# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**LAB REPORT**
**on**

# Object Oriented Java Programming

# (23CS3PCOOJ)

***Submitted by***

**Trishul (1WA23CS023)**

***in partial fulfillment for the award of the degree of***
**BACHELOR OF ENGINEERING**
***in***
**COMPUTER SCIENCE AND ENGINEERING**

**B.M.S. COLLEGE OF ENGINEERING**
**(Autonomous Institution under VTU)**

**BENGALURU-560019**
**Sep-2024 to Jan-2025**

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

**Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
**Department of Computer Science and Engineering**

## CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Trishul (1WA23CS023),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

| | |
|---|---|
| Lab faculty Syed Akram<br>Assistant Professor<br>Department of CSE, BMSCE | Dr. Kavitha Sooda<br>Professor & HOD<br>Department of CSE, BMSCE |

# 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;
public class Quad{
   public static void main(String[] args){
      Scanner s=new Scanner(System.in);
      System.out.println("Enter the coefficients:");
      int a=s.nextInt();
      int b=s.nextInt();
      int c=s.nextInt();
      int d=b*b-4*a*c;
      if(d==0){
         System.out.println("Roots are equal");
         System.out.println("Roots are:");
         System.out.println(-b/2*a);
}
      else if(d>0){
         System.out.println("Roots are unique");
         System.out.println((-b+Math.sqrt(d))/(2*a));
         System.out.println((-b-Math.sqrt(d))/(2*a));

}
      else{
         System.out.println("No real roots");
}
   }
}
```

output:

```
C:\Users\trish\OneDrive\Desktop\java>javac Quad.java

C:\Users\trish\OneDrive\Desktop\java>java Quad
Enter the coefficients:
1
-2
1
Roots are equal
Roots are:
1
```

**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.

```
import java.util.Scanner;

public class Student {
    String name;
    String Usn;
    int n;
    int [] marks;
    int [] credits;

    Student(){
        this.name = "";
        this.Usn = "";
    }

    void display(){
        System.out.println("Name : " +name);
        System.out.println("USN : "+ Usn);
        System.out.println("Marks: ");
        for (int i=0; i< marks.length; ++i) System.out.println(marks[i]);
        for (int i=0; i< credits.length; ++i) System.out.println(credits[i]);
        System.out.println("SGPA : "+SGPA());
    }

    double SGPA() {
        int totalMarks = 0, totalCredits = 0;
        for (int i = 0; i < n; ++i) {
            totalMarks += marks[i] * credits[i];
            totalCredits += credits[i];
        }
        return (double) totalMarks / (totalCredits*10);
    }

    void accept(){
        System.out.println("Enter name");
        Scanner cin = new Scanner(System.in);
        name = cin.nextLine();
        System.out.println("Enter USN");
        Usn = cin.nextLine();
        System.out.println("Enter number of subjects");
        int n = cin.nextInt();
                marks = new int[n];
                credits = new int[n];
```

```
        System.out.println("Enter their marks");
        for (int i=0;i<n;++i) marks[i] = cin.nextInt();
        System.out.println("enter their credits");
        for (int i=0; i< n;++i) credits[i] = cin.nextInt();
    }

    public static void main(String[] args){
        Student stud1 = new Student();
            Student stud2 = new Student();
        stud1.accept();
        stud1.display();
            stud2.accept();
        stud2.display();
    }
}
```

output:

```
C:\Users\trish\OneDrive\Desktop\java>javac Student.java

C:\Users\trish\OneDrive\Desktop\java>java Student
Enter name
trishul
Enter USN
1wa23cs023
Enter number of subjects
4
Enter their marks
78
98
87
67
enter their credits
2
2
4
4
Name : trishul
USN : 1wa23cs023
Marks:
78
98
87
67
2
2
4
4
```

**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.*;

        public class Book{
        String name;
        String author;
        double price;
        int pages;


        Book(String name,String author,double price,int pages){
                this.name = name;
                this.author = author;
                this.price = price;
                this.pages= pages;
        }


        public String getname(){
                return name;
        }

        public void setname(){
                this.name = name;
        }

        public String getauthor(){
                return author;
        }

        public void setauthor(){
                this.author = author;
        }

        public double getprice(){
                return price;
        }

        public void setprice(){
```

```
		this.price = price;
	}

	public int getpages(){
		return pages;
	}

	public void setpages(){
		this.pages = pages;
	}


	@Override
	public String toString(){
		return "Name: " + name  + " Author: " + author + " Price: " + price + " No of pages: "+
pages ;
	}

	public static void main(String [] args){
		Scanner cin = new Scanner(System.in);
		System.out.println("Enter the number of books");
		int n = cin.nextInt();
		cin.nextLine();

		Book[] book = new Book[n];
		System.out.println("Enter the values of each book");
		System.out.println("\n");
		for (int i=0;i<n;i++)
		{
			System.out.println("Enter the name of the book");
			String name=cin.nextLine();
			System.out.println("Enter the name of the author");
			String author=cin.nextLine();
			System.out.println("Enter the price");
			double price=cin.nextDouble();
			System.out.println("Enter the no of pages");
			int pages=cin.nextInt();
			book[i]=new Book(name,author,price,pages);
			cin.nextLine();
		}


		System.out.println("Details of each book are");
		for (int i=0; i<n;i++){
			System.out.println("Book "+(i+1)+book[i].toString());

		}
	}
}
```

output:

```
C:\Users\trish\OneDrive\Desktop\java>javac Book.java

C:\Users\trish\OneDrive\Desktop\java>java Book
Enter the number of books
2
Enter the values of each book


Enter the name of the book
harry potter
Enter the name of the author
james
Enter the price
2345
Enter the no of pages
690
Enter the name of the book
dragonlair
Enter the name of the author
bond
Enter the price
564
Enter the no of pages
567
Details of each book are
Book 1Name: harry potter Author: james Price: 2345.0 No of pages: 690
Book 2Name: dragonlair Author: bond Price: 564.0 No of pages: 567
```

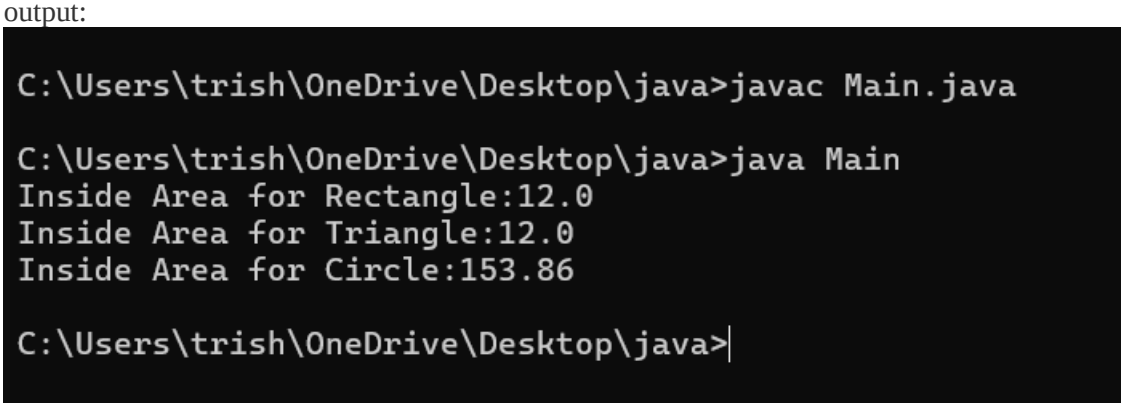
**program 4**

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.

```
abstract class Shape {

    int a, b, c;

    Shape(int a, int b) {
        this.a = a;
        this.b = b;
    }

    Shape(int c) {
        this.c = c;
    }

    abstract void printArea();
}

class Rectangle extends Shape {

    Rectangle(int a, int b) {
        super(a, b);
    }
    double i;

    void printArea() {
        i = a * b;
        System.out.println("Inside Area for Rectangle:" + i);
    }
}

class Triangle extends Shape {

    Triangle(int a, int b) {
        super(a, b);
    }
    double j;

    void printArea() {
        j = (a * b) / 2;
        System.out.println(
                "Inside Area for Triangle:" + j);
    }
}

class Circle extends Shape {
```

```
    Circle(int c) {
        super(c);
    }
    double k;

    void printArea() {
        k = (3.14 * c * c);
        System.out.println("Inside Area for Circle:" + k);
    }
}

class Main {

    public static void main(String args[]) {
        Rectangle r = new Rectangle(3, 4);
        Triangle t = new Triangle(4, 6);
        Circle c = new Circle(7);
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

output:



**program5**

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.

```
import java.io.*;
import java.util.*;

abstract class Account {

    String cusName, accNo;
    double balance;

    Account(String name, String num) {
        this.cusName = name;
        this.accNo = num;
        this.balance = 0.0;
    }

    abstract void deposit(double num);

    abstract void withdraw(double num);

    abstract void display();
}

class SavAcc extends Account {

    double rate;

    SavAcc(String cusName, String accNo, double rate) {
        super(cusName, accNo);
        this.rate = rate;
    }
```

```
    @Override
    void deposit(double num) {
        balance += num;
        System.out.println("The current balance after deposit is:" + balance);
    }

    void withdraw(double num) {
        if (num > balance) {
            System.out.println("Not enough balance");
        } else {
            balance -= num;
            System.out.println("Withdrew amount=" + num);
        }
    }

    void calcInterest() {
        double interest = balance * (rate / 100);
        balance += interest;
        System.out.println("The interest deposited is:" + interest);
    }

    void display() {
        System.out.println("Savings account balance:" + balance);
    }
}

class CurAcc extends Account {

    double minBal, serviceCharge;

    CurAcc(String cusName, String accNo, double minBal, double serviceCharge) {
        super(cusName, accNo);
        this.minBal = minBal;
        this.serviceCharge = serviceCharge;
    }

    @Override
    void deposit(double num) {
        balance += num;
        System.out.println("The current balance after deposit is:" + balance);
    }

    void withdraw(double num) {
        if (num > balance) {
            System.out.println("Not enough balance");
        } else {
            balance -= num;
            System.out.println("Withdrew amount=" + num);
            checkMinBalance();
        }
    }
```

```
    void checkMinBalance() {
        if (balance < minBal) {
            System.out.println("Lower than min balance.");
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge + " applied. NewBalance: " + balance);
        }
    }

    void display() {
         System.out.println("Savings account balance:" + balance);
    }
}

class Bank {

    public static void main(String args[]) {
        int choice, n;
        String name, accno;
        double rate = 5.0, minbal = 500.0, sercha = 25.0;
        double amount;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter name:");
        name = sc.nextLine();
        System.out.println("Enter account no.:");
        accno = sc.nextLine();
        System.out.println("Enter[1.Savings,2.Current]");
        n = sc.nextInt();
        if (n == 1) {
            SavAcc a = new SavAcc(name, accno, rate);
            System.out.println("Savings Account created");
            do {
                System.out.println("Enter[1.Deposit,2.Withdraw,3.GetBalance,4.ADD Interest,5.Exit]");
                choice = sc.nextInt();
                switch (choice) {
                    case 1:
                        System.out.println("Enter the amount to be inserted:");
                        amount = sc.nextInt();
                        a.deposit(amount);
                        break;
                    case 2:
                        System.out.println("Enter the amount to be withdrawn:");
                        amount = sc.nextInt();
                        a.withdraw(amount);
                        break;
                    case 3:
                        a.display();
                        break;
                    case 4:
                        a.calcInterest();
                        break;
                    case 5:
```

```
                System.out.println("Exiting");
                break;
            default:
                System.out.println("Enter proper choice.");
                break;
        }
      } while (choice != 5);
    } else if (n == 2) {
      CurAcc b = new CurAcc(name, accno, minbal, sercha);
      System.out.println("Current account created");
      do {
        System.out.println("Enter[1.Deposit,2.Withdraw,3.GetBalance,4.Exit]");
        choice = sc.nextInt();
        switch (choice) {
          case 1:
            System.out.println("Enter the amount to be inserted:");
            amount = sc.nextInt();
            b.deposit(amount);
            break;
          case 2:
            System.out.println("Enter the amount to be withdrawn:");
            amount = sc.nextInt();
            b.withdraw(amount);
            break;
          case 3:
            b.display();
            break;
          case 4:
            System.out.println("Exiting");
            break;
          default:
            System.out.println("Enter proper choice.");
            break;
        }
      } while (choice != 4);
    } else {
      System.out.println("Invalid account type");
      return;
    }
  }
}
```

output

```
C:\Users\trish\OneDrive\Desktop\java>java Bank
Enter name:
trishul
Enter account no.:
123
Enter[1.Savings,2.Current]
1
Savings Account created
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.ADD Interest,5.Exit]
1
Enter the amount to be inserted:
3454
The current balance after deposit is:3454.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.ADD Interest,5.Exit]
3
Savings account balance:3454.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.ADD Interest,5.Exit]
2
Enter the amount to be withdrawn:
34
Withdrew amount=34.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.ADD Interest,5.Exit]
4
The interest deposited is:171.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.ADD Interest,5.Exit]
5
Exiting
```

```
C:\Users\trish\OneDrive\Desktop\java>javac Bank.java

C:\Users\trish\OneDrive\Desktop\java>java Bank
Enter name:
vaishal
Enter account no.:
345
Enter[1.Savings,2.Current]
2
Current account created
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.Exit]
1
Enter the amount to be inserted:
345
The current balance after deposit is:345.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.Exit]
3
Savings account balance:345.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.Exit]
2
Enter the amount to be withdrawn:
43
Withdrew amount=43.0
Lower than min balance.
Service charge of 25.0 applied. NewBalance: 277.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.Exit]
3
Savings account balance:277.0
Enter[1.Deposit,2.Withdraw,3.GetBalance,4.Exit]
4
Exiting
```

**program6**

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.

```
package cie;

public class Internals {

    public int[] internalMarks = new int[5];

    public Internals(int[] marks) {
        for (int i = 0; i < 5; i++) {
            internalMarks[i] = marks[i];
        }
    }
}

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 see;

import cie.Student;

public class External extends Student {

    public int[] seeMarks = new int[5];

    public External(String usn, String name, int sem, int seeMarks[]) {
        super(usn, name, sem);
        for (int i = 0; i < 5; i++) {
```

```
            this.seeMarks[i] = seeMarks[i];
        }
    }
}


import cie.Internals;
import see.External;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for the " + (i + 1) + " Student " + ":");
            sc.nextLine();
            System.out.print("USN: ");
            String usn = sc.nextLine();
            System.out.print("Name: ");
            String name = sc.nextLine();
            System.out.print("Semester: ");
            int sem = sc.nextInt();
            System.out.println("Enter internal marks for 5 courses (out of 50): ");
            int internalMarks[] = new int[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + " internal marks: ");
                internalMarks[j] = sc.nextInt();
            }
            Internals internal = new Internals(internalMarks);
            System.out.println("Enter SEE marks for 5 courses (out of 100): ");
            int seeMarks[] = new int[5];

            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + " SEE marks: ");
                seeMarks[j] = sc.nextInt();
            }
            External external = new External(usn, name, sem, seeMarks);
            System.out.println("\nFinal Marks for " + external.name + " :" + external.usn + ":for:");
            System.out.println("Semester: " + external.sem);
            System.out.println("Total Marks:");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course " + (j + 1) + ": " + (double) (internal.internalMarks[j]
                        + (external.seeMarks[j] / 2)));
            }
        }
    }
}
```

output:

```
C:\Users\trish\OneDrive\Desktop\java\packages>javac CIE/*.java

C:\Users\trish\OneDrive\Desktop\java\packages>javac CIE/*.java

C:\Users\trish\OneDrive\Desktop\java\packages>javac Main.java

C:\Users\trish\OneDrive\Desktop\java\packages>java Main
Enter the number of students: 1

Enter details for the 1 Student :
USN: 1wa
Name: trishul
Semester: 2
Enter internal marks for 5 courses (out of 50):
Course 1 internal marks: 45
Course 2 internal marks: 50
Course 3 internal marks: 46
Course 4 internal marks: 49
Course 5 internal marks: 48
Enter SEE marks for 5 courses (out of 100):
Course 1 SEE marks: 90
Course 2 SEE marks: 89
Course 3 SEE marks: 99
Course 4 SEE marks: 88
Course 5 SEE marks: 99

Final Marks for trishul :1wa:for:
Semester: 2
Total Marks:
Course 1: 90.0
Course 2: 94.0
Course 3: 95.0
Course 4: 93.0
Course 5: 97.0
```

**program7**

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 Wrong Age( ) when the input age=father’s age.

```
import java.io.*;

class WrongAge extends Exception {

    public WrongAge(String txt) {
        super(txt);
    }
}

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 < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age must be less than father's age.");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
```

```
        return sonAge;
    }
}

class Main {

    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(40, 20);
            System.out.println("Father's Age: " + father.getAge());
            System.out.println("Son's Age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println(e);
        }
        try {
            Son son2 = new Son(-5, 10);
        } catch (WrongAge e) {
            System.out.println(e);
        }
        try {
            Son son3 = new Son(30, 30);
        } catch (WrongAge e) {
            System.out.println(e);
        }
        try {
            Son son4 = new Son(30, -5);
        } catch (WrongAge e) {
            System.out.println(e);
        }
    }
}
```

output:

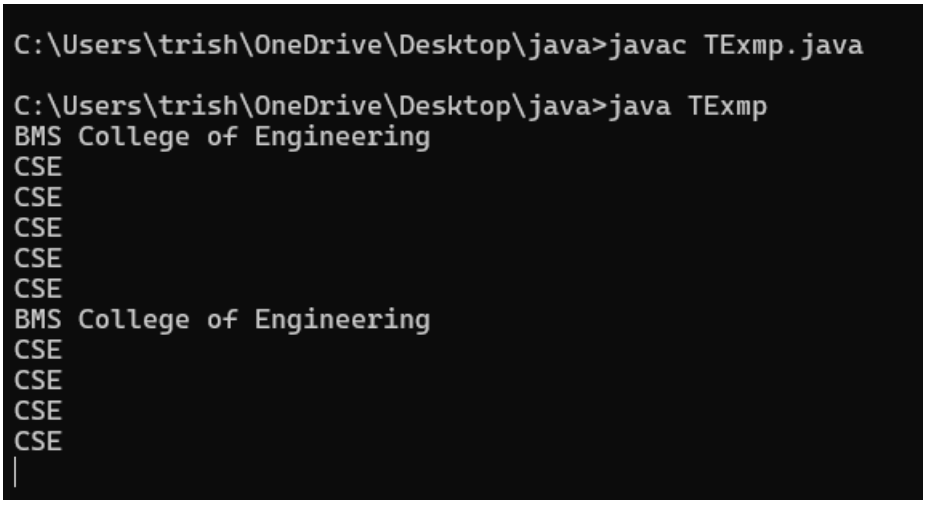
```
C:\Users\trish\OneDrive\Desktop\java>javac Main.java

C:\Users\trish\OneDrive\Desktop\java>java Main
Father's Age: 40
Son's Age: 20
WrongAge: Age cannot be negative.
WrongAge: Son's age must be less than father's age.
WrongAge: Son's age cannot be negative.

C:\Users\trish\OneDrive\Desktop\java>
```

**program8**

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 CollegeThread extends Thread {

    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("CollegeThread interrupted.");
        }
    }
}

class CSEThread extends Thread {

    public void run() {
        try {
            for (int i = 0; i < 25; i++) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted.");
        }
    }
}

public class TExmp {

    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();
        collegeThread.start();
        cseThread.start();
    }
}
```

output:


C:\Users\trish\OneDrive\Desktop\java>javac TExmp.java
C:\Users\trish\OneDrive\Desktop\java>java TExmp
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE

**program9**

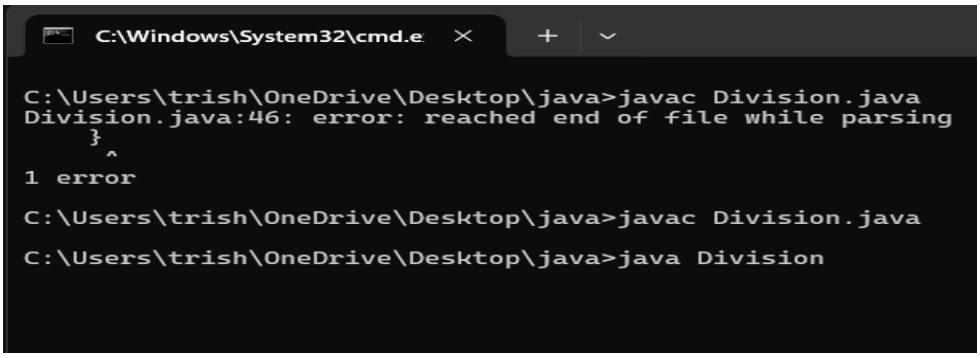
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number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

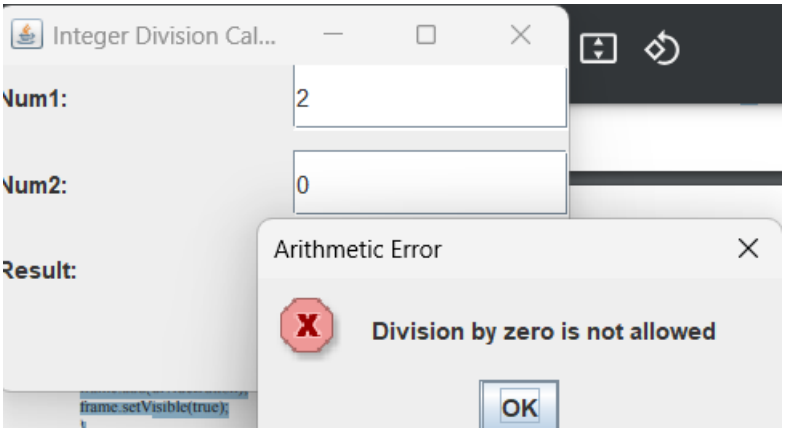
code:

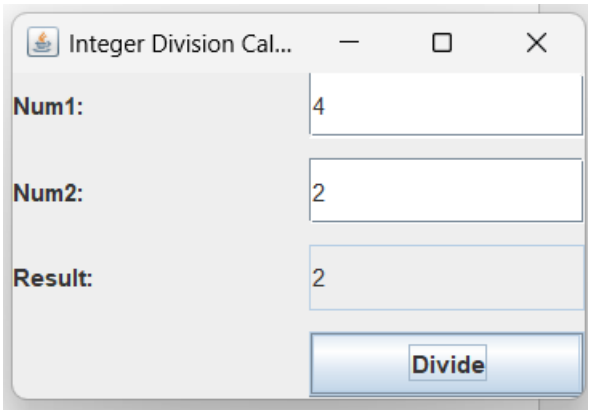
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

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

      // Create components
      JLabel jlab = new JLabel("Enter the divider and dividend:");
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);
      JButton button = new JButton("Calculate");
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();

      // Add components in order
      jfrm.add(err); // To display errors
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      // Add ActionListeners
      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("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText(""); // Clear error message
          } 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 the frame
      jfrm.setVisible(true);
    }

    public static void main(String args[]) {
      // Create frame on Event Dispatching Thread
      SwingUtilities.invokeLater(new Runnable() {
        public void run() {
          new SwingDemo();
        }
      });
    }
}
```

output:


C:\Windows\System32\cmd.e
C:\Users\trish\OneDrive\Desktop\java>javac Division.java
Division.java:46: error: reached end of file while parsing
}
^
1 error
C:\Users\trish\OneDrive\Desktop\java>javac Division.java
C:\Users\trish\OneDrive\Desktop\java>java Division


Integer Division Cal...
Num1:
2
Num2:
0
Result:
Arithmetic Error
Division by zero is not allowed
OK
frame.setVisible(true);


Integer Division Cal...
Num1:
4
Num2:
2
Result:
2
Divide

**program10**

Interprocess Communication and Deadlock

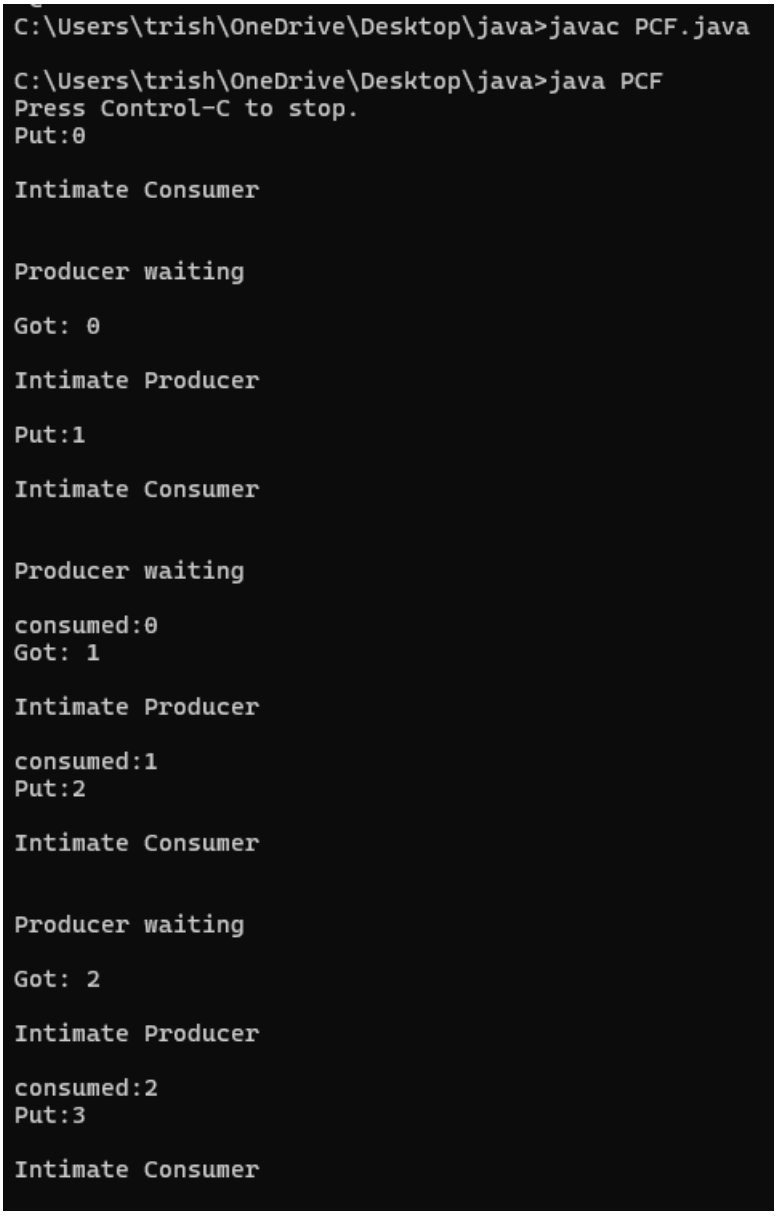
```
class Q {

    int n;
    boolean valueSet = false;

    synchronized int get() {
        while (!valueSet)
try {
            System.out.println("\nConsumer waiting\n");
            wait();
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("\nIntimate Producer\n");
        notify();
        return n;
    }
```

```
    synchronized void put(int n) {
        while (valueSet)
try {
            System.out.println("\nProducer waiting\n");
            wait();
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        this.n = n;
        valueSet = true;
        System.out.println("Put:" + n);
        System.out.println("\nIntimate Consumer\n");
        notify();
    }
}

class Producer implements Runnable {

    Q q;

    Producer(Q q) {
        this.q
                = q;
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i
                = 0;
        while (i < 15) {
            q.put(i++);
```

```
        }
    }
}

class Consumer implements Runnable {

    Q q;

    Consumer(Q q) {
        this.q
                = q;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("consumed:" + r);
            i++;

        }
    }
}

class PCF {

    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
```

```
        System.out.println("Press Control-C to stop.");
    }
}
```

output:




//deadlock

```
class AA {

   synchronized void foo(BB b) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered A.foo");
      try {

         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("A Interrupted");
      }
      System.out.println(name
            + " trying to call B.last()");
      b.last();
   }

   synchronized void last() {
      System.out.println("Inside A.last");
   }
}

class BB {

   synchronized void bar(AA a) {
      String name
            = Thread.currentThread().getName();
      System.out.println(name
            + " entered BB.bar");
      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("B Interrupted");
      }
      System.out.println(name
            + " trying to call A.last()");
      a.last();
   }

   synchronized void last() {
      System.out.println("Inside A.last");
   }
}

class Deadlock implements Runnable {

   AA a = new AA();
   BB b = new BB();

   Deadlock() {
```

```
        Thread.currentThread().setName("MainThread");
        Thread t
                = new Thread(this, "RacingThread");
        t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a); // get lock on b in other thread
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
    }
}
```

Output: