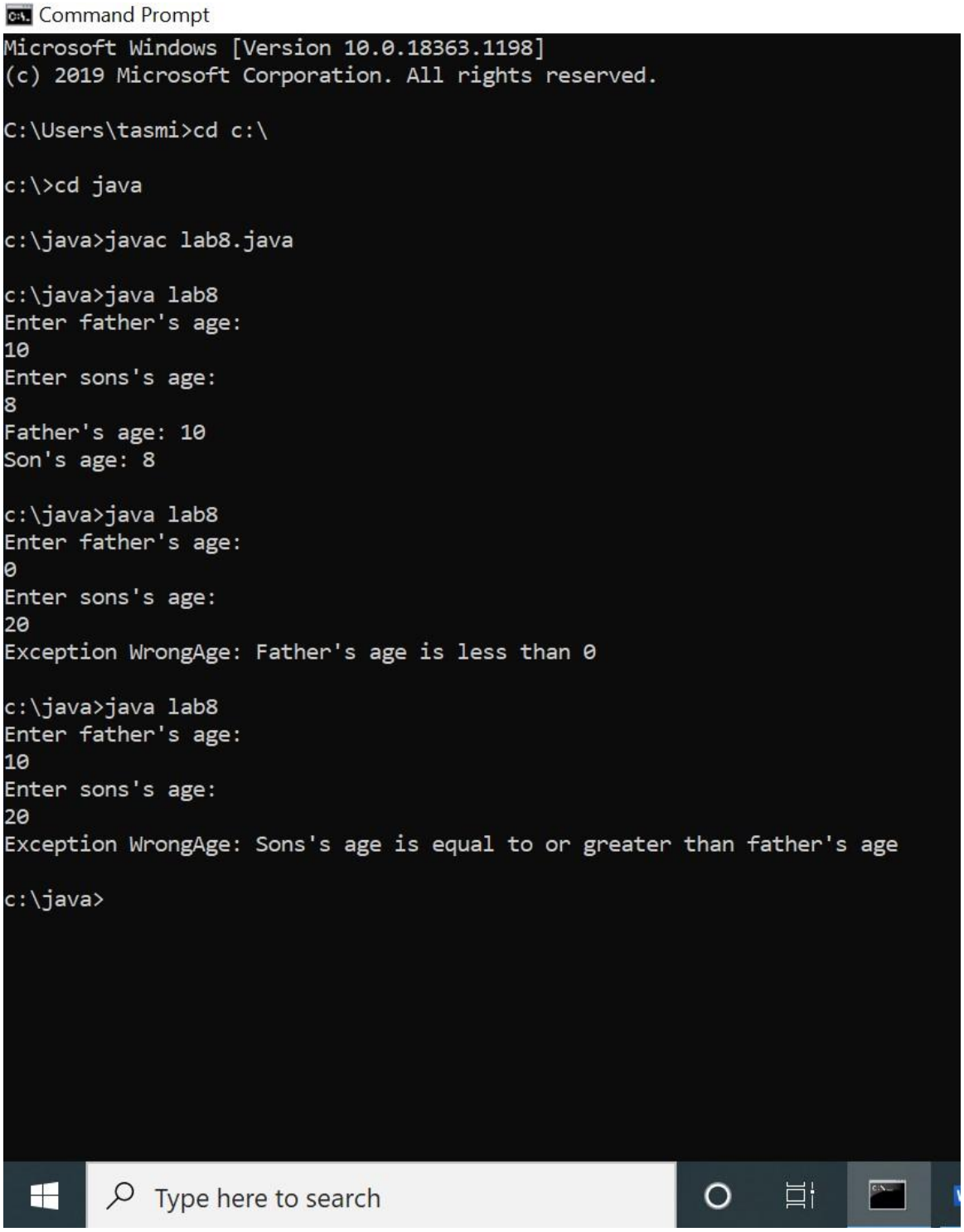
```
        else
        {
            fatherAge = fAge;
        }
    }
}

class Son extends Father
{
    int sonAge;

    Son (int fAge, int sAge) throws WrongAge
    {
        super(fAge);
        sonAge = sAge;

        if (sAge >= fAge)
        {
            throw new WrongAge("Son's age is equal to or
            greater than father's age");
        }
    }

    void Display()
    {
        System.out.println("Father's age :" + fatherAge);
    }
}

public class exp
{
    public static void main (String args[])
    {
        int fAge, sAge;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter father's age :");
```

```
        fAge = sc.nextInt();
        System.out.println("Enter son's age");
        sAge = sc.nextInt();

        try
        {
            Son son = new Son(fAge, sAge);
            son.display();
        }
        catch (WrongAge err)
        {
            System.out.println("Exception " + err);
        }
    }
}
```

```
import java.util.Scanner; class
WrongAge extends Exception {
    public WrongAge(String s){ super(s);


    }
}

class Father { int
    fatherAge;

    Father(int fAge) throws WrongAge{ if(fAge
        <=0){
            throw new WrongAge("Father's age is less than 0");

        } else{
            this.fatherAge = fAge;

        }
    }
}

class Son extends Father {
  int sonAge;
    Son(int fAge, int sAge) throws
        WrongAge{ super(fAge); sonAge=sAge;
        if(sAge >= fAge){
```

```
            throw new WrongAge("Sons's age is equal to or greater than
father's age");
        }


    }
    void Display(){
        System.out.println("Father's age: "+fatherAge);
        System.out.println("Son's age: "+sonAge); }
}


 class lab8 {
    public static void main(String[] args){
        int fAge,sAge;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter father's age: ");
        fAge = sc.nextInt();
        System.out.println("Enter sons's age:
        "); sAge = sc.nextInt(); try{
            Son son = new Son(fAge, sAge);
            son.Display();
        }catch(WrongAge err){
            System.out.println("Exception " + err); }
    }
}
```

OUTPUT:-

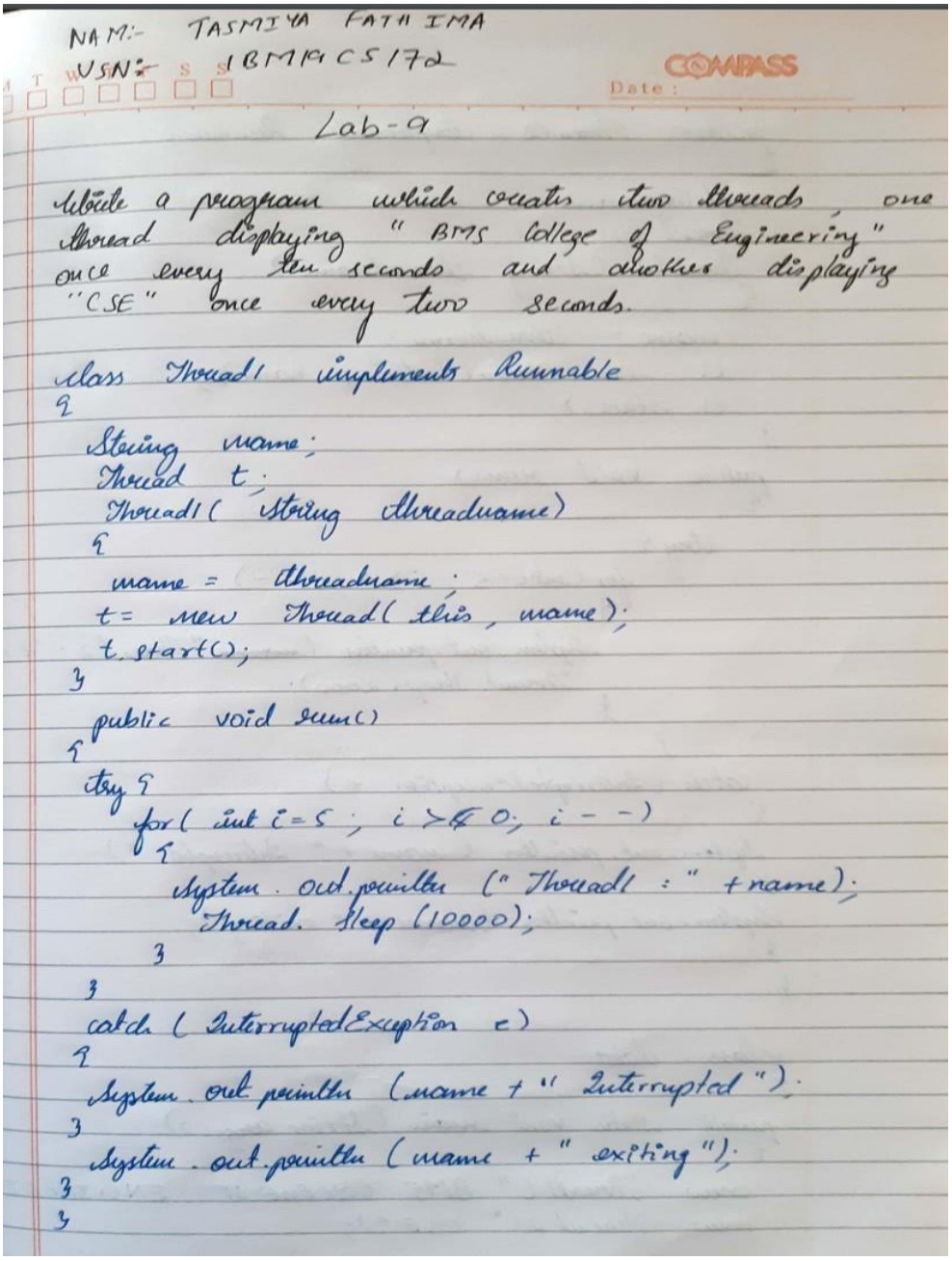


Lab Program: 9

Write a program which creates two threads, one thread displaying "BMS College of

Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

WRITE UP:-



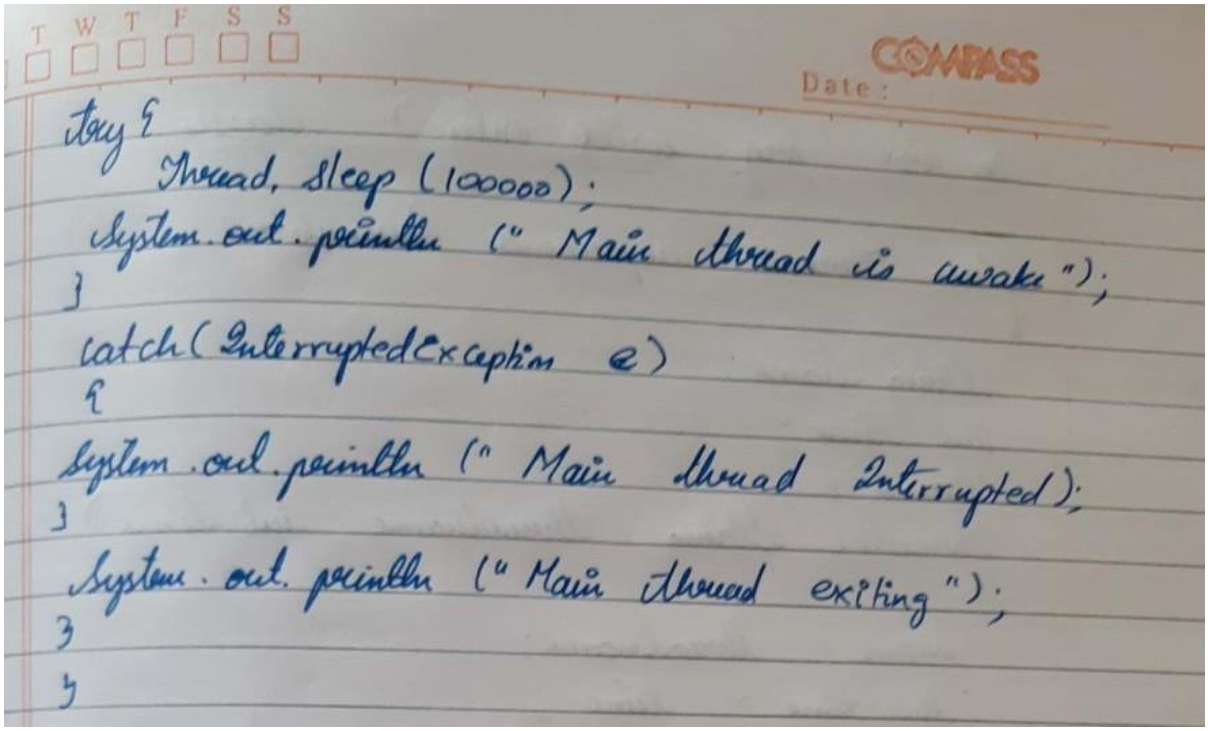
NAM:- TASMIYA FATHIMA
USN:- 1BM19CS172

Lab-9

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class Thread1 implements Runnable
{
  String name;
  Thread t;
  Thread1( String threadname)
  {
    name = threadname;
    t= new Thread( this, name);
    t.start();
  }
  public void run()
  {
   try{
     for( int i=5; i>=0; i--)
     {
       System.out.println(" Thread1 :" +name);
       Thread.sleep(10000);
     }
   }
   catch ( InterruptedException e)
   {
    System.out.println(name + " Interrupted ");
   }
   System.out.println(name + " exiting ");
  }
}
```

```
class Thread2 implements Runnable
{
  String name;
  Thread t1;
  Thread2 (String threadname)
  {
    name = threadname;
    t1 = new Thread (this, name);
    t1. start();
  }
  public void run()
  {
    try {
        for (int i=5; i>0; i--)
        {
          System.out.println (name "Thread 2 : "+ name);
          Thread. sleep(2000);
        }
    }
  catch (InterruptedException e)
  {
  System.out.println (name + " Interrupted ");
  }
  System.out.println (name + " exiting");
  }
}

class lab9
{
public static void main (String args[])
{
  new Thread1 (" BMS COLLEGE OF ENGINERING");
  new Thread2 (" CSE ");
```



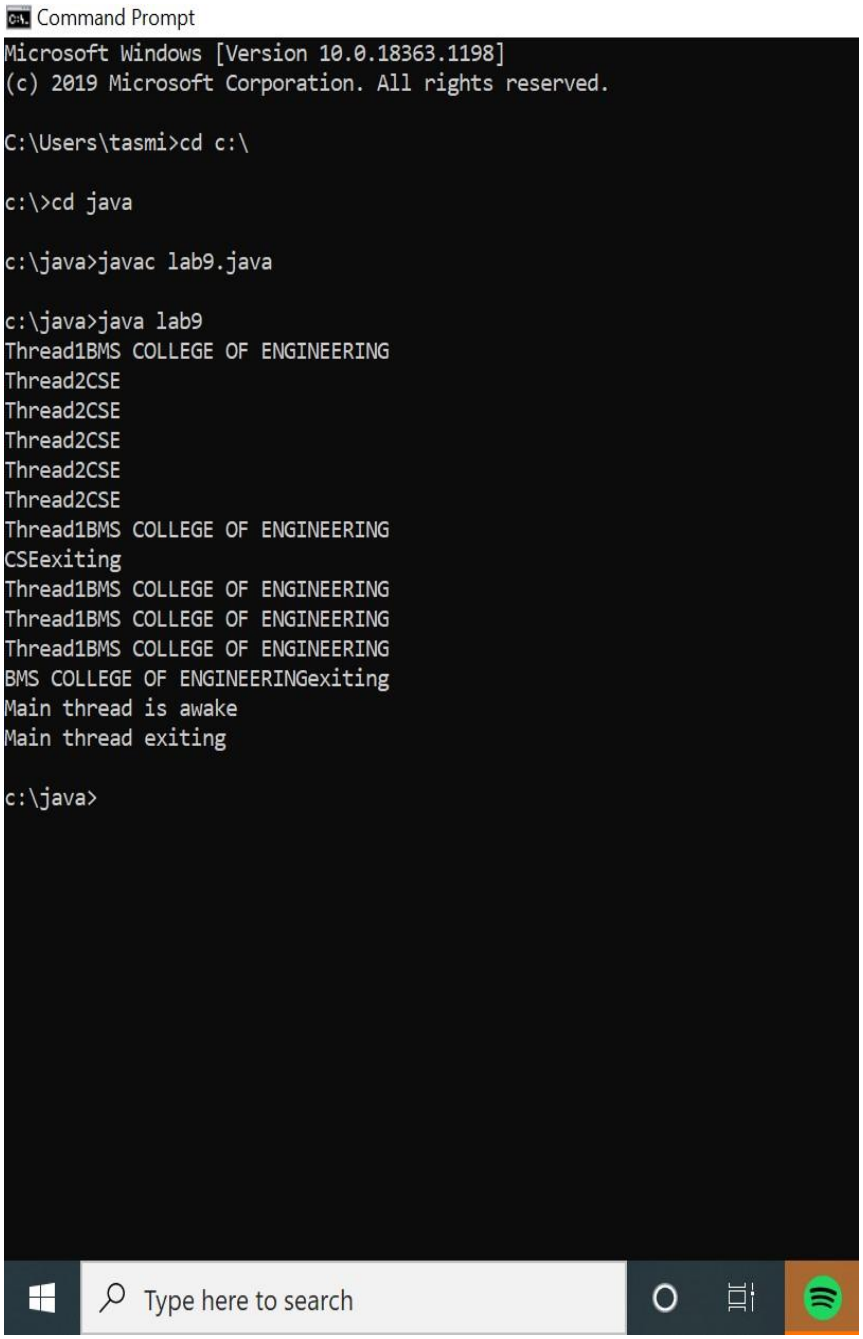

```
class Thread1 implements Runnable {

  String name;
  Thread t;
  Thread1(String threadname)
  {

    name=threadname; t=new
    Thread(this,name);
    t.start();
  }

  public void run()
  {
```

```
    try{
      for(int i=5;i>0;i--)
      {
        System.out.println("Thread1"+name);
        Thread.sleep(10000);
      }
    }

    catch(InterruptedException e)
    {

      System.out.println(name+"Interrupted");
    }

    System.out.println(name+"exiting");
  }
}



class Thread2 implements Runnable
{

  String name;
  Thread t1;
  Thread2(String threadname)
  {
```

```
    name=threadname; t1=new
    Thread(this,name);
    t1.start();
  }

  public void run()
  {
    try{ for(int
      i=5;i>0;i--)
      {

        System.out.println("Thread2"+name);
        Thread.sleep(2000);
      }
    }

    catch(InterruptedException e)
    {

      System.out.println(name+"Interrupted"); }


    System.out.println(name+"exiting");
  }
}

class lab9{
```

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

    new Thread1("BMS COLLEGE OF ENGINEERING");
    new Thread2("CSE"); try{
      Thread.sleep(100000);
      System.out.println("Main thread is awake");
    } catch(InterruptedException
    e)
    {
      System.out.println("Main thread Interrupted");
    }
      System.out.println("Main thread exiting");
  }
}
```

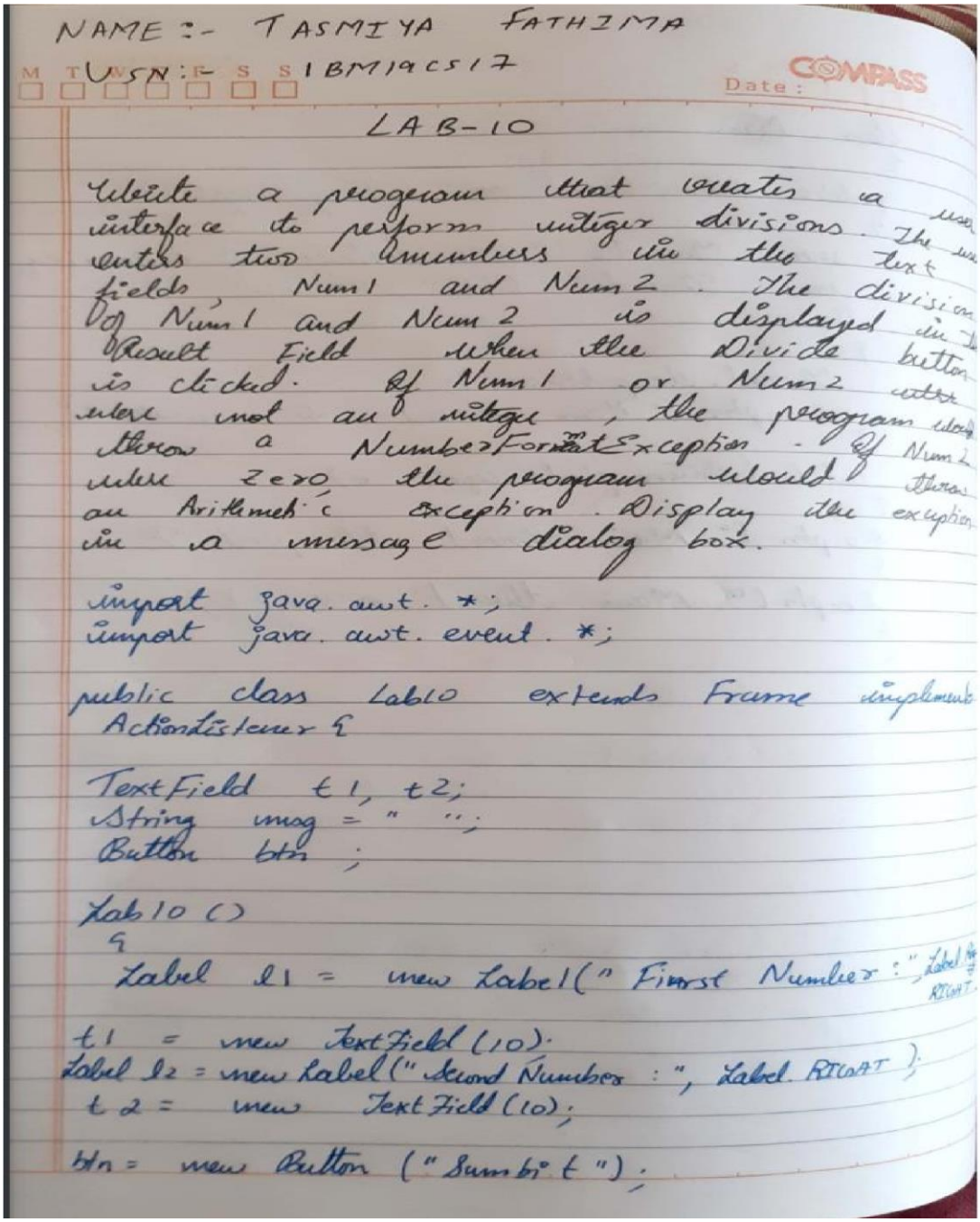
OUTPUT:-



Lab Program: 10

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

WRITE UP :-



NAME :- TASMIYA FATHIMA

USN : 1BM19CS17

Date :

LAB-10

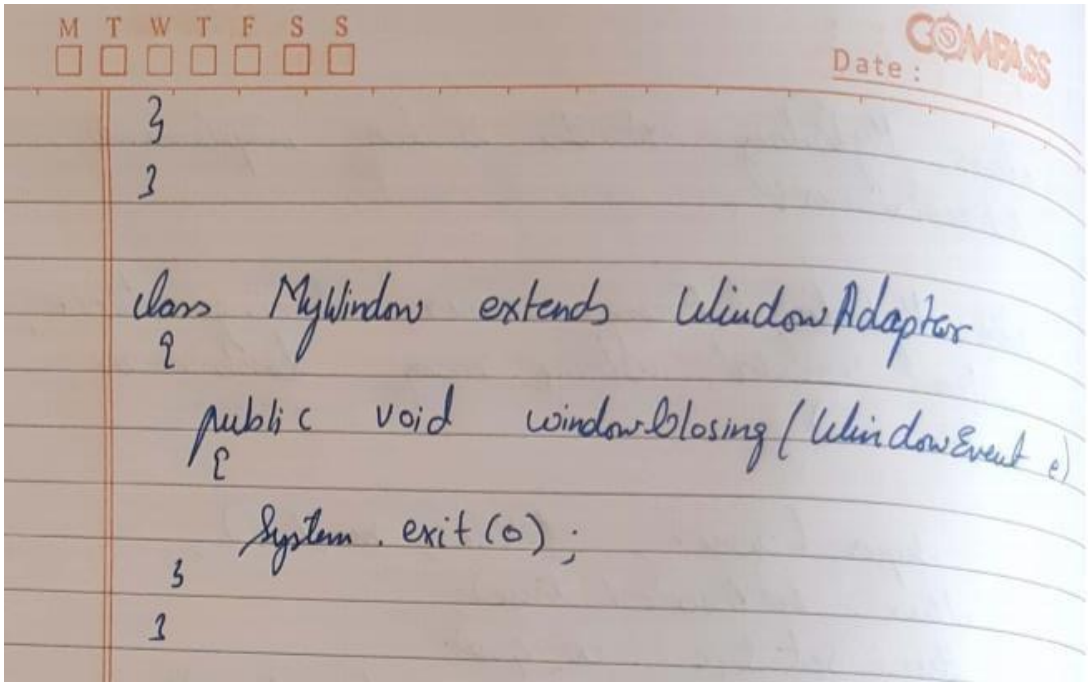
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an Arithmetic exception. Display the exception in a message dialog box.

```
import java.awt.*;
import java.awt.event.*;

public class Lab10 extends Frame implements
 ActionListener {

TextField t1, t2;
String msg = " ";
Button btn;

Lab10 ()
 {
  Label l1 = new Label(" First Number :", Label.RIGHT);

t1 = new TextField(10);
Label l2 = new Label(" Second Number :", Label.RIGHT);
t2 = new TextField(10);

btn = new Button ("Sum bit");
```

```
l1.setBackground(Color.YELLOW);
l2.setBackground(Color.YELLOW);

this.add(l1);
this.add(t1);
this.add(l2);
this.add(t2);

this.add(btn, BorderLayout.CENTER);
this.setVisible(true);
this.setSize(600, 20);
this.setLayout(new FlowLayout(FlowLayout.CENTER, 20
                                                 , 10));
btn.addActionListener(this);
addWindowListener(new MyWindow());
setBackground(Color.YELLOW);
}

@Override
public Insets getInsets()
{
  return new Insets(50, 10, 10, 20);
}

public void actionPerformed(ActionEvent e)
{
  String st1 = t1.getText();
  String st2 = t2.getText();
  double n1, n2;
  n1 = 0.0;
  n2 = 0.0;

if (st1.equals("") || st2.equals(""))
{
```

```
        msg = "You cannot leave the text
              elements blank";
    }
    else
    {
      try
        {
            n1 = Double.parseDouble(st1);
            n2 = Double.parseDouble(st2);

      try {
            double res = n1/n2;
            msg = "Result of division: " +res;
        }
      catch (ArithmeticException e1)
      {
          msg = e1.toString();
      }
    }
    catch (NumberFormatException e2)
    {
    msg = "Enter only numbers and not other
      strings";
    }
    }
new Dialog(this, "Result Dialog", false, msg,W,H);
}
public static void main(String args[])
{
    new Lab10();
}
}
```

```
class MyDialog extends Dialog implements
ActionListener{

public MyDialog (Frame owner, String title,
boolean modal, String msg, double n1,
double n2)
{
    super (owner, title, modal);
  this. setVisible (true);
  this. setSize (300, 400);
  this. setLayout ( new FlowLayout ());

Label l1 = new Label (" Updates on the
                        result :  ");

  this . add(l1);
this .add (new Label (" First number :" +n1));
this .add ( new Label ("Second number: "+n2));
this . add (new Label(msg));

Button b = new Button ("Close");
  this.add (b);
b. addActionListener ( this);
this . addWindowListener (new WindowAdapter()
{
public void windowClosing (WindowEvent e)
{
    dispose ();
  }
});
}
public void actionPerformed (ActionEvent e)
{
    disp dispose ();
```




```
import java.awt.*;
import java.awt.event.*;

public class Lab10 extends Frame implements ActionListener{
	TextField t1,t2;
	String msg="";
	Button btn;
	Lab10(){
		Label l1 = new Label("First Number: ",Label.RIGHT);
	t1 = new TextField(10);

		Label l2 = new Label("Second Number:

",Label.RIGHT);		t2 = new TextField(10);		btn = new

Button("Submit");			//Label l = new Label("Updates:");

		l1.setBackground(Color.YELLOW);
```

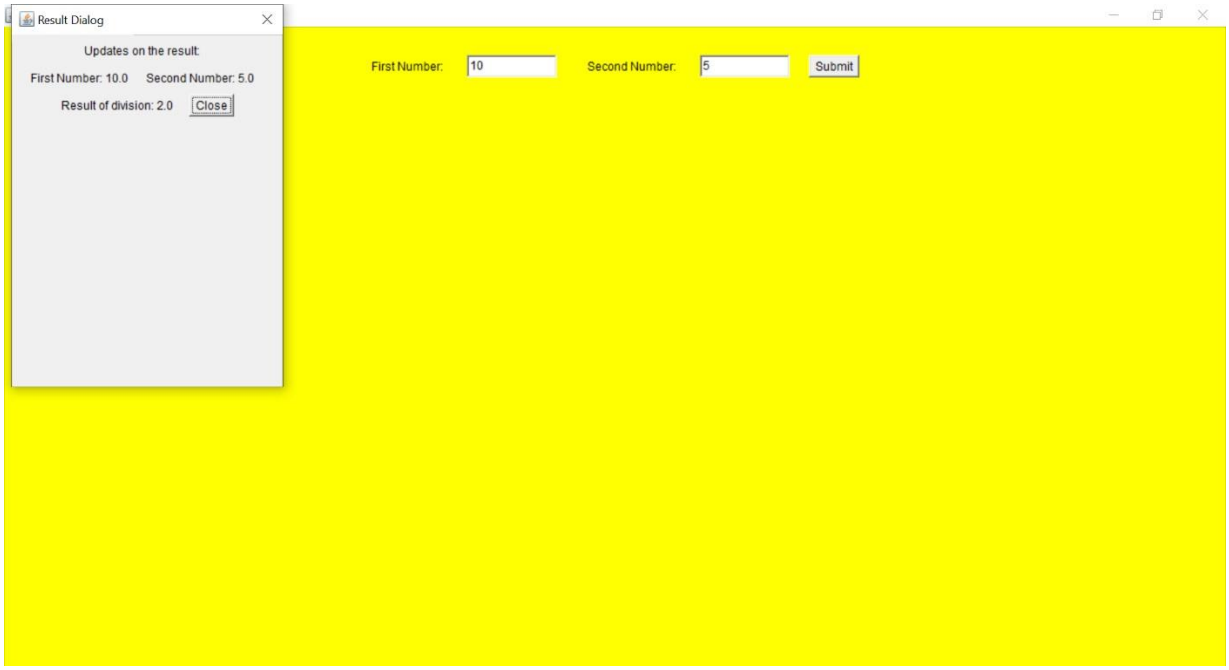
```
                l2.setBackground(Color.YELLOW);


//this.setResizable(false);
this.add(l1);                   this.add(t1);
        this.add(l2);
this.add(t2);


//the following command will make sure that the input char is not visible to the user
        //(it has been added just to demonstrate). Can be used for passwords.
                //t1.setEchoChar('*');
//t2.setEchoChar('#');
this.add(btn,BorderLayout.CENTER);
this.setVisible(true);


                this.setSize(600, 300);             this.setLayout(new
FlowLayout(FlowLayout.CENTER,20,10));
                //t1.addActionListener(this);
btn.addActionListener(this);            addWindowListener(new
MyWindow());            setBackground(Color.YELLOW);
                //System.out.println(BorderLayout.CENTER);
```

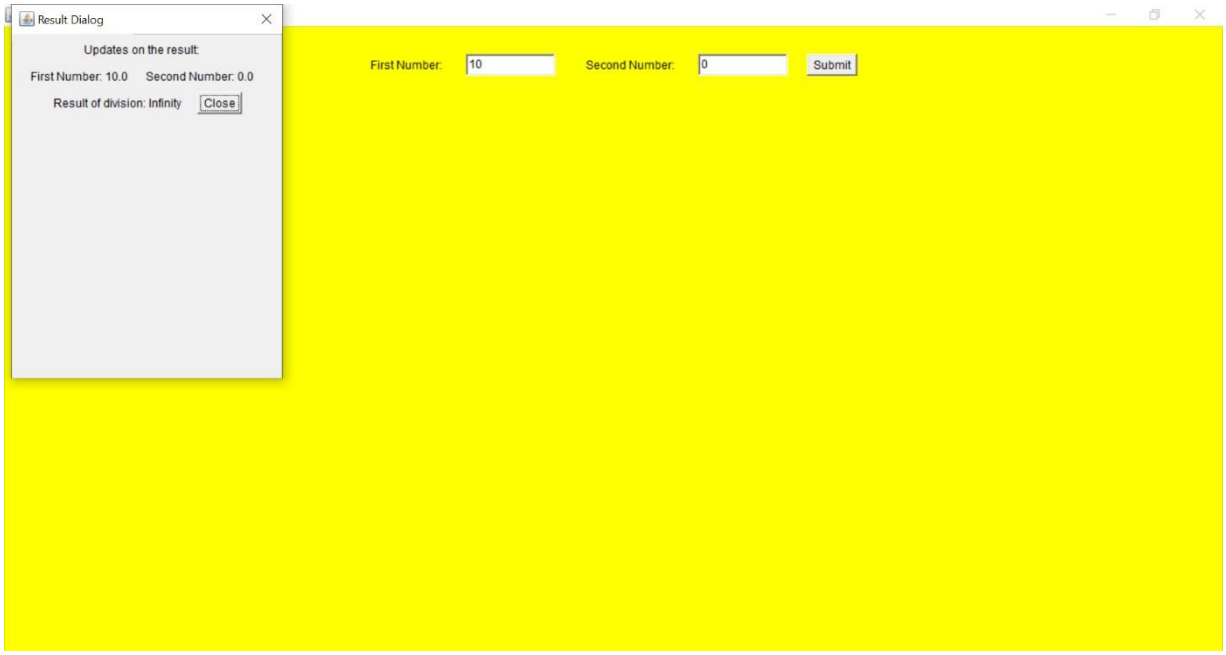
```
        }
        @Override
          public Insets getInsets() {
              return new Insets(50,10,10,20);


        }



            @Override
       public void actionPerformed(ActionEvent e) {


              String st1 = t1.getText();
              String st2 =
t2.getText();                  double
n1,n2;                n1 = 0.0;
n2 = 0.0;
              if(st1.equals("")||st2.equals("")) {

                    msg="You cannot leave the text elements blank";
              }else{
                    try {
                          n1 = Double.parseDouble(st1);
                    n2 = Double.parseDouble(st2);
                          try {
                                double res = n1/n2;
                                msg = "Result of division: "+res;
                          }catch(ArithmeticException e1) {
                    msg = e1.toString();
                          }
```

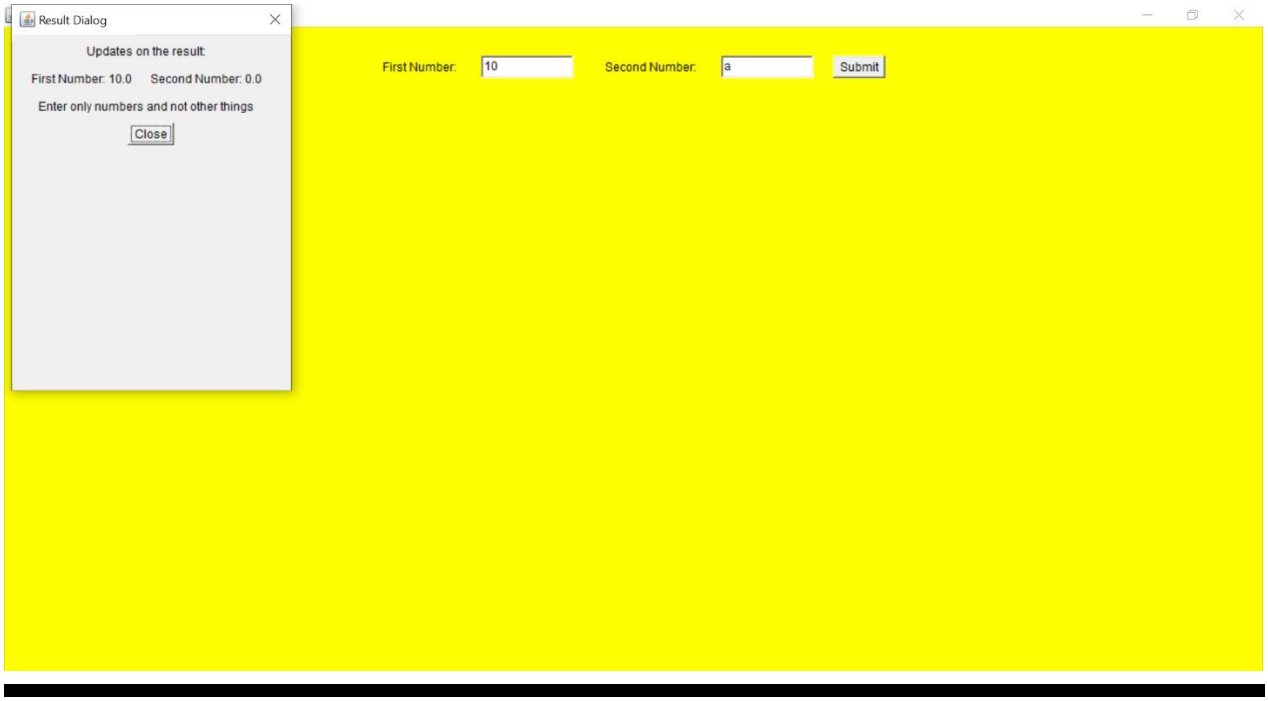
```
                }catch(NumberFormatException e2) {
        msg = "Enter only numbers and not other things";
                }
            }
            new MyDialog(this,"Result Dialog",false,msg,n1,n2);
        }
public static void main(String[] args) {
            new Lab10();
        }
}

class MyDialog extends Dialog implements ActionListener{

 public MyDialog(Frame owner, String title, boolean modal,String msg, double n1, double n2) {
            super(owner, title, modal);
this.setVisible(true);          this.setSize(300,
400);           this.setLayout(new
FlowLayout()); //System.out.println(owner);

            Label l1 = new Label("      Updates on the result:      ");
            //l1.setSize(300, 20);
this.add(l1);
            this.add(new Label("First Number: "+n1));
this.add(new Label("Second Number: "+n2));
this.add(new Label(msg));

            Button b = new Button("Close");
this.add(b);
```

```
            b.addActionListener(this);
this.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
                        dispose();
                    }

    });
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        dispose();

    }


}

class MyWindow extends WindowAdapter{
public void windowClosing(WindowEvent e) {
System.exit(0);
    }
}
```

OUTPUT


Result Dialog
Updates on the result:
First Number: 10.0 Second Number: 5.0
Result of division: 2.0
Close
First Number:
10
Second Number:
5
Submit


Result Dialog
Updates on the result:
First Number: 10.0 Second Number: 0.0
Result of division: Infinity
Close
First Number:
10
Second Number:
0
Submit


Result Dialog
Updates on the result:
First Number: 10.0 Second Number: 0.0
Enter only numbers and not other things
Close
First Number:
10
Second Number:
a
Submit