VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Rishi Kumar Chourasia (1BF24CS253)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Aug-2025 to Jan-2026

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 **Rishi Kumar Chourasia (1BF24CS253),** who is a bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment 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.

Dr. Seema Patil Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Github Link: https://github.com/Rishi-KC-BMS/23CS3PCOOJ

Program 1

Implement Quadratic Equation

CODE

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

      int a,b,c,d;
      double r1,r2;
      System.out.println("\nRishi Kumar Chourasia - 1BF24CS253\n");
      Scanner in = new Scanner(System.in);

      System.out.print("Enter value of coefficient a - ");
      a = in.nextInt();
      System.out.println();

      System.out.print("Enter value of coefficient b - ");
      b = in.nextInt();
      System.out.println();

      System.out.print("Enter value of coefficient c - ");
      c = in.nextInt();
      System.out.println();

      if(a == 0)
      {
         System.out.println("Not a Quadratic Equation");
      }

      else
      {
         d= b*b - 4*a*c;

         if(d==0)
         {
            r1 = (-b)/(2*a);
            System.out.println("Roots are real and equal -> "+r1);
         }

         else if(d>0)
         {
            r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
```

```
            System.out.println("Roots are real and distinct-> "+r1+" and "+r2);
        }

        else
        {
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(2*a);
        System.out.println("Roots are imaginary -> "+r1+" and "+r2);
        }

        }
    }
}
```

Terminal Output :

```
Rishi Kumar Chourasia - 1BF24CS253
Enter value of coefficient a - 0

Enter value of coefficient b - 1

Enter value of coefficient c - 1

Not a Quadratic Equation
PS C:\Users\Admin\Desktop\1BF24CS253> cd "c:\Users\Admin\Desktop\1BF24CS253\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }

Rishi Kumar Chourasia - 1BF24CS253
Enter value of coefficient a - 1

Enter value of coefficient b - -2

Enter value of coefficient c - 1

Roots are real and equal -> 1.0
PS C:\Users\Admin\Desktop\1BF24CS253> cd "c:\Users\Admin\Desktop\1BF24CS253\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }

Rishi Kumar Chourasia - 1BF24CS253
Enter value of coefficient a - 2

Enter value of coefficient b - 5

Enter value of coefficient c - -3

Roots are real and distinct-> 0.5 and -3.0
PS C:\Users\Admin\Desktop\1BF24CS253> cd "c:\Users\Admin\Desktop\1BF24CS253\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }

Rishi Kumar Chourasia - 1BF24CS253

Enter value of coefficient a - 1

Enter value of coefficient b - 2

Enter value of coefficient c - 5

Roots are imaginary -> -1.0 and 2.0
PS C:\Users\Admin\Desktop\1BF24CS253>
```

Program 2

Implement SGPA Calculator

CODE

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;

    public Subject() {
        this.subjectMarks = 0;
        this.credits = 0;
        this.grade = 0;
    }
}

class Student {
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject[] subject;

    public Student() {
        s = new Scanner(System.in);
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
    }

    public void getStudentDetails() {
        System.out.print("Enter student name: ");
        name = s.nextLine();
        System.out.print("Enter student USN: ");
        usn = s.nextLine();
    }

    public void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            subject[i].subjectMarks = s.nextInt();
```

```
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            subject[i].credits = s.nextInt();

            subject[i].grade = (subject[i].subjectMarks / 10) + 1;

            if (subject[i].grade == 11) {
                subject[i].grade = 10;
            }
            if (subject[i].grade <= 4) {
                subject[i].grade = 0;
            }
        }
    }
    public void computeSGPA() {
        int effectiveScore = 0;
        int totalCredits = 0;

        for (int i = 0; i < 8; i++) {
            effectiveScore += (subject[i].grade * subject[i].credits);
            totalCredits += subject[i].credits;
        }

        SGPA = (double) effectiveScore / totalCredits;
    }


    public void display() {
        System.out.println("\nStudent Details:");
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }
}

public class Main {
     public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Student[] students = new Student[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1) + ":");
            students[i] = new Student();
```

```
            students[i].getStudentDetails();
            students[i].getMarks();
            students[i].computeSGPA();
        }

        System.out.println("\n--- Student Results ---");
        for (int i = 0; i < n; i++) {
            students[i].display();
        }

        scanner.close();
    }
```

Terminal Output :

```
Enter number of students: 2

Enter details for student 1:
Enter student name: Rishi Kumar Chourasia
Enter student USN: 1BF24CS253
Enter marks for subject 1: 95
Enter credits for subject 1: 4
Enter marks for subject 2: 90
Enter credits for subject 2: 4
Enter marks for subject 3: 85
Enter credits for subject 3: 3
Enter marks for subject 4: 80
Enter credits for subject 4: 3
Enter marks for subject 5: 87
Enter credits for subject 5: 3
Enter marks for subject 6: 82
Enter credits for subject 6: 3
Enter marks for subject 7: 90
Enter credits for subject 7: 1
Enter marks for subject 8: 77
Enter credits for subject 8: 1

Enter details for student 2:
Enter student name: Sagarmatha Khatri
Enter student USN: 1BF24CS260
Enter marks for subject 1: 80
Enter credits for subject 1: 4
Enter marks for subject 2: 75
Enter credits for subject 2: 4
Enter marks for subject 3: 73
Enter credits for subject 3: 3
Enter marks for subject 4: 77
Enter credits for subject 4: 3
Enter marks for subject 5: 90
Enter credits for subject 5: 3
Enter marks for subject 6: 55
Enter credits for subject 6: 3
Enter marks for subject 7: 85
Enter credits for subject 7: 1
Enter marks for subject 8: 90
Enter credits for subject 8: 1

--- Student Results ---

Student Details:
Name: Rishi Kumar Chourasia
USN: 1BF24CS253
SGPA: 9.363636363636363

Student Details:
Name: Sagarmatha Khatri
USN: 1BF24CS260
SGPA: 8.318181818181818
```

Lab Program 3

Implement Bookstore management program

CODE

```
import java.util.Scanner;

class Books {
   String name;
   String author;
   int price;
   int num_pages;

   Books(String name, String author, int price, int num_pages)
   {
      this.name = name;
      this.author = author;
      this.price = price;
      this.num_pages
 = num_pages
;
   }
   public String toString()
   {
      String name = "Book name: " + this.name + "\n";
      String author = "Author name: " + this.author + "\n";
      String price = "Price: " + this.price + "\n";
      String num_pages
 = "Number of pages: " + this.num_pages
 + "\n";
      return name + author + price + num_pages
;
   }
}

public class Main
{
   public static void main(String args[])
   {
      Scanner in = new Scanner(System.in);
      String name,author;
      int price, num_pages
;

      System.out.print("Enter number of books: ");
      int n = in.nextInt();
```

```
        Books[] b = new Books[n];

        for (int i = 0; i < n; i++)
        {
            System.out.println("\n\nEnter details for book " + (i + 1) + ":");

            System.out.print("Name: ");
            name = in.next();

            System.out.print("Author: ");
            author = in.next();

            System.out.print("Price: ");
            price = in.nextInt();

            System.out.print("Number of pages: ");
            num_pages
     = in.nextInt();

            b[i] = new Books(name, author, price, num_pages
    );
        }

        System.out.println("\n\nBook details:\n");

        for (int i = 0; i < n; i++)
        {
            System.out.println(b[i].toString());
        }

        in.close();
    }
}
```

Terminal Output :

```
C:\Users\Admin\Desktop\1BF24CS253>java Main
Enter number of books: 2


Enter details for book 1:
Name: Harry_Potter
Author: JK_Rowling
Price: 450
Number of pages: 223


Enter details for book 2:
Name: Percy_Jackson
Author: Rick_Riordan
Price: 399
Number of pages: 350


Book details:

Book name: Harry_Potter
Author name: JK_Rowling
Price: 450
Number of pages: 223

Book name: Percy_Jackson
Author name: Rick_Riordan
Price: 399
Number of pages: 350
```

Lab Program 4

Implement a program to calculate attributes of different shapes

CODE

```
import java.util.Scanner;

class InputScanner
{
   Scanner in = new Scanner(System.in);
}

abstract class Shape extends InputScanner
{
   int x, y;

   abstract void printArea();

   void inputDimensions(int shapecode)
   {
      if(shapecode==1)
      {
            System.out.print("Enter length: ");
            x = in.nextInt();
            System.out.print("Enter breadth: ");
            y = in.nextInt();
      }

      else if(shapecode==2)
      {
            System.out.print("Enter base: ");
            x = in.nextInt();
            System.out.print("Enter height: ");
            y = in.nextInt();
      }

      else if(shapecode==3)
      {
            System.out.print("Enter radius: ");
            x = in.nextInt();
      }
      else
      {
            System.out.println("Error");
      }
   }
```

```
}

class Rectangle extends Shape {
   void printArea() {
      double area = x * y;
      System.out.println("Area of Rectangle: " + area);
   }
}

class Triangle extends Shape {
   void printArea() {
      double area = 0.5 * x * y;
      System.out.println("Area of Triangle: " + area);
   }
}

class Circle extends Shape {
   void printArea() {
      double area = Math.PI * x * x;
      System.out.println("Area of Circle: " + area);
   }
}

public class Shapemain
{
   public static void main(String[] args)
   {
      Rectangle rect = new Rectangle();
      Triangle tri = new Triangle();
      Circle cir = new Circle();

      System.out.println("\nRectangle");
      rect.inputDimensions(1);
      rect.printArea();

      System.out.println("\nTriangle");
      tri.inputDimensions(2);
      tri.printArea();

      System.out.println("\nCircle");
      cir.inputDimensions(3);
      cir.printArea();
   }
}
```

Terminal Output :

```
Rectangle
Enter length: 12
Enter breadth: 24
Area of Rectangle: 288.0

Triangle
Enter base: 5
Enter height: 12
Area of Triangle: 30.0

Circle
Enter radius: 10
Area of Circle: 314.1592653589793
```

Lab Program 5

Implement a program for a banking software

CODE

```
import java.util.Scanner;

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

   Account(String customerName, String accountNumber, String accountType, double initialBalance)
{
      this.customerName = customerName;
      this.accountNumber = accountNumber;
      this.accountType = accountType;
      this.balance = initialBalance;
   }

   void deposit(double amount)
   {
      if (amount > 0)
      {
         balance += amount;
         System.out.println("Deposited: " + amount);
      }
      else
      {
         System.out.println("Invalid deposit amount.");
      }
   }


   void displayBalance() {
      System.out.println("Current Balance: " + balance);
   }
}

class Saving extends Account
{
   double INTEREST_RATE = 0.05;

   Saving(String customerName, String accountNumber, double initialBalance)
```

```
    {
        super(customerName, accountNumber, "Savings", initialBalance);
    }

    void Interest(int years) {
        double interest = balance * Math.pow((1 + INTEREST_RATE), years) - balance;
        balance += interest;
        System.out.println("Interest of " + String.format("%.2f", interest) + " added to your account.");
    }
    void withdraw(double amount)
    {
        if (amount <= balance)
        {
            balance -= amount;
            System.out.println("Withdrawn:" + amount);
        } else
        {
            System.out.println("Insufficient balance!");
        }
    }
}

class Current extends Account {
    static final double MIN_BALANCE = 1000.0;
    static final double SERVICE_CHARGE = 50.0;

    public Current(String customerName, String accountNumber, double initialBalance)
    {
        super(customerName, accountNumber, "Current", initialBalance);
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: ₹" + amount);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient balance!");
        }
    }

    void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Balance below minimum! Service charge of ₹" + SERVICE_CHARGE +
" imposed.");
        }
```

```
    }
}

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

        System.out.println("Enter Customer Name:");
        String name = in.nextLine();

        System.out.println("Enter Account Number:");
        String accNo = in.nextLine();

        System.out.println("Enter Account Type (savings/current):");
        String type = in.nextLine().toLowerCase();

        System.out.println("Enter Initial Balance:");
        double balance = in.nextDouble();

        Account account;

        if (type.equals("savings"))
        {
            account = new Saving(name, accNo, balance);
        }
        else
        {
            account = new Current(name, accNo, balance);
        }

        int choice;
        do {
            System.out.println("\nOptions Available");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            if (account instanceof Saving)
                System.out.println("4. Compute and Deposit Interest");
            System.out.println("5 . Exit");
            System.out.print("Enter choice: ");
            choice = in.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = in.nextDouble();
                    account.deposit(depositAmount);
```

```
                break;

            case 2:
                System.out.print("Enter amount to withdraw: ");
                double withdrawAmount = in.nextDouble();
                if (account instanceof Saving) {
                    ((Saving) account).withdraw(withdrawAmount);
                } else {
                    ((Current) account).withdraw(withdrawAmount);
                }
                break;

            case 3:
                account.displayBalance();
                break;

            case 4:
                if (account instanceof Saving) {
                    System.out.print("Enter number of years for interest: ");
                    int years = in.nextInt();
                    ((Saving) account).Interest(years);
                } else {
                    System.out.println("Interest computation not available for Current Account.");
                }
                break;

            case 5:
                System.out.println("Exiting Program ");
                break;

            default:
                System.out.println("Invalid choice. Try again.");
            }
        } while (choice != 0);

        in.close();
    }
}
```

Terminal Output :

```
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\1bf24cs253>javac main.java

C:\Users\Admin\Desktop\1bf24cs253>java main
Enter Customer Name:
Rishi Kumar Chourasia
Enter Account Number:
99887766
Enter Account Type (savings/current):
savings
Enter Initial Balance:
10000

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 1
Enter amount to deposit: 100
Deposited: 100.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 2
Enter amount to withdraw: 500
Withdrawn:500.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 3
Current Balance: 9600.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 4
Enter number of years for interest: 3
Interest of 1513.20 added to your account.

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 5
Exiting Program
```

Lab Program 6

Implement a program to show packages in java

CODE

Package CIE Programs :

1.

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {
   protected int marks[] = new int[5];

   public void inputCIEmarks() {
      Scanner in = new Scanner(System.in);
      System.out.println("Enter CIE marks for 5 subjects:");
      for (int i = 0; i < 5; i++) {
         System.out.print("CIE Mark " + (i + 1) + ": ");
         marks[i] = in.nextInt();
      }
   }
}
```

2.

```
package CIE;

import java.util.Scanner;

public class Student {
   protected String usn;
   protected String name;
   protected int sem;

   public void inputStudentDetails() {
      Scanner in = new Scanner(System.in);
      System.out.print("Enter USN: ");
      usn = in.nextLine();
      System.out.print("Enter Name: ");
      name = in.nextLine();
      System.out.print("Enter Semester: ");
      sem = in.nextInt();
```

```
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

Package SEE Program :

```
package SEE;

import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {

    protected int marksSEE[];
    protected int finalMarks[];

    public Externals() {
        marksSEE = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("SEE Mark " + (i + 1) + ": ");
            marksSEE[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + marksSEE[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final marks (CIE + SEE):");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
```

```
        }
    }
}
```

Main Program :

```
import SEE.Externals;

class Main {
    public static void main(String args[]) {
        int n = 0;
        java.util.Scanner s = new java.util.Scanner(System.in);

        System.out.print("Enter number of students: ");
        n = s.nextInt();

        Externals arr[] = new Externals[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details of student " + (i + 1));
            arr[i] = new Externals();
            arr[i].inputStudentDetails();
            arr[i].inputCIEmarks();
            arr[i].inputSEEmarks();
            arr[i].calculateFinalMarks();
        }

        System.out.println("\nFINAL MARKS ");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            arr[i].displayFinalMarks();
        }
    }
}
```

Terminal Output :

```
Enter Semester: 2
Enter CIE marks for 5 subjects:
CIE Mark 1: 41
CIE Mark 2: 45
CIE Mark 3: 44
CIE Mark 4: 50
CIE Mark 5: 47
Enter SEE marks for 5 subjects:
SEE Mark 1: 50
SEE Mark 2: 40
SEE Mark 3: 44
SEE Mark 4: 41
SEE Mark 5: 38

Enter details of student 2
Enter USN: 1BF24CS260
Enter Name: Sagarmatha Khatri
Enter Semester: 2
Enter CIE marks for 5 subjects:
CIE Mark 1: 35
CIE Mark 2: 45
CIE Mark 3: 43
CIE Mark 4: 41
CIE Mark 5: 40
Enter SEE marks for 5 subjects:
SEE Mark 1: 45
SEE Mark 2: 41
SEE Mark 3: 42
SEE Mark 4: 40
SEE Mark 5: 38

FINAL MARKS

Student 1:
USN: 1BF24CS253
Name: Rishi Kumar Chourasia
Semester: 2
Final marks (CIE + SEE):
Subject 1: 91
Subject 2: 85
Subject 3: 88
Subject 4: 91
Subject 5: 85

Student 2:
USN: 1BF24CS260
Name: Sagarmatha Khatri
Semester: 2
Final marks (CIE + SEE):
Subject 1: 80
Subject 2: 86
Subject 3: 85
Subject 4: 81
Subject 5: 78

C:\Users\Admin\Desktop\1bf24cs253\Lab Program-6 Package>
```

Lab Program 7

Implement a java program to show error handling

CODE

```
import java.util.Scanner;

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

class Father
{
   int fatherAge;

   Father(int age) throws WrongAge
   {
      if (age<0)
      {
         throw new WrongAge("Father's age cannot be negative!");
      }
      this.fatherAge = age;
   }
   void display()
   {
      System.out.println("Father's age : "+fatherAge);
   }
}


class Son extends Father{
   int sonAge;

   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!"
         );
```

```
        }

        if (sonAge<0)
        {
           throw new WrongAge("Son's age cannot be negative!");
        }

        this.sonAge=sonAge;
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int s_age,f_age;
        System.out.println("Enter father's age : ");
        f_age = in.nextInt();
        System.out.println("Enter son's age : ");
        s_age = in.nextInt();
        try{
             Son s = new Son(f_age,s_age);
             System.out.println("Father and Son objects created successfully!");
             s.display();
        }
        catch(WrongAge e)
        {
           System.out.println("Wrong Age Error: " + e.getMessage());
        }
    }
}
```

Terminal Output :

```
PS C:\Users\Admin\Desktop\1BF24CS253> cd "c:\Users\Admin\Desktop\1BF24CS253\" ; if ($?) { javac newMain.java } ; if ($?) { java newMain }
Enter father's age : 10
Enter son's age : 15
Wrong Age Error: Son's age cannot be greater than or equal to father's age!
PS C:\Users\Admin\Desktop\1BF24CS253> cd "c:\Users\Admin\Desktop\1BF24CS253\" ; if ($?) { javac newMain.java } ; if ($?) { java newMain }
Enter father's age : 30
Enter son's age : -1
Wrong Age Error: Son's age cannot be negative!
PS C:\Users\Admin\Desktop\1BF24CS253> cd "c:\Users\Admin\Desktop\1BF24CS253\" ; if ($?) { javac newMain.java } ; if ($?) { java newMain }
Enter father's age : 35
Enter son's age : 15
Father and Son objects created successfully!
Father's age : 35
PS C:\Users\Admin\Desktop\1BF24CS253>
```

Lab Program 8

Implement a java program to show working of Threads

CODE

```
class MessageThread extends Thread {
   private String message;
   private int interval;

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

   public void run() {
      try {
         while (true) {
            System.out.println(message);
            Thread.sleep(interval);
         }
      } catch (InterruptedException e) {
         System.out.println("Thread interrupted.");
      }
   }
}

public class Main {
   public static void main(String[] args) {
      MessageThread t1 = new MessageThread("BMS College of Engineering", 10000);
      MessageThread t2 = new MessageThread("CSE", 2000);

      t1.start();
      t2.start();
   }
}
```

Terminal Output :

```
PS C:\Users\Admin\Desktop\1bf24cs253> cd "c:\Users\Admin\Desktop\1bf24cs253\" ; if ($?) { javac Main.java } ; if ($?) { ja
va Main }
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
PS C:\Users\Admin\Desktop\1bf24cs253>
```

Lab Program 9

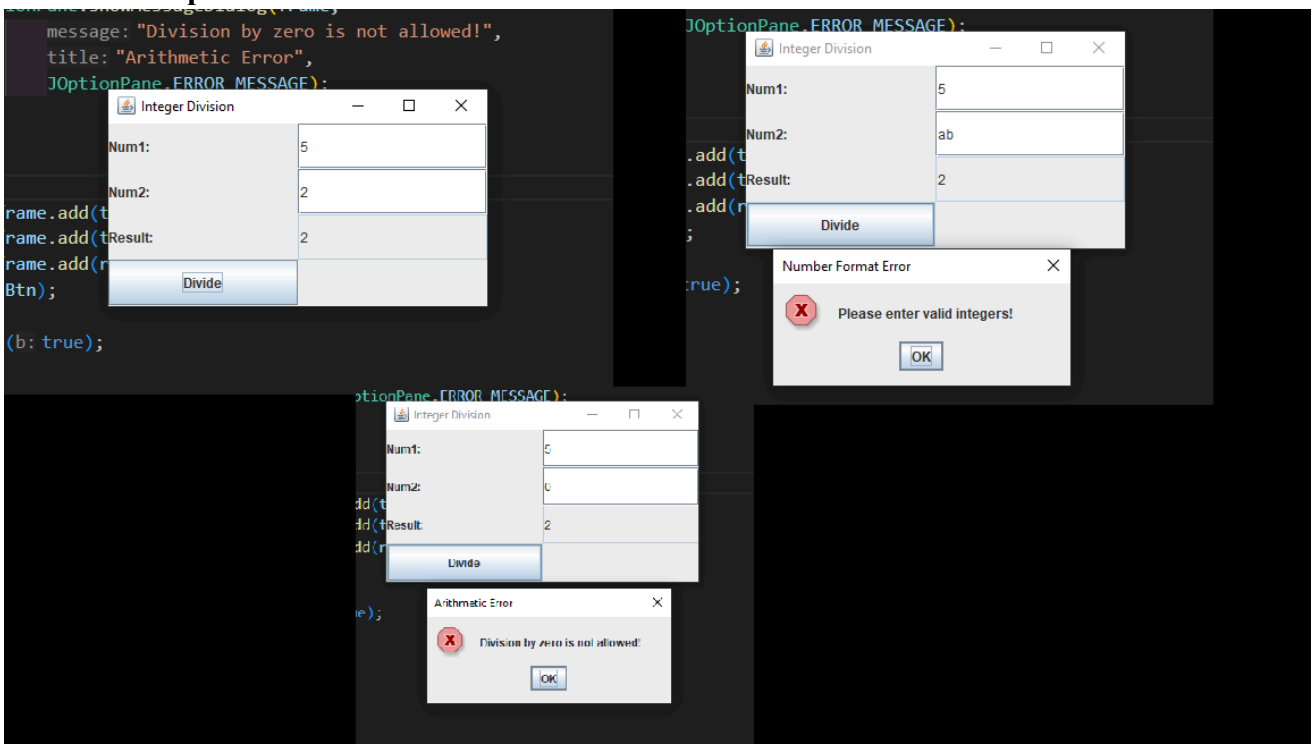
Open Ended Question 1

CODE

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

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

      JFrame frame = new JFrame("Integer Division");
      frame.setSize(350, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setLayout(new GridLayout(4, 2));

      JLabel l1 = new JLabel("Num1:");
      JTextField t1 = new JTextField();

      JLabel l2 = new JLabel("Num2:");
      JTextField t2 = new JTextField();

      JLabel l3 = new JLabel("Result:");
      JTextField result = new JTextField();
      result.setEditable(false);

      JButton divideBtn = new JButton("Divide");

      divideBtn.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            try {
               int num1 = Integer.parseInt(t1.getText());
               int num2 = Integer.parseInt(t2.getText());

               int res = num1 / num2;
               result.setText(Integer.toString(res));
            }
            catch (NumberFormatException ex) {
               JOptionPane.showMessageDialog(frame,
                     "Please enter valid integers!",
                     "Number Format Error",
```

```
                JOptionPane.ERROR_MESSAGE);
            }
            catch (ArithmeticException ex) {
                JOptionPane.showMessageDialog(frame,
                    "Division by zero is not allowed!",
                    "Arithmetic Error",
                    JOptionPane.ERROR_MESSAGE);
            }
        }
    });

    frame.add(l1); frame.add(t1);
    frame.add(l2); frame.add(t2);
    frame.add(l3); frame.add(result);
    frame.add(divideBtn);

    frame.setVisible(true);
  }
}
```

Terminal Output :



Lab Program 10

Open Ended Question 2

CODE

```
class Q {
   int n;
   boolean valueSet = false;

   synchronized int get() {
      while (!valueSet) {
         try {
            System.out.println("  Consumer waiting");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      }

      System.out.println("Got: " + n);
      valueSet = false;

      System.out.println("Intimate Producer");
      notify();
      return n;
   }

   synchronized void put(int n) {
      while (valueSet) {
         try {
            System.out.println("Producer waiting");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      }

      this.n = n;
      valueSet = true;

      System.out.println("Put: " + n);
      System.out.println("Intimate Consumer");
      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 < 3) {
            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 < 3) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

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

Terminal Output :

```
PS C:\Users\Admin\Desktop\1bf24cs253> cd "c:\Users\Admin\Desktop\1bf24cs253\" ; if ($?) { javac PCDemo.java } ; if ($?) {
java PCDemo }
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
Consumed: 0
Got: 1
Intimate Producer
Consumed: 1
Put: 2
Intimate Consumer
Got: 2
Intimate Producer
Consumed: 2
PS C:\Users\Admin\Desktop\1bf24cs253>
```