Name: SaiPraveen Marni

USN: 1BM19CS138

Dept: CSE

Section: C

Lab *batch*: C-2

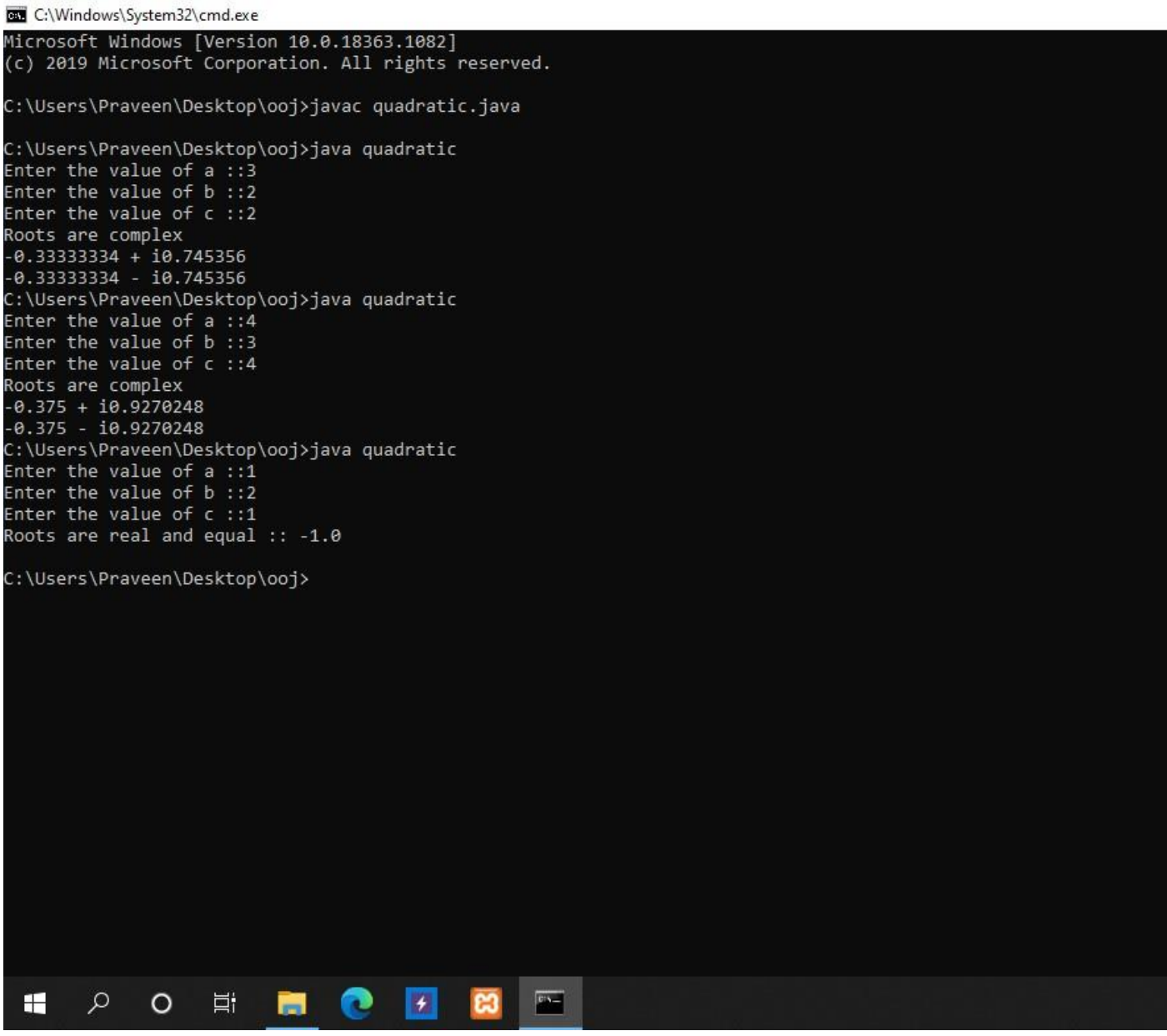
Lab Program-1(09/10/2020)

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.Scanner;
import static java.lang.Math.*;
 class quadratic
{
 public static void main(String args[])
  {
    quadratic obj = new quadratic();
    Scanner sc = new Scanner(System.in);
System.out.print("Enter the value of a ::");     float a =
sc.nextFloat();
```

```
 System.out.print("Enter the value of b ::");     float b =
sc.nextFloat();

   System.out.print("Enter the value of c ::");     float c =
sc.nextFloat();


    if (a == 0)
     {
               System.out.println("Invalid");
              return;
     }
            float d = b*b - 4*a*c;
floatsqrt_val = (float)Math.sqrt(abs(d));  float root1= (-
b + sqrt_val) / (2 * a);  float root2=(-b - sqrt_val) / (2 *
a);

    if(d == 0)
     {
       System.out.println("Roots are real and equal :: "+root1);
     }
    else if (d > 0)
     {
                System.out.print("Roots are real and different \n");
               System.out.print(root1 + "\n"+ root2);
     }
    else
     {
      System.out.print("Roots are complex \n");
 System.out.print( -b / ( 2 * a ) + " + i"+ sqrt_val/( 2 * a ) + "\n" + -b / ( 2 * a )+ " - i" + sqrt_val/( 2 * a ));
     }
  }
```

Output :



Lab Program-2(09/10/2020)

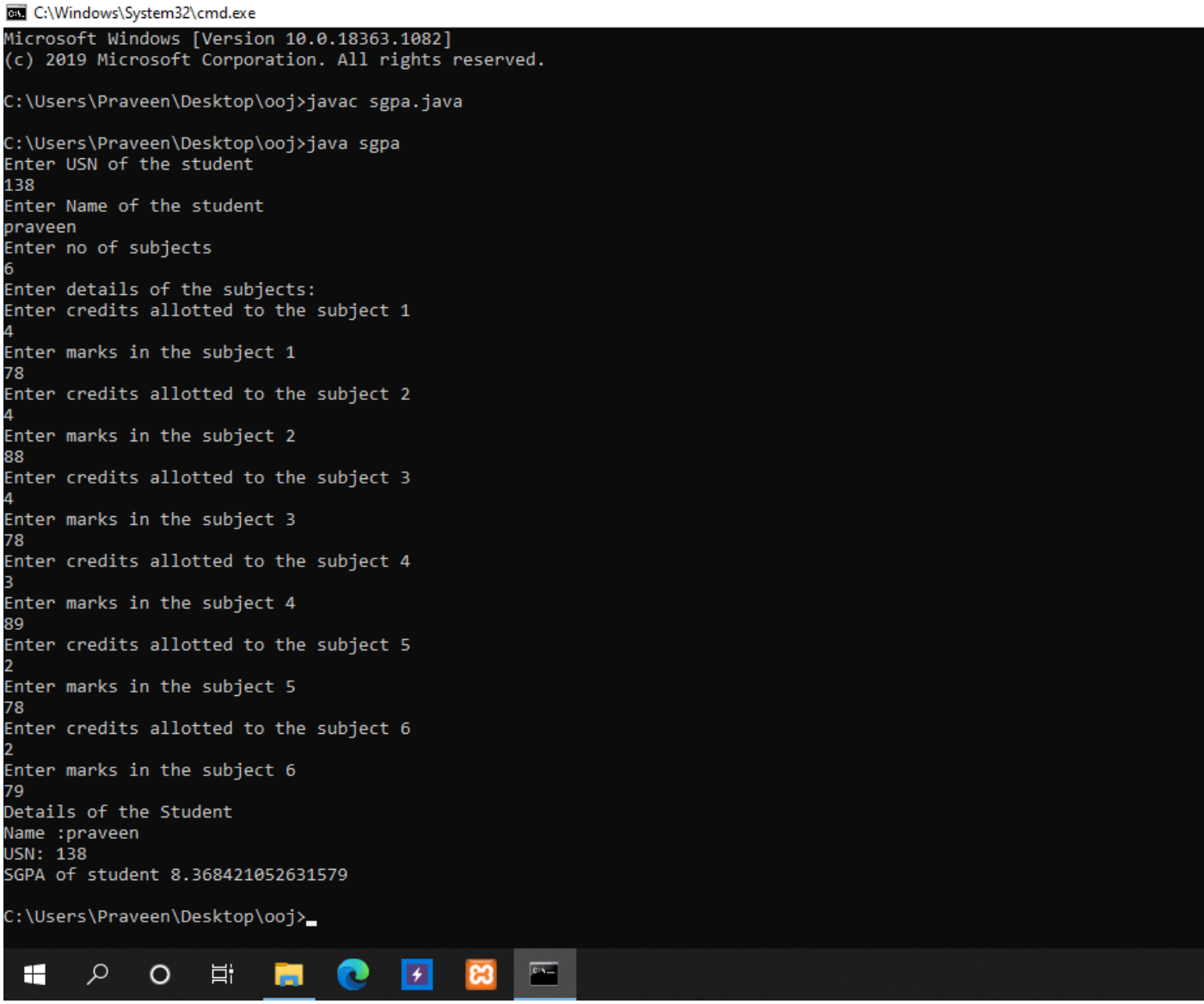
2) Develop a Java program to create a class student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class Student
{
 String USN;  String
name;
 int n;
 double SGPA = 0;
 int Credits = 0;
Scanner in = new Scanner(System.in);

 void Details()
 {
 System.out.println("Enter USN of the student");
 USN = in.nextLine();
 System.out.println("Enter Name of the student");  name =
in.nextLine();
 System.out.println("Enter no of subjects");
```

```
 n = in.nextInt();  int credits[] = new
int[n];  double marks[] = new
double[n];

 System.out.println("Enter details of the subjects:");

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

 {

  System.out.println("Enter credits allotted to the subject "+(i+1));   credits[i] =
in.nextInt();

  System.out.println("Enter marks in the subject "+(i+1));

  marks[i] = in.nextInt();

  Calculate(credits[i],marks[i],i);

 }

   }

    void Calculate(int credit,double mark,int j)

    {

 Credits = Credits + credit;

  if(mark>=90&&mark<=100)    SGPA =
SGPA + (10*credit);   else if(mark>=80
&& mark<=89)    SGPA = SGPA +
(9*credit);   else
if(mark>=70&&mark<=79)    SGPA =
SGPA + (8*credit);   else
if(mark>=60&&mark<=69)    SGPA =
SGPA + (7*credit);   else if(mark>=50
&& mark<=59)    SGPA = SGPA +
(6*credit);   else
if(mark>=40&&mark<=49)    SGPA =
SGPA + (5*credit);   else

    System.out.println("Failed in subject "+(j+1));

 }

 void Display()

 {

  System.out.println("Details of the Student");
```

```
 System.out.println("Name :"+name);
 System.out.println("USN: "+USN);
 System.out.println("SGPA of student "+(SGPA/Credits));
 }
}

public class sgpa
{
 public static void main(String args[])
 {
  Student s1 = newStudent();
 s1.Details();
 s1.Display();
 }
}
```

Output :



Lab Program-3(16/10/2020)

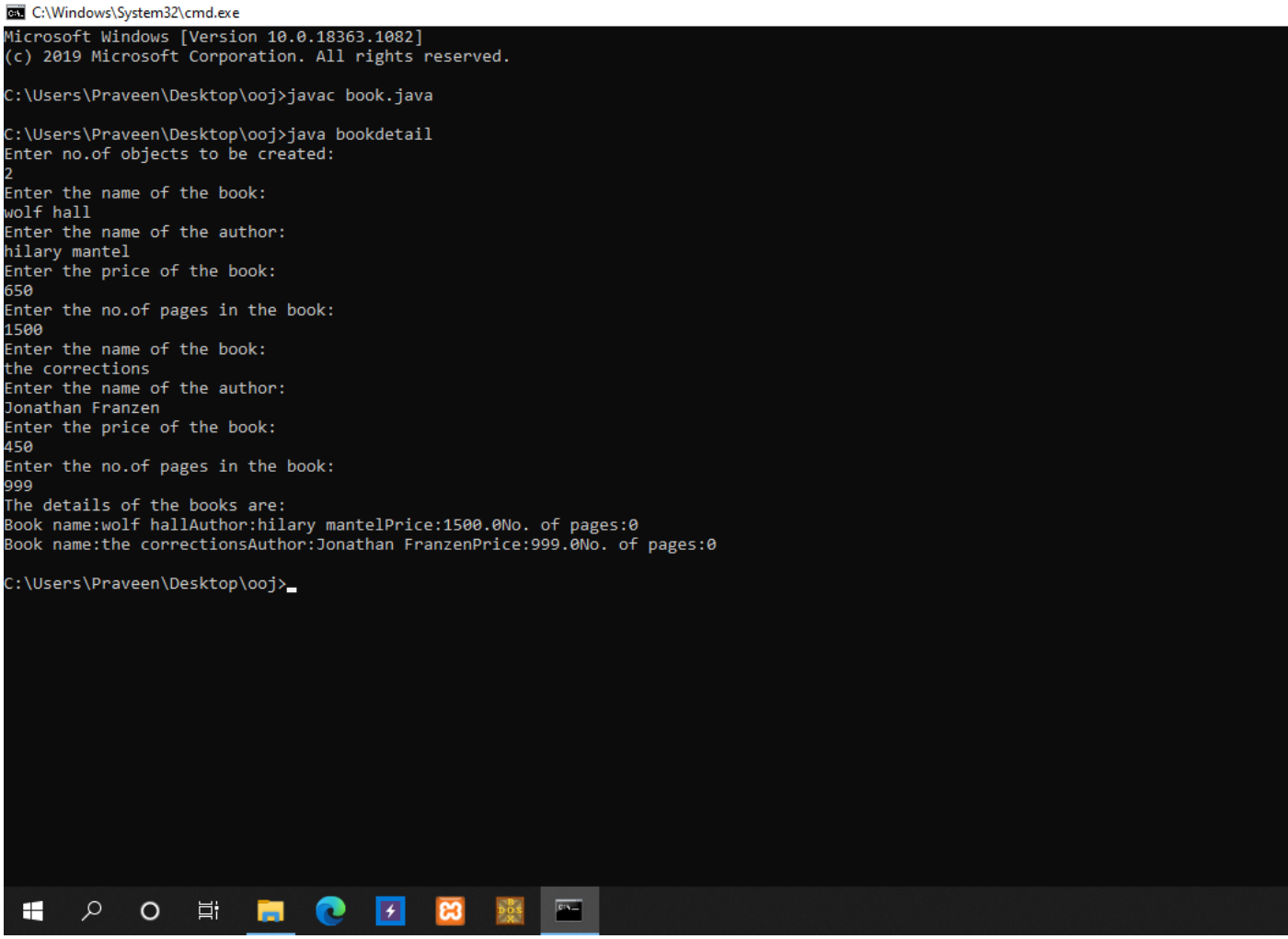
3) Create a class Book which contains four members: name, author, price, num_pages. Include a

Constructor to set the values for the members. Include a toString() method that could display the complete details of the Book. Develop a Java program to create n book objects.

```
import java.util.*;
class book
{
  String name;
  String author;
  double price;
  int num_pages;
 Scanner in=new Scanner(System.in);
book()
{
System.out.println("Enter the name of the book:");
name=in.nextLine();
System.out.println("Enter the name of the author:");
author=in.nextLine();
System.out.println("Enter the price of the book:");
price=in.nextDouble();
System.out.println("Enter the no.of pages in the book:");
price=in.nextInt();
}
public String toString()
{
return("Book name:"+name+"Author:"+author+"Price:"+price+"No. of pages:"+num_pages);
}
```

```
}


class bookdetail
{
public static void main(String[] args)
{
int i,n;
Scanner in=new Scanner(System.in);
System.out.println("Enter no.of objects to be created:");
n=in.nextInt();
book obj[];
obj=new book[n];
for(i=0;i<n;i++)
{
obj[i]=new book();
}
System.out.println("The details of the books are:");
for(i=0;i<n;i++)
{
System.out.println(obj[i].toString());
}
}
}
```

Output:



Lab Program-4(06/11/2020)

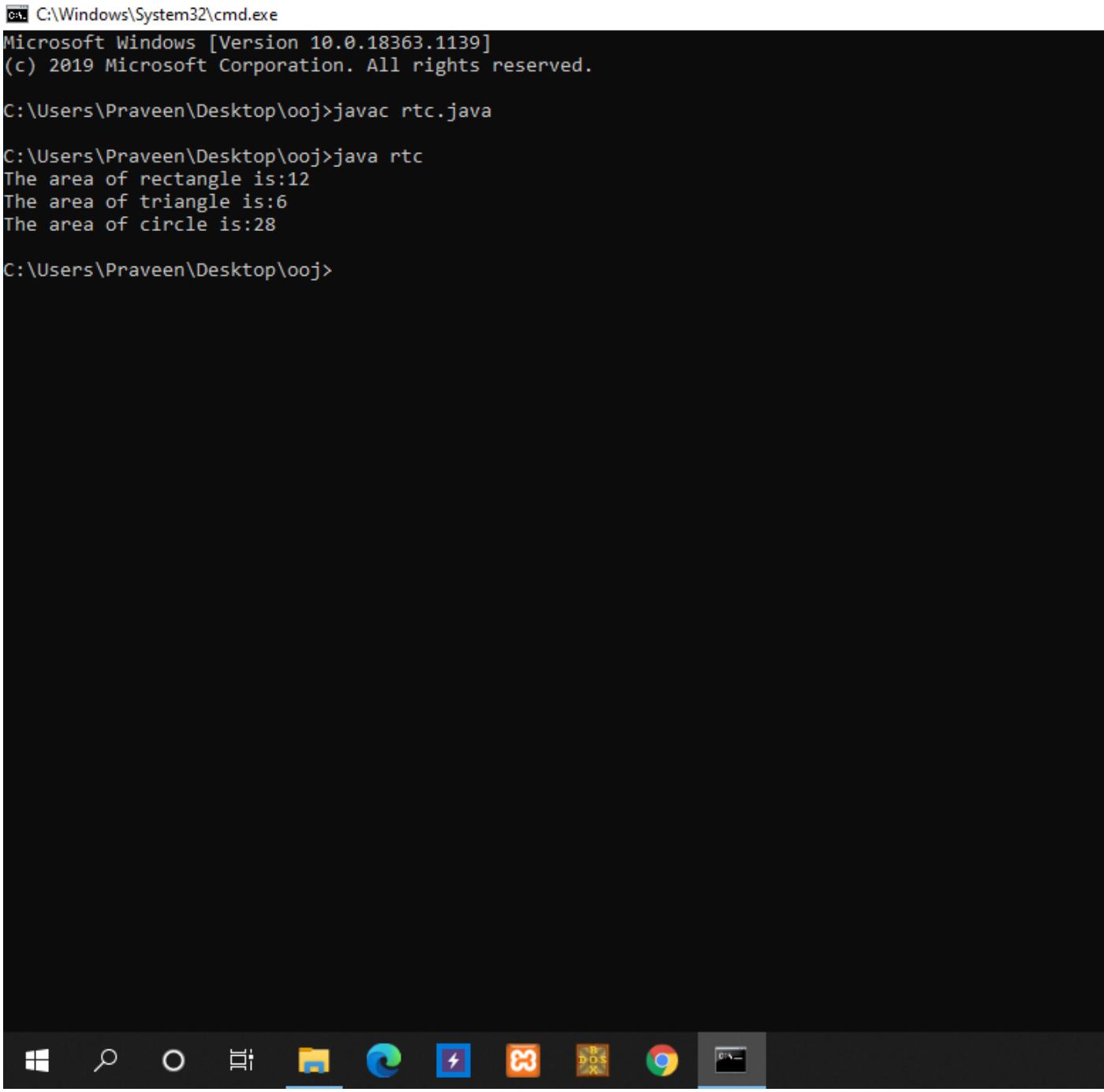
4) Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each

one of the classes contain only the method printArea( ) that prints the area of the given shape.

```
import java.util.Scanner;

abstract class shape

{

int a=3,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);
 }
}


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

        rectangle r=new rectangle();
        r.print_area();
        triangle t=new triangle();
        t.print_area();
        circle r1=new circle();
        r1.print_area();
 }
}
```

Output :



Lab Program-5(06/11/2020)

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 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: .Accept deposit from customer and update the balance. Display the balance. Compute and deposit the interest. Permit withdrawal and update the balance. Check for the minimum balance , impose the penalty if necessary and update the balance.

```
import java.util.Scanner;


class Account

{

String name;

int accountNo;
```

```
String accountType;
double balance;
Account(String name,int
accountNo,String accountType,double balance){
this.name = name;
this.accountNo = accountNo;
this.accountType = accountType;
this.balance = balance;
}
void DisplayStatus()
{
System.out.println("**"+this.accountType+"***");
System.out.println("Name: "+this.name);
System.out.println("Account no.:  "+this.accountNo);
System.out.println("Account Type:  "+this.accountType);
System.out.println("Balance: "+this.balance);
}
}

class SavAcct extends Account
{
double depositAmount;
double Withdrawmount;
SavAcct(String name,int accountNo,String accountType,double balance)
{
super(name,accountNo,accountType,balance);
}
```

```
static Scanner input = new Scanner(System.in);

private void checkBalance()

{

if(balance<0)

{

System.out.println("Transaction is not possible. Balance becomes less than zero");

balance+=Withdrawmount;

Withdrawmount=0;

Withdraw();

}

}

void CalInterest()

{

System.out.println("Interest To Be added");

System.out.println("Annual rate of interest: 4%");

System.out.println("Enter the tenure in terms of year");

int tenure = input.nextInt();

balance = balance*Math.pow(1.04, tenure);

}

void Deposit()

{

System.out.println("Enter the Deposit amount");

depositAmount = input.nextDouble();

balance+=depositAmount;

}

void Withdraw()

{
```

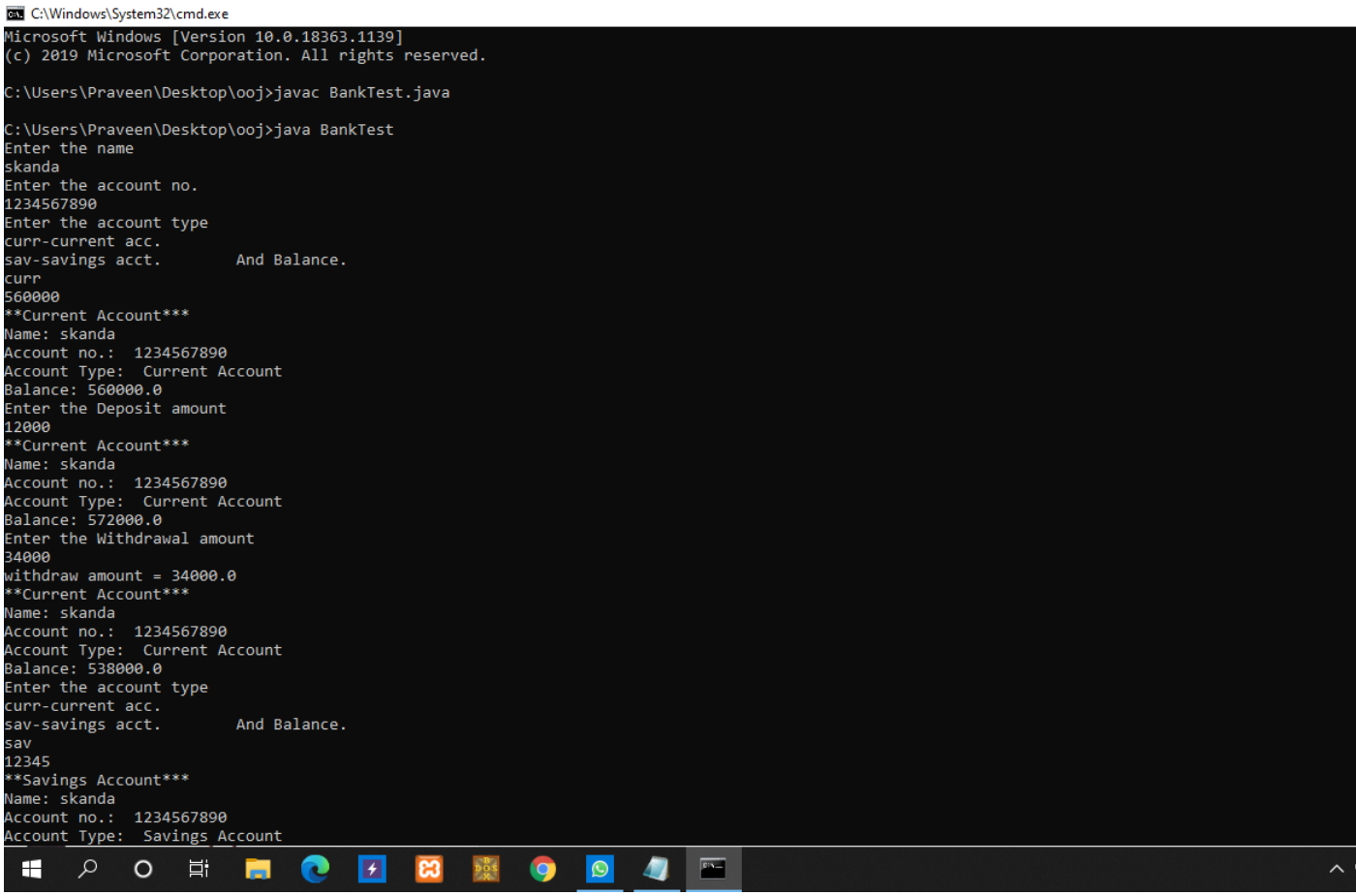
```
System.out.println("Enter the Withdrawal amount");
Withdrawmount = input.nextDouble();
balance-=Withdrawmount;
checkBalance();
System.out.println("Withdraw amount = "+Withdrawmount);
}
}
class CurrAcct extends Account
{
double minBalance = 1000;
double depositAmount;
double Withdrawmount;
static Scanner input = new Scanner(System.in);
CurrAcct(String name,int accountNo,String accountType,double balance)
{
super(name,accountNo,accountType,balance);
}
private void checkBalance()
{
if(balance<minBalance)
{
System.out.println("Transaction is not possible. Balance becomes less than minimum balance.");
balance+=Withdrawmount;
System.out.println("Do u still want to do the transaction with added service charges");
String ans = input.next();
if(ans.toLowerCase().equals("yes"))
{
```

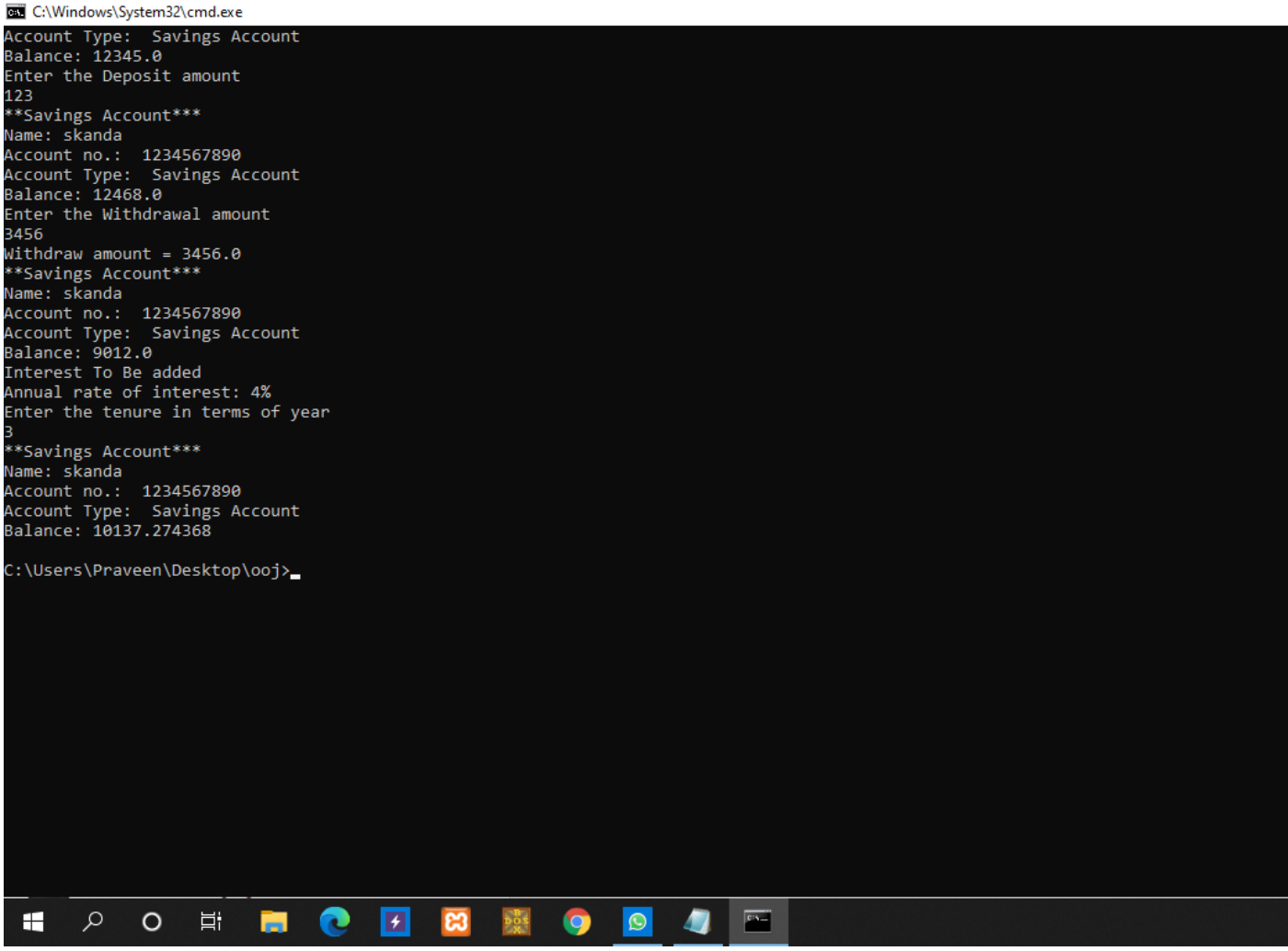
```
balance-=(Withdrawmount+(0.05*Withdrawmount)+1000);
System.out.println("ALERT: Negative balance.\nService Charge added: "+(0.05*Withdrawmount));
}
else
{
Withdrawmount=0;
}
}
}
void Deposit()
{
System.out.println("Enter the Deposit amount");
depositAmount = input.nextDouble();
balance+=depositAmount;
}
void Withdraw()
{
System.out.println("Enter the Withdrawal amount");
Withdrawmount = input.nextDouble();
balance-=Withdrawmount;
checkBalance();
System.out.println("withdraw amount = "+Withdrawmount);
}
}

public class BankTest
{
```

```
public static void main(String[] args)
{
Scanner in = new Scanner(System.in);
System.out.println("Enter the name");
String name = in.next();
System.out.println("Enter the account no.");
int num = in.nextInt();
int i=0;
while(i<2)
{
System.out.println("Enter the account type\ncurr-current acc.\nsav-savings acct.\t And Balance.");
String type = in.next();
if(type.equals("curr"))
{
double bal = in.nextInt();
CurrAcct c1 = new CurrAcct(name,num,"Current Account",bal);
c1.DisplayStatus();
c1.Deposit();
c1.DisplayStatus();
c1.Withdraw();
c1.DisplayStatus();
}
else if(type.toLowerCase().equals("sav"))
{
double bal = in.nextInt();
SavAcct s1 = new SavAcct(name,num,"Savings Account",bal);
s1.DisplayStatus();
```

```
s1.Deposit();

s1.DisplayStatus();

s1.Withdraw();

s1.DisplayStatus();

s1.CalInterest();

s1.DisplayStatus();

}

i++;

}

in.close();

}

}
```

Output:





Lab Program-*6*(*20*/*11*/2020)

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.

Student.java

```
packa
ge
cie;
        import java.util.*;
        public class student
        {
              public String usn;
              public String name;
              public int sem;

              public void read()
              {
                    Scanner sc=new Scanner(System.in);
              System.out.print("Enter usn of the student : ");
                    usn=sc.next();
              System.out.print("Enter name of the student : ");
                    name=sc.next();
        System.out.print("Enter semester of the student : ");
                    sem=sc.nextInt();
              }
        }
```

Internals.java

```
 packa
 ge
 cie;
         import java.util.*;

         public class internals extends student
         {
               public int[] cie_m=new int[3];

               public void read()
               {
                     super.read();
                   Scanner sc=new Scanner(System.in);
         System.out.println("Enter the CIE marks : ");
                    for(int i=0;i<3;i++)
```

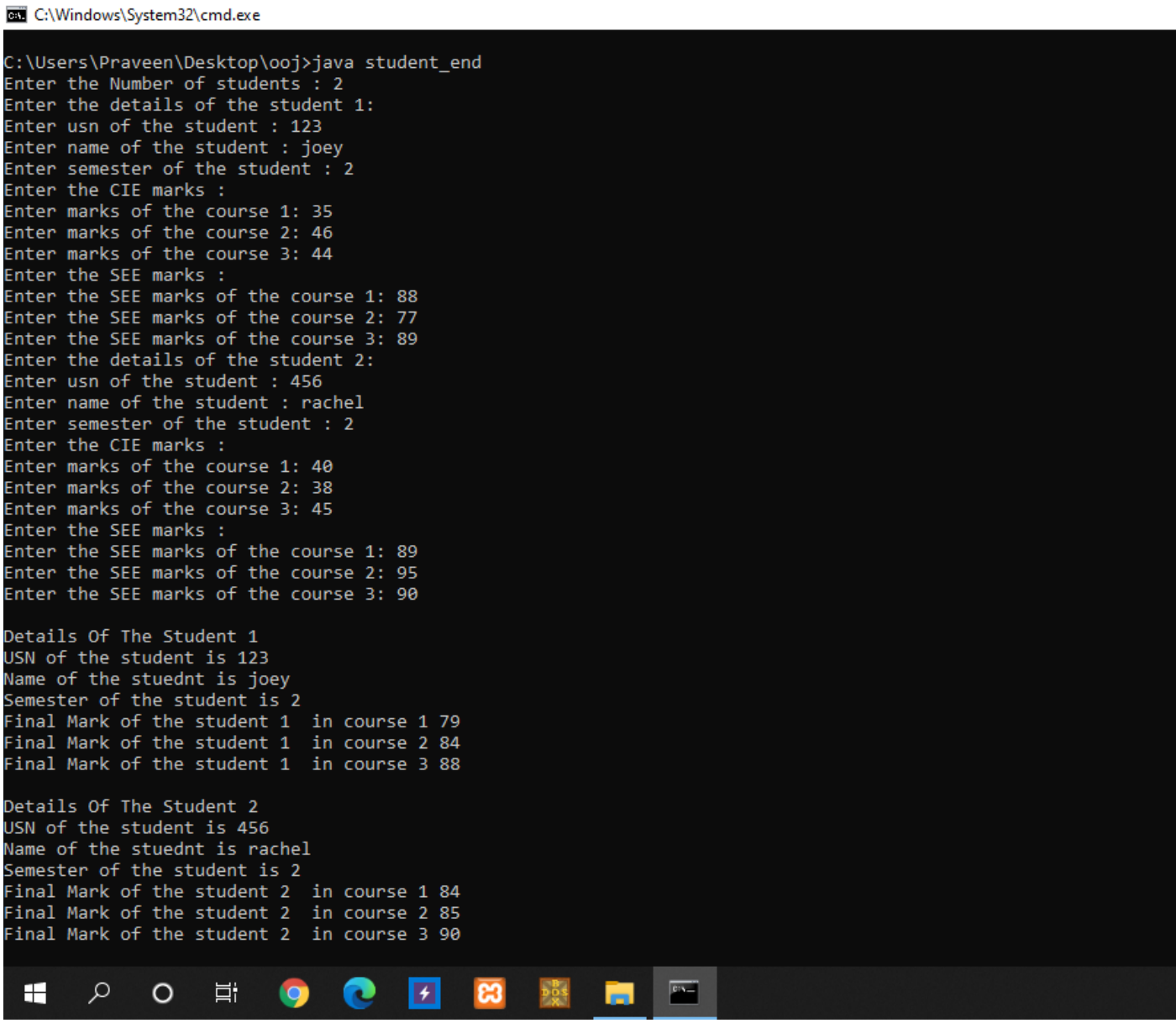
```
          {
System.out.print("Enter marks of the course " +
(i+1)+": ");
           cie_m[i]=sc.nextInt();
          }
     }
     public void display()
     {
System.out.println("USN of the student is " +
usn);
System.out.println("Name of the stuednt is " +
name);
System.out.println("Semester of the student is "
+ sem);
                                                  }
package                                           }
see;
         import java.util.*;
         import java.io.*;

         import java.lang.*;

         public class external extends
         cie.student
         {
              public int[] see_m=new
         int[3];
              public int[] mar;
              public void read()
              {
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter the SEE
         marks : ");
                   for(int i=0;i<3;i++)
                   {
         System.out.print("Enter the SEE
         marks of the course " + (i+1)+":
         ");
         see_m[i]=sc.nextInt();
                   }
              }
         }
Externals.java
```

Main_stu.java

```
Import
java.util
.*;
            import java.io.*;
            import java.lang.*;
            import cie.*;
            import see.*;


            public class student_end
            {
            public static void main(String[]
            args)
                  {
                        int n;
            Scanner sc=new Scanner(System.in);
                        int final_mark;
            System.out.print("Enter the Number
            of students : ");
                        n=sc.nextInt();
            internals[] in=new internals[n];
            external[] ex=new external[n];
            internals ob1=new internals();
            external ob2=new external();
                        ob2.mar=new int[n];

                        for(int i=0;i<n;i++)
                        {
            System.out.println("Enter the
            details of the student " + (i+1)+":
            ");
                  in[i]=new internals();
                  in[i].read();
                  ex[i]=new external();
                  ex[i].read();
                        }
                System.out.println();
                        for(int i=0;i<n;i++)
                        {
            System.out.println("Details Of The
            Student " + (i+1));
            System.out.println("USN of the
            student is " + in[i].usn);

            System.out.println("Name of the
            stuednt is " + in[i].name);
```

```
System.out.println("Semester of the
student is " + in[i].sem);
for(int j=0;j<3;j++)
                {
final_mark=in[i].cie_m[j]+((ex[i].s
ee_m[j])/2);

System.out.println("Final Mark of
the student " + (i+1) + " " + " in
course " + (j+1) + " " +
final_mark);

                }
```

Output:



Lab Program-7(27/11/2020)

Write a program to demonstrate generics with multiple object parameters.

```
importjava.io
.*;
                import java.lang.*;
```

```
import java.util.*;

class gen<T>
{
      T ob;
      gen(T o)
      {
            ob=o;
      }
      T getob()
      {
            return ob;
      }
      void showtype()
      {
System.out.println("Type of T is " +
ob.getClass().getName());
      }
}

class generic
{
      public static void main(String[] args)
      {
            String n;
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the Integer
Number to Be Displayed Using the generic style");
            n=sc.next();
            gen<Integer> ob1=new
gen<Integer>(Integer.parseInt(n));
            ob1.showtype();
            int val=ob1.getob();
            System.out.println("Value is: " + val);

            System.out.println();

            System.out.println("Enter the String to
Be Displayed Using the generic style");
            n=sc.next();
            gen<String> ob2=new gen<String>(n);
            ob2.showtype();
            String x=ob2.getob();
            System.out.println("Value : " + x);

            System.out.println();
```

```
System.out.println("Enter the Double Number to Be
Displayed Using the generic style");
n=sc.next();
gen<Double> ob3=new
gen<Double>(Double.parseDouble(n));
ob3.showtype();
double ans=ob3.getob();
System.out.println("Value : " + ans);

}
}
```

Output:

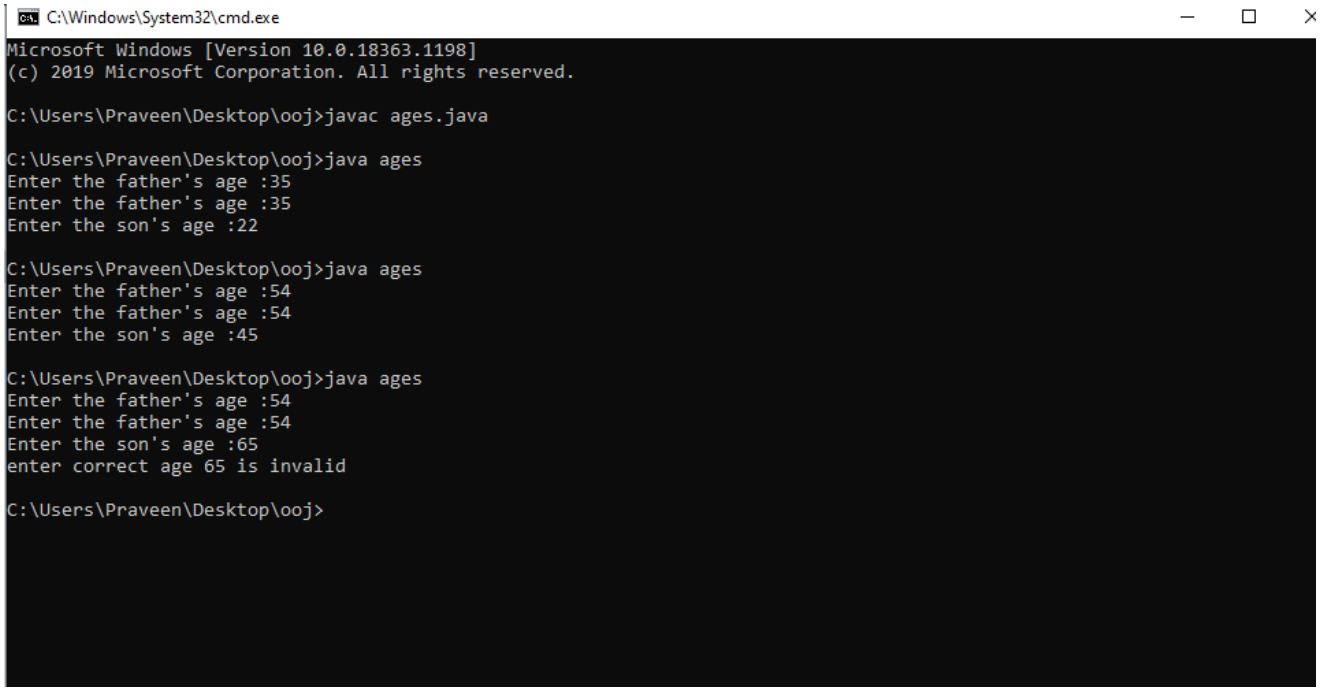
```
D:\Java file\OOJ-lab-codes>cd Lab program 7

D:\Java file\OOJ-lab-codes\Lab program 7>javac Lab7.java

D:\Java file\OOJ-lab-codes\Lab program 7>java Lab7
The type of object java.lang.Integer
Object Value 88
The type of object java.lang.Double
Object Value 88.889
The type of object java.lang.String
Object Value abcdefghij
The type of object java.lang.Integer
Object Value 12
```

Lab Program-8(*27*/11/2020)

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.

```
importjava.util.*;
                    class WrongAge extends Exception {
                     int detail;
                     WrongAge(int a) {
                      detail = a;
                     }
                     public String toString() {
                      return "enter correct age "+detail+" is
                    invalid";
                     }

                    }
                    class father{
                     public int age;
                     Scanner in =new Scanner(System.in);
                     father() throws WrongAge{
                    System.out.print("Enter the father's age :");
                      age= in.nextInt();
                      if(age<0)
                        throw new WrongAge(age);
                     }
                    }
                    class son extends father{
                     Scanner in =new Scanner(System.in);
                     int fage;
                     son(father f) throws WrongAge{
                      this.fage=f.age;
                      System.out.print("Enter the son's age :");
                      this.age= in.nextInt();
                      if(this.age<0)
                        throw new WrongAge(age);
                      if(this.age>f.age)
                        throw new WrongAge(age);
                     }

                    }
                    class ages{
                     public static void main(String[] args){
                      try{
                       father f= new father();
                       son s= new son(f);
                      }
                      catch(Exception e){
                        System.out.println(e);
                     }
                     }
```

Output:



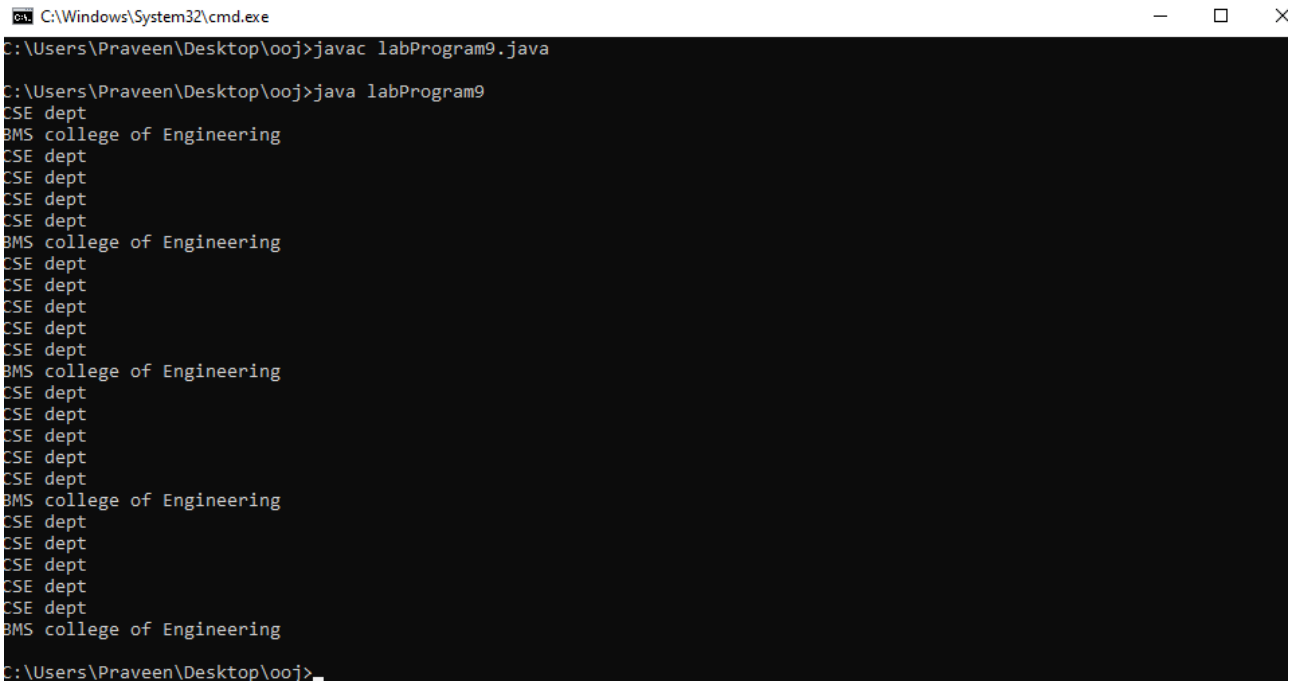
Lab Program-9(*11*/12/2020)

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
implementsRunnable{
                        Thread t;
                        String name;
                        Thread1(String name){
                            this.name = name;
                            t = new Thread(this,this.name);
                            t.start();
                        }
                        public void run(){
                            try{
                                for(int i=0;i<20;i++){
                                System.out.println("CSE dept");
                                    Thread.sleep(2000);
                                }
                            }
```

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

class Thread2 implements Runnable{
    Thread t;
    String name;
    Thread2(String name){
        this.name = name;
        t = new Thread(this,this.name);
        t.start();
    }
    public void run(){
        try{
            for(int i=0;i<5;i++){
 System.out.println("BMS college of
Engineering");
                Thread.sleep(10000);
            }
        }
catch(InterruptedException e){
            System.out.println(e);
        }
    }
}

class labProgram9{
    public static void main(String[] args){
   Thread1 obj1 = new Thread1("Dept. name");
 Thread2 obj2 = new Thread2("College name");
//System.out.println(obj1.name+"
"+obj1.t.isAlive());
//System.out.println(obj2.name+"
"+obj2.t.isAlive());
        try{
            obj1.t.join();
            obj2.t.join();
        }
catch(Exception e){

System.out.println("Interrupted");
        }
    }
}
```

Output:



Lab Program-10(*11*/12/2020)

Write a program that creates a user interface to perform integer divisions of Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and 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.BorderLa
yout;
                    import java.awt.Button;
                                 import java.awt.Color;
                                 import java.awt.Dialog;
                                 import java.awt.FlowLayout;
                                 import java.awt.Frame;
                                 import java.awt.Graphics;
                                 import java.awt.Insets;
```

```
import java.awt.Label;
import java.awt.TextField;
import
java.awt.event.ActionEvent;
import
java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import
java.awt.event.TextListener;
import
java.awt.event.WindowAdapter;
import
java.awt.event.WindowEvent;

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(BorderLayou
t.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: