B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfillment of the

requirements for Lab Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

ADITYA KUMAR

1BM22CS018

Department of Computer Science and

Engineering, B.M.S College of Engineering, Bull

Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

INDEX

PROGRAM_1:

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

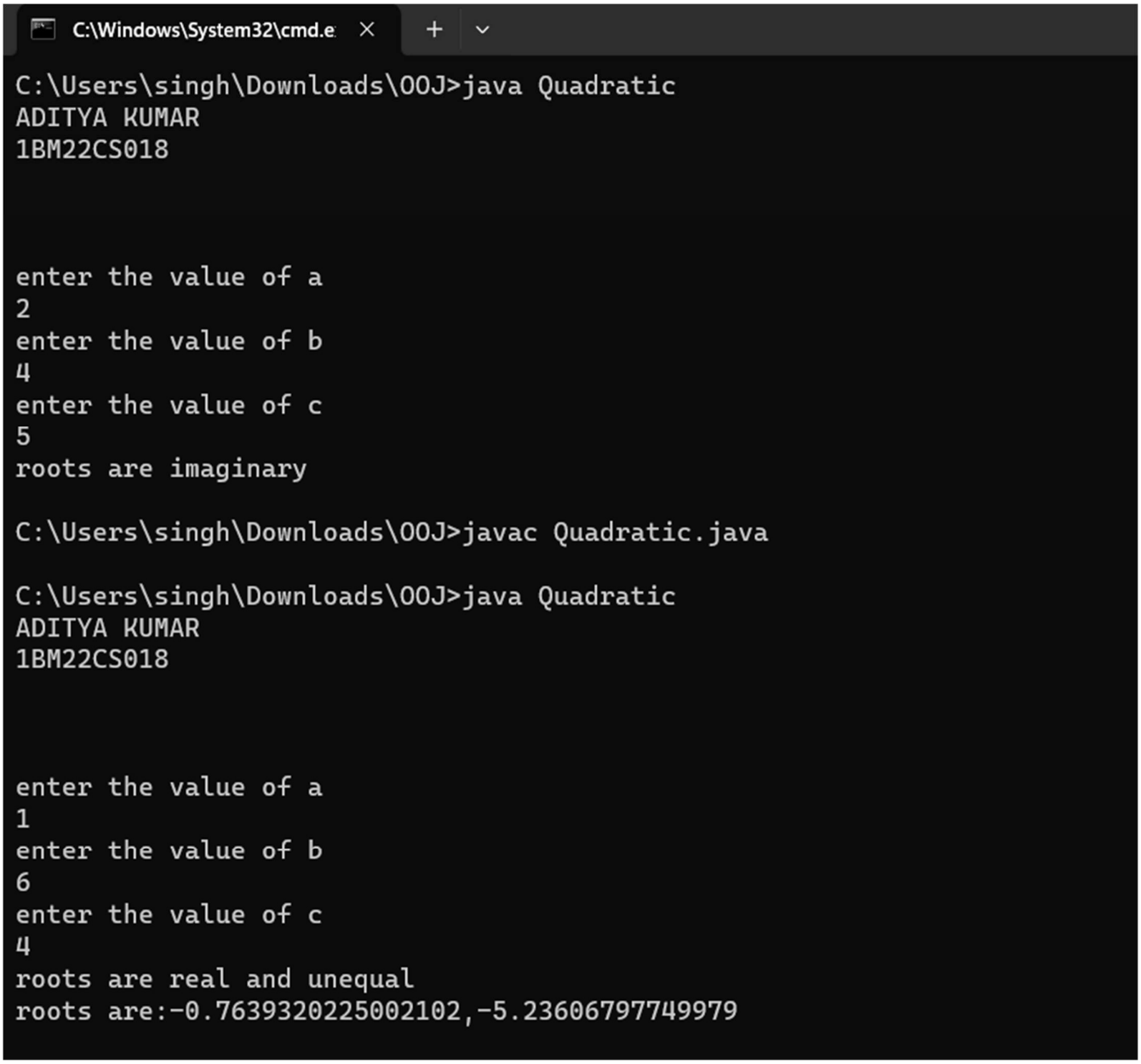
Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a

message stating that there are no real solutions. code:

```
import java.util.Scanner;

class Quadratic

{

public static void main(String args[])

{

System.out.println("ADITYA KUMAR");

System.out.println("1BM22CS018\n");

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



float a,b,c,d;

double x1,x2;  //math.sqrt returns double

Scanner obj =new Scanner(System.in);

System.out.println("enter the value of a");

a=obj.nextFloat();

System.out.println("enter the value of b");

b=obj.nextFloat();

System.out.println("enter the value of c");

c=obj.nextFloat();

d=b*b-(4*a*c);

if(a==0)
```

```
{
System.out.println("invalid input");
}
else
{
if(d==0)
{
System.out.println("roots are real and equal");
x1=x2=(-b)/(2*a);
System.out.println("roots are:"+x1 +","+x2);
}
else if(d>0)
{
System.out.println("roots are real and unequal");
x1=((-b)+Math.sqrt(d))/(2*a);
x2=((-b)-Math.sqrt(d))/(2*a);
System.out.println("roots are:"+x1 +","+x2);
}
else
{
System.out.println("roots are imaginary");
}
}
}
```

}



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. code:

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

```
class Student {
    String usn;
    String name;
    int[] credits = new int[8];
    int[] marks = new int[8];
    // Method to accept student details
    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = scanner.nextLine();

        System.out.print("Enter Name: ");
        name = scanner.nextLine();

        System.out.println("Enter details for each subject : \n");

        for (int i = 0; i < credits.length; i++) {
            System.out.print("\nEnter credits for Subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("\nEnter marks for Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
        scanner.close();
    }

    // Method to calculate SGPA
    public double calculateSGPA() {
        int totalCredits = 0;
        int weightedSum = 0;
```

```
        double ans;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            int gradePoints;

            gradePoints = (marks[i]/10)+1;

            if(gradePoints == 11){
                gradePoints=10;
            }

            else if(gradePoints<=4){
                gradePoints=0;
            }

            weightedSum += gradePoints * credits[i];
        }

        ans = (double) weightedSum / (double) totalCredits;
        return ans;
    }
}

public class sgpa {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Create a Student object
        Student student = new Student();

        // Accept and display details
```

```
        student.acceptDetails();


        System.out.println("\nStudent Details :");


        System.out.println("USN : " + student.usn);
        System.out.println("Name : " + student.name);
        // Calculate and display SGPA
        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);
        scanner.close();
    }
}
```

OUTPUT:


C:\Windows\System32\cmd.e
Enter USN: 1BM22CS018
Enter Name: ADITYA KUMAR
Enter details for each subject :
Enter credits for Subject 1: 4
Enter marks for Subject 1: 98
Enter credits for Subject 2: 4
Enter marks for Subject 2: 87
Enter credits for Subject 3: 4
Enter marks for Subject 3: 85
Enter credits for Subject 4: 3
Enter marks for Subject 4: 82
Enter credits for Subject 5: 3
Enter marks for Subject 5: 90
Enter credits for Subject 6: 3
Enter marks for Subject 6: 79
Enter credits for Subject 7: 1
Enter marks for Subject 7: 90
Enter credits for Subject 8: 1
Enter marks for Subject 8: 87
Student Details :
USN : 1BM22CS018
Name : ADITYA KUMAR
SGPA: 9.217391304347826

PROGRAM_3:

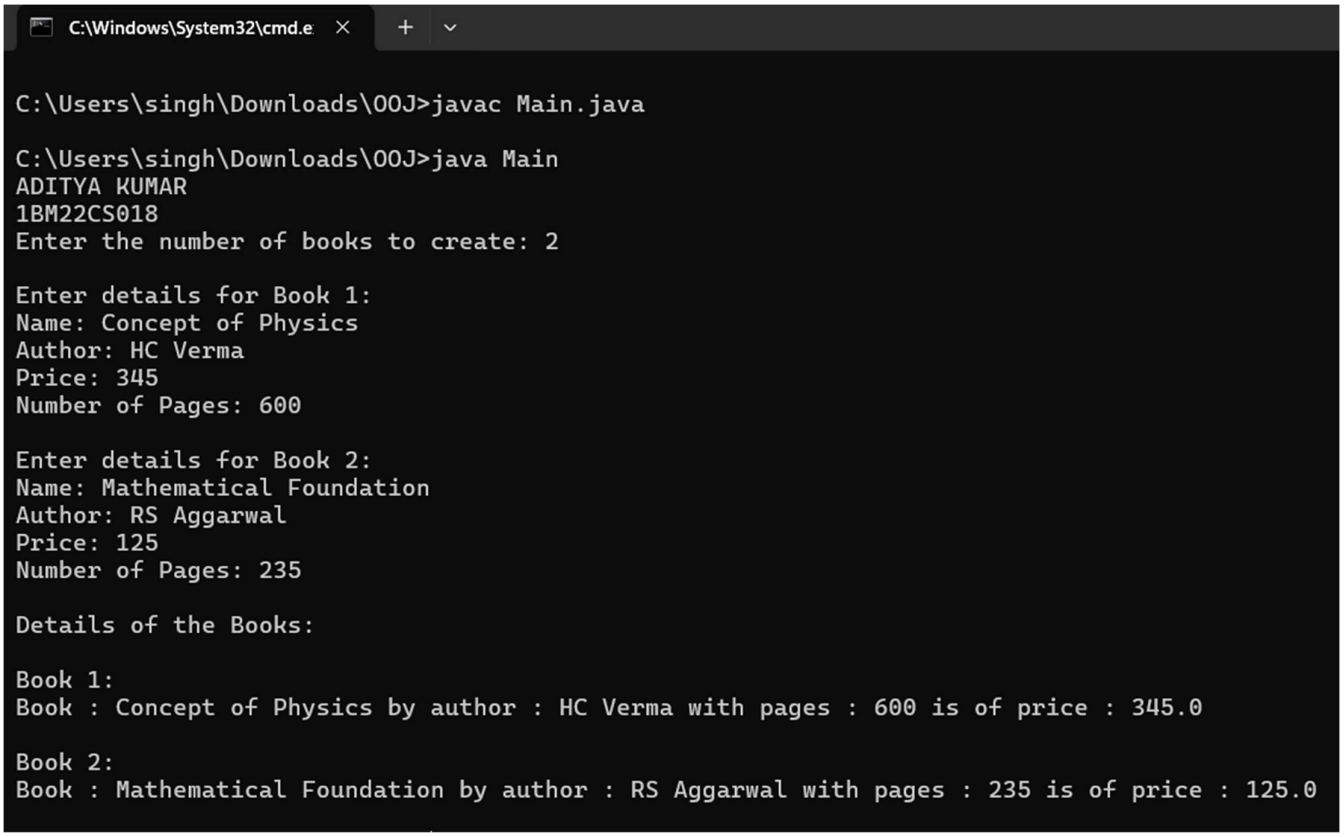
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program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

```
import java.util.Scanner;
class Book{
 String name;
 String author;
 double price;
 int num_pages;
 Book(String name, String author, double price, int num_pages){
  this.name = name;
  this.author = author;
  this.price = price;
  this.num_pages = num_pages;
 }
 void setDetails(String name, String author, double price, int num_pages){
   this.name = name;
  this.author = author;
  this.price = price;
  this.num_pages = num_pages;
 }
 void getDetails(){
   String S = "Book : " + name +" by author : "+ author + " with pages : "+num_pages+" is of price : "+price;
   System.out.println(S);
 }
 public String toString(){
  String S = "Book : " + name +" by author : "+ author + " with pages : "+num_pages+" is of price : "+price;
  return S;
 }
```

```
}




public class Main {
public static void main(String [] args){
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the number of books to create: ");
    int n = scanner.nextInt();
    Book[] books = new Book[n];

    for (int i = 0; i < n; i++) {
      scanner.nextLine();
      System.out.println("\nEnter details for Book " + (i + 1) + ":");
      System.out.print("Name: ");
      String name = scanner.nextLine();
      System.out.print("Author: ");
      String author = scanner.nextLine();
      System.out.print("Price: ");
      double price = scanner.nextDouble();
      System.out.print("Number of Pages: ");
      int numPages = scanner.nextInt();

      books[i] = new Book(name, author, price, numPages);
    }

    System.out.println("\nDetails of the Books:");
    for (int i = 0; i < n; i++) {
      System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toString());
    }

    scanner.close();
```

}

}**OUTPUT:**


C:\Windows\System32\cmd.e
C:\Users\singh\Downloads\OOJ>javac Main.java
C:\Users\singh\Downloads\OOJ>java Main
ADITYA KUMAR
1BM22CS018
Enter the number of books to create: 2
Enter details for Book 1:
Name: Concept of Physics
Author: HC Verma
Price: 345
Number of Pages: 600
Enter details for Book 2:
Name: Mathematical Foundation
Author: RS Aggarwal
Price: 125
Number of Pages: 235
Details of the Books:
Book 1:
Book : Concept of Physics by author : HC Verma with pages : 600 is of price : 345.0
Book 2:
Book : Mathematical Foundation by author : RS Aggarwal with pages : 235 is of price : 125.0

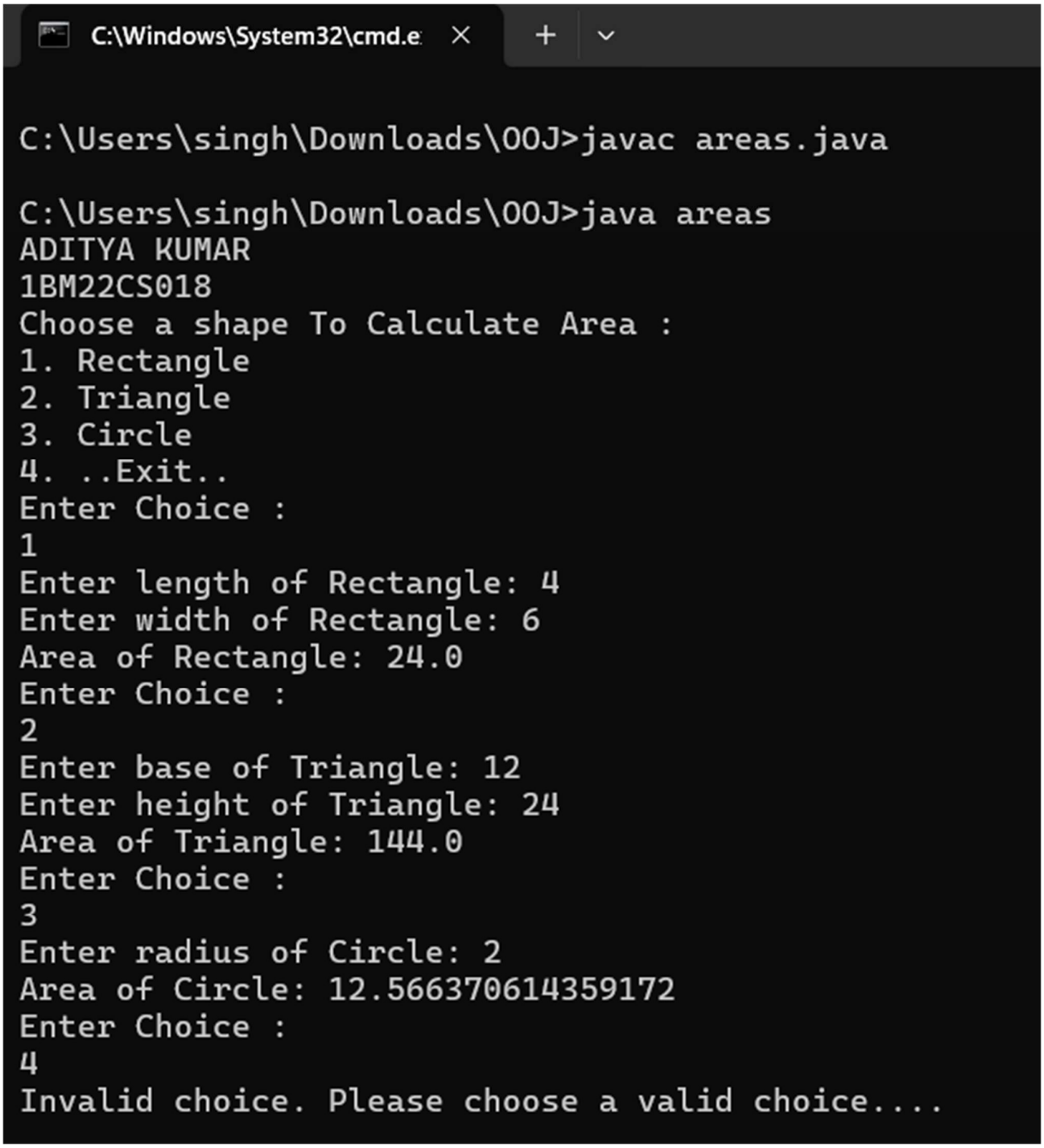
PROGRAM_4:

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. code:

```
import java.util.Scanner;

abstract class Shape {
   double dimension1;
   double dimension2;

   // Abstract method to print the area
   public abstract void printArea();
}

class Rectangle extends Shape {
   // Constructor to initialize dimensions
   public Rectangle(double length, double width) {
      this.dimension1 = length;
      this.dimension2 = width;
   }

   // Implementation of abstract method to print area
   @Override
   public void printArea() {
      double area = dimension1 * dimension2;
      System.out.println("Area of Rectangle: " + area);
   }
}

class Triangle extends Shape {
   // Constructor to initialize dimensions
   public Triangle(double base, double height) {
      this.dimension1 = base;
      this.dimension2 = height;
   }

   // Implementation of abstract method to print area
   @Override
   public void printArea() {
      double area = 0.5 * dimension1 * dimension2;
      System.out.println("Area of Triangle: " + area);
   }
}

class Circle extends Shape {
   // Constructor to initialize dimension
   public Circle(double radius) {
      this.dimension1 = radius;
   }
```

```
    // Implementation of abstract method to print area
    @Override
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class areas {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose a shape To Calculate Area :");
        System.out.println("1. Rectangle");
        System.out.println("2. Triangle");
        System.out.println("3. Circle");
        System.out.println("4. ..Exit..");

        int run = 1;
        while(run==1){
            System.out.println("Enter Choice : ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter length of Rectangle: ");
                    double length = scanner.nextDouble();
                    System.out.print("Enter width of Rectangle: ");
                    double width = scanner.nextDouble();
                    Rectangle rectangle = new Rectangle(length, width);
                    rectangle.printArea();
                    break;

                case 2:
                    System.out.print("Enter base of Triangle: ");
                    double base = scanner.nextDouble();
                    System.out.print("Enter height of Triangle: ");
                    double height = scanner.nextDouble();
                    Triangle triangle = new Triangle(base, height);
                    triangle.printArea();
                    break;

                case 3:
                    System.out.print("Enter radius of Circle: ");
                    double radius = scanner.nextDouble();
                    Circle circle = new Circle(radius);
                    circle.printArea();
                    break;

                case 4:
                    run=0;
                default:
                    System.out.println("Invalid choice. Please choose a valid choice....");
            }
        }


        scanner.close();
    }
```

}**OUTPUT:**



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 Cur-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

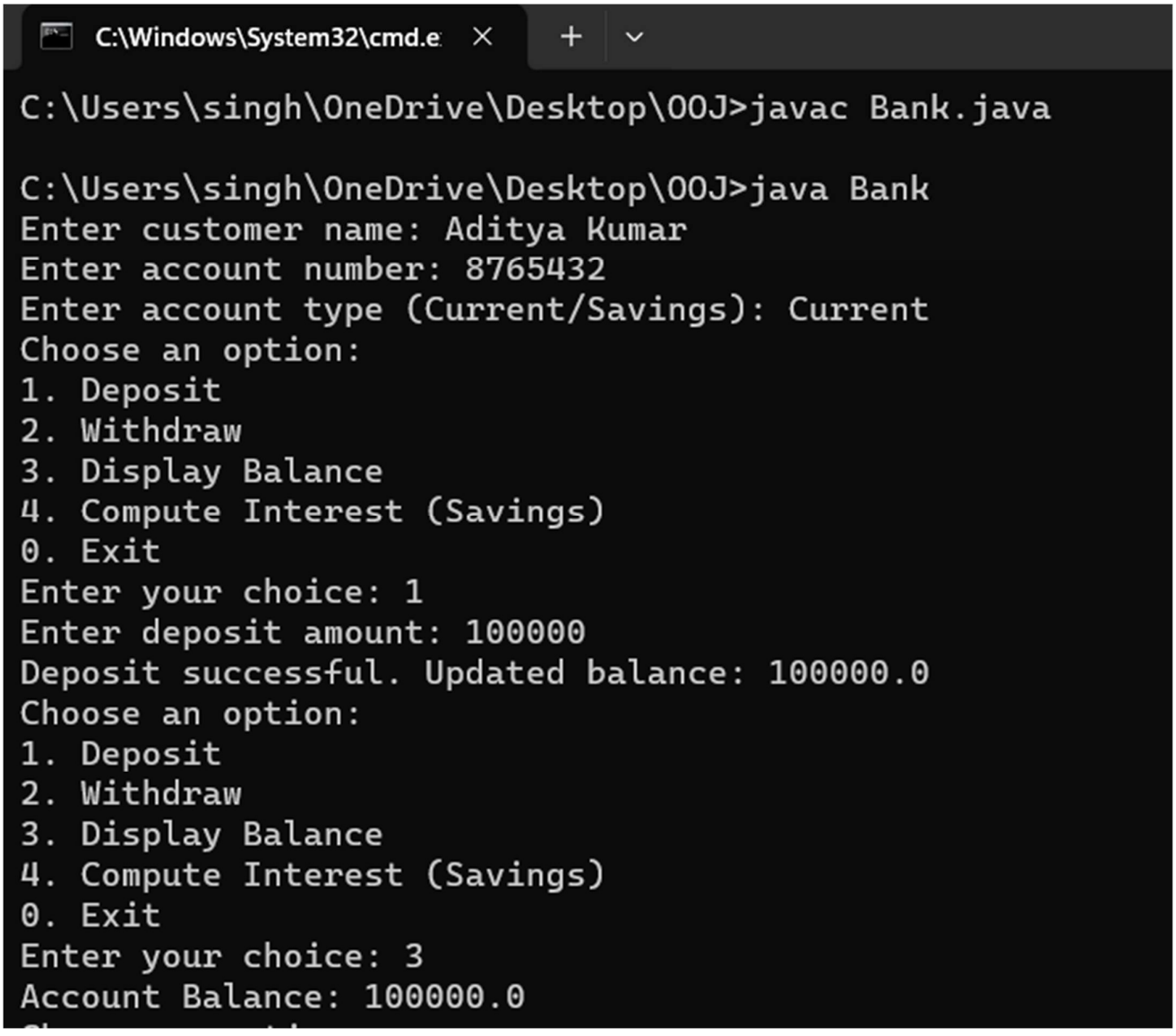
Check for the minimum balance, impose penalty if necessary and update the balance. code:

```
class Account {
   String customerName;
   long accountNumber;
   String accountType;
   double balance;

   // Constructor to initialize account details
   public Account(String customerName, long accountNumber, String accountType, double balance) {
      this.customerName = customerName;
      this.accountNumber = accountNumber;
      this.accountType = accountType;
      this.balance = balance;
   }

   // Method to accept deposit and update the balance
   public void deposit(double amount) {
      balance += amount;
      System.out.println("\nDeposit of Rs." + amount + " successful. =>> Updated balance: Rs." + balance);
   }

   // Method to display the balance
   public void displayBalance() {
      System.out.println("\nAccount Balance for " + accountType + " Account (Account Number: " + accountNumber + ") =>> " +
balance);
   }

   // Method to compute and deposit interest
   public void depositInterest(double rate) {
      if (accountType=="Savings") {
         double interest = balance * (rate / 100);
         System.out.println("\n------Interest Deposit : ------");
         deposit(interest);
      } else {
         System.out.println("\nInterest is not applicable for Current Account.");
      }
   }

   // Method to permit withdrawal and update the balance
   public void withdraw(double amount) {
      if (balance >= amount) {
         balance -= amount;
         System.out.println("\nWithdrawal of Rs." + amount + " successful. =>> Updated balance: Rs." + balance);
      } else {
         System.out.println("\nInsufficient funds. Withdrawal not allowed.");
      }
   }
}
```

```
class CurAcct extends Account {
    private double minBalance;
    private double serviceCharge;

    // Constructor to initialize additional details for current account
    public CurAcct(String customerName, long accountNumber, double balance, double minBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    // Method to check for the minimum balance, impose penalty if necessary, and update the balance
    public void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("\nMinimum balance requirement not met. Service charge of Rs." + serviceCharge + " imposed.");
            System.out.print("=> Updated balance =>> Rs." + balance);
        } else {
            System.out.println("\nMinimum balance requirement is met. No service charge imposed.");
        }
    }

    private int chequeNumber = 1001; // Starting cheque number

    // Method to issue a cheque
    public void issueCheque(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("\nCheque of Rs." + amount + " issued. Updated balance =>> Rs." + balance);
            System.out.println("\nCheque Number: " + chequeNumber++);
        } else {
            System.out.println("\nInsufficient funds. Cheque not issued.");
        }
    }

}

class SavAcct extends Account {
    double interestRate;

    // Constructor to initialize additional details for savings account
    public SavAcct(String customerName, long accountNumber, double balance, double interestRate) {
        super(customerName, accountNumber, "Savings", balance);
        this.interestRate = interestRate;
    }
}

public class main5{
    public static void main(String[] args) {
        // Create a Savings Account
        SavAcct savingsAccount = new SavAcct("John Doe", 123456789, 5000, 5);

        // Create a Current Account
        CurAcct currentAccount = new CurAcct("Jane Smith", 987654321, 7000, 1000, 20);

        // Demonstrate operations on Savings Account
        System.out.println("\n====== Operations on Savings Account: ======\n");
```

```
        savingsAccount.displayBalance();
        savingsAccount.deposit(2000);
        savingsAccount.depositInterest(5);
        savingsAccount.displayBalance();
        savingsAccount.withdraw(1000);
        savingsAccount.displayBalance();

        // Demonstrate operations on Current Account
        System.out.println("\n\n ====== Operations on Current Account: ======\n");
        currentAccount.displayBalance();
        currentAccount.deposit(1500);
        currentAccount.checkMinBalance();
        currentAccount.withdraw(8000); // This withdrawal will incur a service charge
        currentAccount.displayBalance();
        currentAccount.checkMinBalance();

        System.out.println("\n\nCheque Issuance for Current Account:");
        currentAccount.issueCheque(300); // Issue a cheque from Current Account
        currentAccount.issueCheque(500);
        currentAccount.displayBalance();
    }
}
```

OUTPUT:



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. code:

```
package CIE;

public class student {
   public String usn;
   public String name;
   public int sem;
   public student(String usn, String name, int sem) {
      this.usn = usn;
      this.name = name;
      this.sem = sem;
   }
}

package CIE;
public class internals extends student {
   public int[] internalMarks;
   public internals(String usn, String name, int sem, int[]
internalMarks) {
      super(usn, name, sem);
      this.internalMarks = internalMarks;
   }
}

package SEE;

import CIE.student;
public class externals extends student {
   public int[] seeMarks;
   public externals(String usn, String name, int sem, int[]
seeMarks) {
      super(usn, name, sem);
      this.seeMarks = seeMarks;
   }
}

import CIE.internals;
import SEE.externals;
import java.util.Scanner;
public class main1 {
   public static void main(String[] args) {
      System.out.println("ADITYA KUMAR");
      System.out.println("1BM22CS018");
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of students: ");
```

```
        int n = scanner.nextInt();
        internals[] cieStudents = new internals[n];
        externals[] seeStudents = new externals[n];

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

System.out.println("Enter details for CIE of student " + (i
+ 1));
        }

            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int
 sem = scanner.nextInt();
            int[] cieMarks = new int[5];
            System.out.print("Enter CIE marks for 5 courses: ");
            for (int j = 0; j < 5; j++) {
               cieMarks[j] = scanner.nextInt();
            }
            cieStudents[i] = new internals(usn, name, sem, cieMarks);

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for SEE of student " + (i
+ 1));
            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int
 sem = scanner.nextInt();
            int[] seeMarks = new int[5];
            System.out.print("Enter SEE marks for 5 courses: ");
            for (int j = 0; j < 5; j++) {
               seeMarks[j] = scanner.nextInt();
            }

 seeStudents[i] = new externals(usn, name, sem,
seeMarks);
        }


        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of Student " + (i + 1));
            System.out.println("USN: " + cieStudents[i].usn);
            System.out.println("Name: " + cieStudents[i].name);
            System.out.println("Semester: " + cieStudents[i].sem);
            System.out.println("CIE Marks: ");
            for (int j = 0; j < 5; j++) {
               System.out.print(cieStudents[i].internalMarks[j] + " ");
            }
            System.out.println("\nSEE Marks: ");
            for (int j = 0; j < 5; j++) {
               System.out.print(seeStudents[i].seeMarks[j] + " ");
```

```
    }
  }
}
```

OUTPUT:

```
Name:Aditya Kumar
USN:1BM22CS018
Enter n:
1
Enter details 1
Enter USN, Name, Sem:

1BM18
Aditya
3
Enter internal and external of sub 1
32
43
Enter internal and external of sub 2
32
34
Enter internal and external of sub 3
45
45
Enter internal and external of sub 4
43
43
Enter internal and external of sub 5
34
21
Final marks of Aditya
Course 1 = 75
Course 2 = 66
Course 3 = 90
Course 4 = 86
Course 5 = 55
```

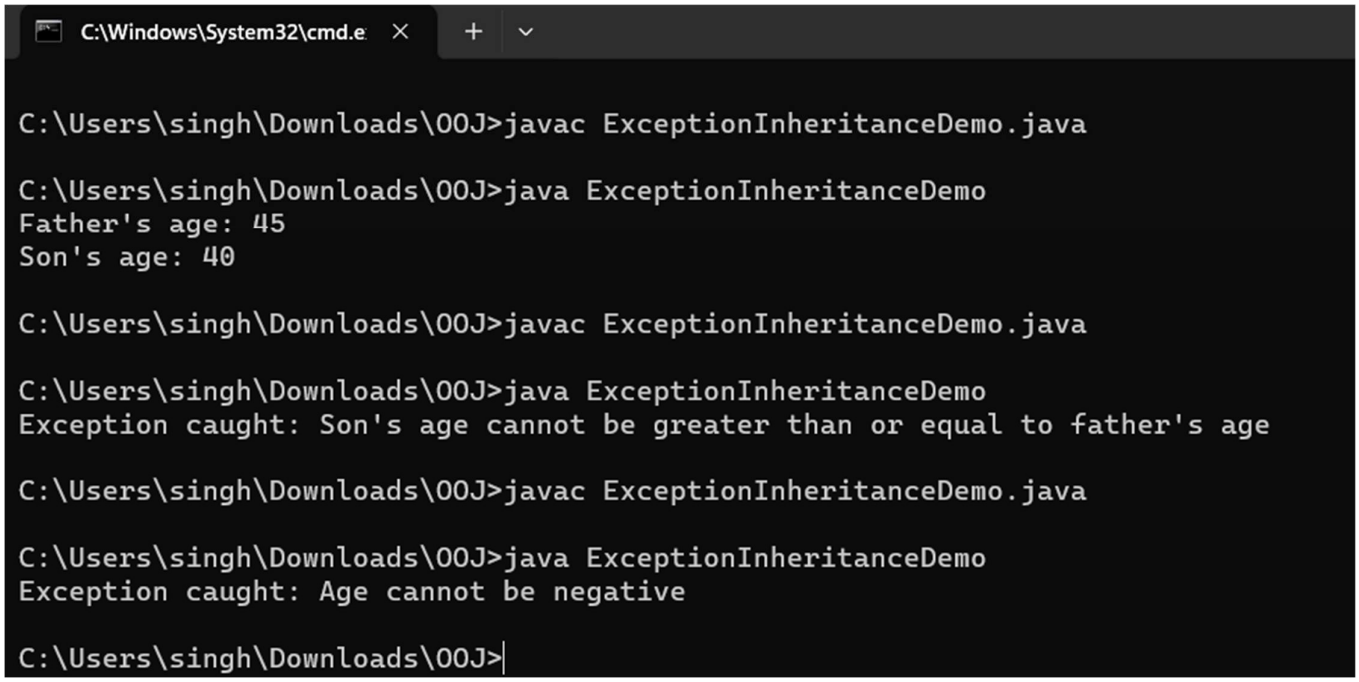
PROGRAM_7:

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. code:

```
class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
```

```
            throw new WrongAge("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 40);
            System.out.println("ADITYA KUMAR");
            System.out.println("1BM22CS018");

            System.out.println("Father's age: " + father.getAge());
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
```

}**OUTPUT:**


C:\Windows\System32\cmd.e
C:\Users\singh\Downloads\OOJ>javac ExceptionInheritanceDemo.java
C:\Users\singh\Downloads\OOJ>java ExceptionInheritanceDemo
Father's age: 45
Son's age: 40
C:\Users\singh\Downloads\OOJ>javac ExceptionInheritanceDemo.java
C:\Users\singh\Downloads\OOJ>java ExceptionInheritanceDemo
Exception caught: Son's age cannot be greater than or equal to father's age
C:\Users\singh\Downloads\OOJ>javac ExceptionInheritanceDemo.java
C:\Users\singh\Downloads\OOJ>java ExceptionInheritanceDemo
Exception caught: Age cannot be negative
C:\Users\singh\Downloads\OOJ>

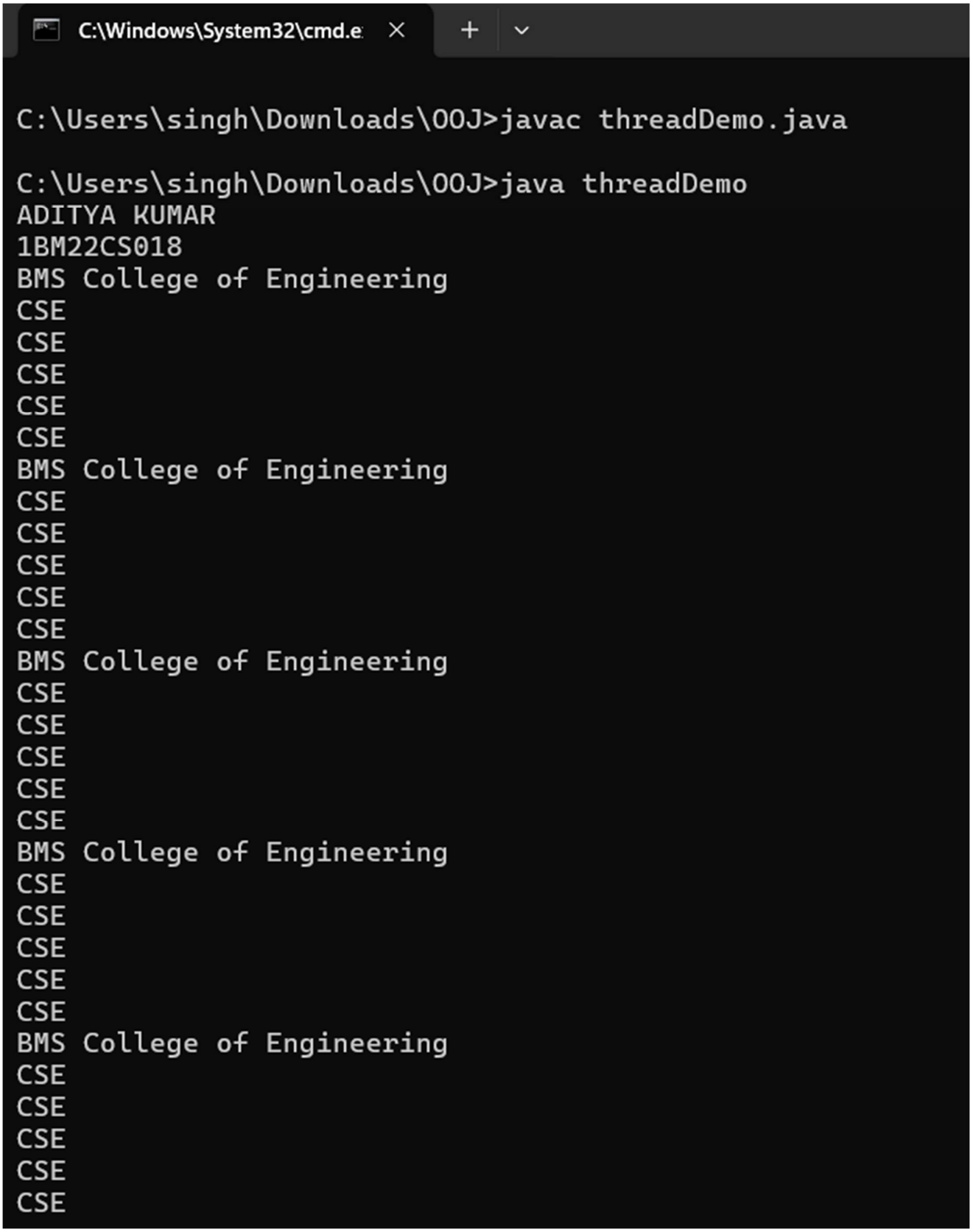
PROGRAM_8:

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. code:

```
class DisplayMessage extends Thread {
   String message;
   int interval;

   public DisplayMessage(String message, int interval) {
      this.message = message;
      this.interval = interval;
   }

   public void run() {
      while (true) {
         try {
            System.out.println(message);
            Thread.sleep(interval * 1000); // Convert seconds to milliseconds
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

public class threadDemo {
   public static void main(String[] args) {
      DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10);
      DisplayMessage thread2 = new DisplayMessage("CSE", 2);uu

      thread1.start();
      thread2.start();
```

}

}**OUTPUT:**


C:\Windows\System32\cmd.e
C:\Users\singh\Downloads\OOJ>javac threadDemo.java
C:\Users\singh\Downloads\OOJ>java threadDemo
ADITYA KUMAR
1BM22CS018
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

PROGRAM_9:

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

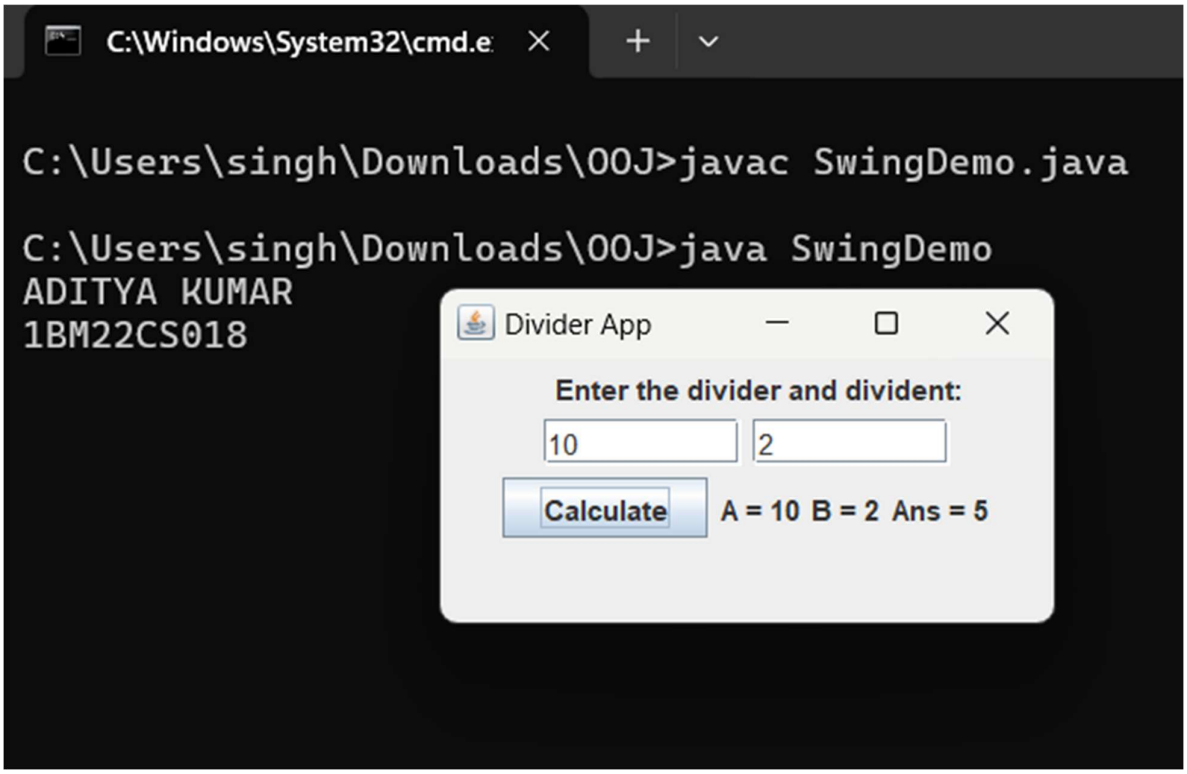
Zero, the program would throw an Arithmetic Exception Display the exception in a message

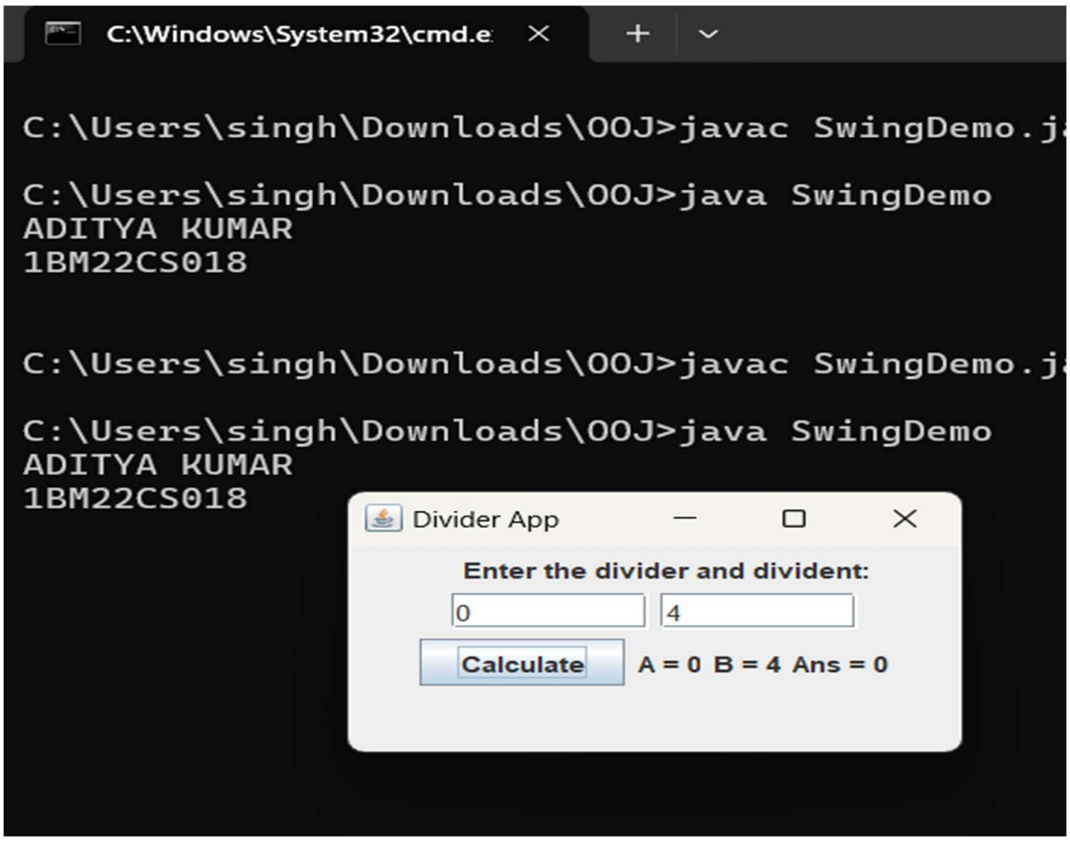
dialog box. code:

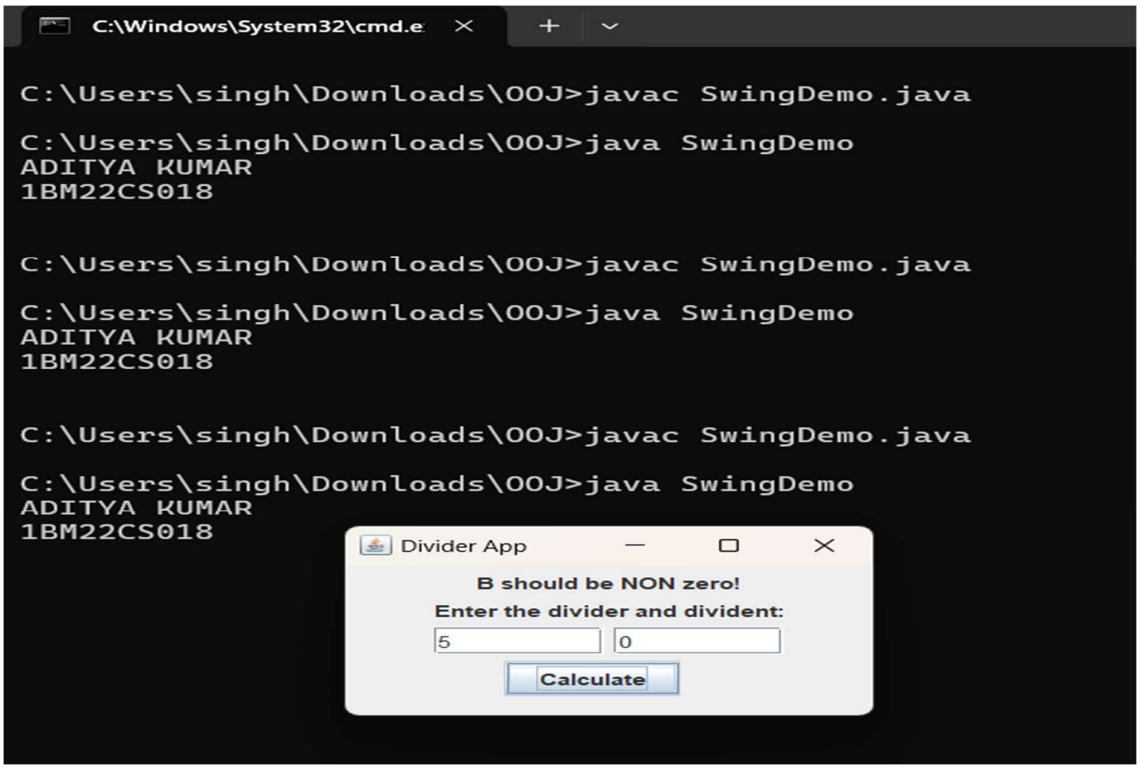
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo{
   SwingDemo(){
      // create jframe container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      // to terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // text label
      JLabel jlab = new JLabel("Enter the divider and divident:");

      // add text field for both numbers
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      // calc button
      JButton button = new JButton("Calculate");

      // labels
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
 JLabel anslab = new JLabel();

      // add in order :)
      jfrm.add(err);  // to display error bois
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      ActionListener l = new ActionListener() {
         public void actionPerformed(ActionEvent evt) {
            System.out.println("Action event from a text field");
         }
```

```
    };
    ajtf.addActionListener(l);
    bjtf.addActionListener(l);

    button.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        try{
          int a = Integer.parseInt(ajtf.getText());
          int b = Integer.parseInt(bjtf.getText());
          int ans = a/b;

          alab.setText("\nA = " + a);
          blab.setText("\nB = " + b);
          anslab.setText("\nAns = "+ ans);
        }
        catch(NumberFormatException e){
          alab.setText("");
          blab.setText("");
          anslab.setText("");
err.setText("Enter Only Integers!");
        }
        catch(ArithmeticException e){
          alab.setText("");
          blab.setText("");
          anslab.setText("");
          err.setText("B should be NON zero!");
        }
      }
    });

    // display frame
    jfrm.setVisible(true);
  }

  public static void main(String args[]){
    // create frame on event dispatching thread
System.out.println("ADITYA KUMAR");
System.out.println("1BM22CS018\n");
    SwingUtilities.invokeLater(new Runnable(){
      public void run(){
        new SwingDemo();
      }
    });
  }
}
```

OUTPUT:


C:\Windows\System32\cmd.e
C:\Users\singh\Downloads\OOJ>javac SwingDemo.java
C:\Users\singh\Downloads\OOJ>java SwingDemo
ADITYA KUMAR
1BM22CS018
Divider App
Enter the divider and divident:
10
2
Calculate
A = 10 B = 2 Ans = 5


C:\Windows\System32\cmd.e
C:\Users\singh\Downloads\OOJ>javac SwingDemo.j
C:\Users\singh\Downloads\OOJ>java SwingDemo
ADITYA KUMAR
1BM22CS018
C:\Users\singh\Downloads\OOJ>javac SwingDemo.j
C:\Users\singh\Downloads\OOJ>java SwingDemo
ADITYA KUMAR
1BM22CS018
Divider App
Enter the divider and divident:
0
4
Calculate
A = 0 B = 4 Ans = 0


C:\Windows\System32\cmd.e
C:\Users\singh\Downloads\OOJ>javac SwingDemo.java
C:\Users\singh\Downloads\OOJ>java SwingDemo
ADITYA KUMAR
1BM22CS018
C:\Users\singh\Downloads\OOJ>javac SwingDemo.java
C:\Users\singh\Downloads\OOJ>java SwingDemo
ADITYA KUMAR
1BM22CS018
C:\Users\singh\Downloads\OOJ>javac SwingDemo.java
C:\Users\singh\Downloads\OOJ>java SwingDemo
ADITYA KUMAR
1BM22CS018
Divider App
B should be NON zero!
Enter the divider and divident:
5
0
Calculate