VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

TANMAY (1BM23ME115)

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)

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 **Tanmay (1BM23ME115),** 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 Incharge Name	Dr. Jyothi S Nayak
Assistant Professor	Professor & HOD
Department of CSE, BMSCE	Department of CSE, BMSCE

Index

Github Link:

<u>Program 1</u>

Implement Quadratic Equation

Algorithm:

Lab Program 1

~~import java.util.Scanner~~

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

~~import java.~~

```java
import java.util.scanner;

class Quadratic {
    int a, b, c;
    double r1, r2, d;

    void getd() {
        scanner s = new Scanner (System.in);
        System.out.println ("Enter the coefficients of a, b, c:");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }

    void compute() {
        if (a == 0) {
            System.out.println ("Not a quadratic equation");
            return;
        }
        d = b*b - 4*a*c;
```

```java
            if (d < 0) {
                System.out.println("There are no real solutions.");
            }
            else if (d == 0) {
                x1 = -b / (2.0 * a);
                System.out.println("Roots are real and equal");
                System.out.println("Root1 = Root2 = " + x1);
            }

            else {
                x1 = (-b + Math.sqrt(d)) / (2.0 * a);
                x2 = (-b + - Math.sqrt(d)) / (2.0 * a);
                System.out.println("Roots are real and distinct.");
                System.out.println("Root1 = " + x1 + ", Root2 = " + x2);
            }
        }
    }
}
public class QuadraticMain {
    public static void main(String[] args) {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("TANMAY");
        System.out.println("IBM23ME115");
    }
}
```

Output:

Enter the coefficients of a, b, c:

2

3

1

Roots are real and distinct.

Root 1 = -0.5, Root 2 = -1.0

JANMAY

1BM23ME115

Code:

```java
import java.util.Scanner;

class Quadratic {
    int a, b, c;
    double r1, r2, d;


    void getd() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a, b, c:");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }


    void compute() {

        if (a == 0) {
            System.out.println("Not a quadratic equation");
            return;
        }


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


        if (d < 0) {
            System.out.println("There are no real solutions.");
        }

        else if (d == 0) {
            r1 = -b / (2.0 * a);
            System.out.println("Roots are real and equal.");
            System.out.println("Root1 = Root2 = " + r1);
        }

        else {
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
            r2 = (-b - Math.sqrt(d)) / (2.0 * a);
            System.out.println("Roots are real and distinct.");
            System.out.println("Root1 = " + r1 + ", Root2 = " + r2);
        }
    }
}
```

```
public class QuadraticMain {
    public static void main(String[] args) {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
            System.out.println("TANMAY");
            System.out.println("1BM23ME115");
    }
}
```

```
D:\1BM23ME115\Program 1>javac QuadraticMain.java

D:\1BM23ME115\Program 1>java QuadraticMain
Enter the coefficients of a, b, c:
2
3
1
Roots are real and distinct.
Root1 = -0.5, Root2 = -1.0
TANMAY
1BM23ME115
```

Program 2

SGPA Calculation

Algorithm:

Lab Program - 2

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

```java
import java.util.scanner;
class Subject {
    int subjectMarks;
    int credits;
    int grade;
}

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

    Student () {
        subject = new Subject [9];
        for (int i=0; i<9; i++){
            subject [i] = new Subject ();
        }
        s = new Scanner (System.in);
    }
}
```

```java
void getStudentDetails() {
    System.out.print("Enter your Name : ");
    name = s.next();
    System.out.print("Enter your USN : ");
    usn = s.next();
}

void getMarks() {
    for (int i = 0; i < 4; 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();

        int marks = subject[i].subjectMarks;
        if (marks >= 90) subject[i].grade = 10;
        else if (marks >= 80) subject[i].grade = 9;
        else if (marks >= 70) subject[i].grade = 8;
        else if (marks >= 60) subject[i].grade = 7;
        elsif (marks >= 50) subject[i].grade = 6;
        else if (marks >= 40) subject[i].grade = 5;
        else subject[i].grade = 0;
    }
}
```

```java
    void compute SGPA() {
        int effectiveScore = 0;
        int totalCredits = 0;
        for (int i = 0; i < 4; i++) {
            effectiveScore += (subject[i].grade * subject[i].credits)
            totalCredits += subject[i].
        }
        SGPA = (double) effectiveScore / totalCredits;
    }
}

public class Main {
    public static void main(String args[]) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("Name" + s1.name);
        System.out.println("USN: " + s1.usn);
        System.out.println("SGPA : " + s1.SGPA);
        System.out.println("IBM23ME115" - TANMAY");
    }
}
```

Output :

Enter y
Enter
Enter
Enter
Enter
Enter
Ent
En
Fr
E
F
E

Output :

Enter your Name : Tanmay
Enter your USN : 1BM23ME115
Enter marks for subject 1 : 95
Enter credits for subject 1 : 4
Enter marks for subject 2 : 92
Enter credit for subject 2 : 4
Enter credit for subject 3 : 87
Enter credits for subject 3 : 3
Enter marks for subject 4 : 76
Enter credits for subject 4 : 3
Enter marks for subject 5 : 81
Enter credits for subject 5 : 3
Enter marks for subject 6 : 96
"   credits "   " : 2
Enter marks for subject 7 : 98
"   credits for "   : 1
Enter marks for subject 8 : 100
"   credits "   " : 1
Enter marks for subject 9 ; 93
Enter credits for subject 9 : 1

Name : Tanmay
USN : 1BM23ME115

SGPA : 9.4545 45 45 4545

1BM23ME115 - TANMAY

Code:

```
import java.util.Scanner;

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

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


    Student() {
        subject = new Subject[9];
        for (int i = 0; i < 9; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }


    void getStudentDetails() {
        System.out.print("Enter your Name: ");
        name = s.next();
        System.out.print("Enter your USN: ");
        usn = s.next();
    }


    void getMarks() {
        for (int i = 0; i < 9; 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();
```

```java
        int marks = subject[i].subjectMarks;
        if (marks >= 90) subject[i].grade = 10;
        else if (marks >= 80) subject[i].grade = 9;
        else if (marks >= 70) subject[i].grade = 8;
        else if (marks >= 60) subject[i].grade = 7;
        else if (marks >= 50) subject[i].grade = 6;
        else if (marks >= 40) subject[i].grade = 5;
        else subject[i].grade = 0;
      }
    }


    void computeSGPA() {
      int effectiveScore = 0;
      int totalCredits = 0;
      for (int i = 0; i < 9; i++) {
        effectiveScore += (subject[i].grade * subject[i].credits);
        totalCredits += subject[i].credits;
      }
      SGPA = (double) effectiveScore / totalCredits;
    }
}

public class Main {
    public static void main(String args[]) {
      Student s1 = new Student();
      s1.getStudentDetails();
      s1.getMarks();
      s1.computeSGPA();
      System.out.println("Name: " + s1.name);
      System.out.println("USN: " + s1.usn);
      System.out.println("SGPA: " + s1.SGPA);
        System.out.println("1BM23ME115 - TANMAY");
    }
}
```

Output

```
D:\1BM23ME115\LAB-2>javac Main.java

D:\1BM23ME115\LAB-2>java Main
Enter your Name: Tanmay
Enter your USN: 1BM23ME115
Enter marks for subject 1: 95
Enter credits for subject 1: 4
Enter marks for subject 2: 92
Enter credits for subject 2: 4
Enter marks for subject 3: 87
Enter credits for subject 3: 3
Enter marks for subject 4: 76
Enter credits for subject 4: 3
Enter marks for subject 5: 81
Enter credits for subject 5: 3
Enter marks for subject 6: 96
Enter credits for subject 6: 2
Enter marks for subject 7: 98
Enter credits for subject 7: 1
Enter marks for subject 8: 100
Enter credits for subject 8: 1
Enter marks for subject 9: 93
Enter credits for subject 9: 1
Name: Tanmay
USN: 1BM23ME115
SGPA: 9.454545454545455
1BM23ME115 - TANMAY
```

Program 3

Create n book objects

Algorithm:

Lab Program 3

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

```java
import java.util.scanner;

class Book {
    private String name, author;
    private double price;
    private int numPages;

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

    @Override
    public String toString(){
        return "Book Details :\n Name :" + "\nAuthor :" + author +
            "\n Price : $" + price + "\nPages :" + numPages;
    }
}
```

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

        System.out.print ("Enter the number of books: ");
        int n = scanner.nextInt ();
        scanner.nextLine ();

        Book [] books = new Book [n];

        for (int i=0; i<n; i++) {
            System.out.println ("\n Enter 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 ("Pages: ");
            int pages = scanner.nextInt ();

            scanner.nextLine ();
            books [i] = new Book(name, author, price, pages);
        }

        System.out.println ("\n Books Entered : ");
        for (Book book : books) {
                System.out.println (book);
        }

        scanner.close ();
    }
}
```

Output:

Enter the number of books : 2

Enter details for Book 1 :
Name : The Alchemist
Author : Paulo Coelho
Price : 15.99
Pages : 208

Enter details for Book 2 :
Name : 1984
Author : George Orwell
Price : 12.50
Pages : 328

Books Entered :
Book Details :

Name : The Alchemist
Author : Paulo Coelho
Price : $15.99
Pages : 208

Book Details :
Name : 1984
Author : George Orwell
Price : $12.50
Pages : 328

Code:

```java
import java.util.Scanner;

class Book {
    private String name, author;
    private double price;
    private int numPages;

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

    @Override
    public String toString() {
        return "Book Details:\nName: " + name + "\nAuthor: " + author + "\nPrice: $" + price +
"\nPages: " + numPages;
    }
}

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

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            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("Pages: ");
            int pages = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            books[i] = new Book(name, author, price, pages);
        }
```

```java
        System.out.println("\nBooks Entered:");
        for (Book book : books) {
            System.out.println(book);
        }

        scanner.close();
    }
}
```

```
Enter the number of books: 2

Enter details for Book 1:
Name: The Alchemist
Author: Paulo Coelho
Price: 15.99
Pages: 208

Enter details for Book 2:
Name: 1984
Author: George Orwell
Price: 12.50
Pages: 328

Books Entered:
Book Details:
Name: The Alchemist
Author: Paulo Coelho
Price: $15.99
Pages: 208

Book Details:
Name: 1984
Author: George Orwell
Price: $12.50
Pages: 328
```

Program 4

Print Area of Different Shapes

Algorithm:

Lab Program 4

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

```java
import java.util.scanner;

abstract class Shape {
        int dim1, dim2;
        Shape (int dim1, int dim2) {
                this.dim1 = dim1;
                this.dim2 = dim2;
        }
        abstract void printArea();
}

class Rectangle extends Shape {
        Rectangle (int length, int breadth) {
                super (length, breadth);
        }
        void printArea() {
                System.out.println ("Area of Rectangle: " + (dim1 * dim2));
        }
}
```

```java
class Triangle extends Shape {
    Triangle (int base, int height) {
        super (base, height);
    }
    void perimeter()
    void printArea() {
        System.out.println ("Area of Triangle:" + (0.5 * dim1 * dim2));
    }
};

class Circle extends Shape {
    Circle (int radius) {
        super (radius, 0);
    }
    void printArea () {
        System.out.println ("Area of Circle:" + (Math.PI * dim1 * dim1));
    }
}

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

        System.out.println (" Enter length & Breadth of rectangle:");
        Shape rectangle = new Rectangle (scanner.nextInt(), scanner.nextInt());

        System.out.print (" Enter base & height of triangle:");
        Shape triangle = new Triangle (scanner.nextInt(), scanner.nextInt());

        System.out.print (" Enter radius of circle:");
        Shape circle = new Circle (scanner.nextInt());
```

```
rectangle.printArea();
triangle.printArea();
circle.printArea();

        scanner.close();
    }
}
```

Output :

Enter length & breadth of rectangle : 5   10
Enter base & height of triangle : 4 6
Enter radius of circle : 7

Calculating Areas :

Area of rectangle : 50
Area of triangle : 12.0
Area of circle : 153.93804002584985

Code:

```java
import java.util.Scanner;

abstract class Shape {
   int dim1, dim2;

   Shape(int dim1, int dim2) {
      this.dim1 = dim1;
      this.dim2 = dim2;
   }

   abstract void printArea();
}

class Rectangle extends Shape {
   Rectangle(int length, int breadth) {
      super(length, breadth);
   }

   void printArea() {
      System.out.println("Area of Rectangle: " + (dim1 * dim2));
   }
}

class Triangle extends Shape {
   Triangle(int base, int height) {
      super(base, height);
   }

   void printArea() {
      System.out.println("Area of Triangle: " + (0.5 * dim1 * dim2));
   }
}

class Circle extends Shape {
   Circle(int radius) {
      super(radius, 0);
   }


   void printArea() {
      System.out.println("Area of Circle: " + (Math.PI * dim1 * dim1));
   }
}
```

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

        System.out.print("Enter length and breadth of the rectangle: ");
        Shape rectangle = new Rectangle(scanner.nextInt(), scanner.nextInt());

        System.out.print("Enter base and height of the triangle: ");
        Shape triangle = new Triangle(scanner.nextInt(), scanner.nextInt());

        System.out.print("Enter radius of the circle: ");
        Shape circle = new Circle(scanner.nextInt());

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();

        scanner.close();
    }
}
```

Output:

```
Enter length and breadth of the rectangle: 5 10
Enter base and height of the triangle: 4 6
Enter radius of the circle: 7

Calculating Areas:
Area of Rectangle: 50
Area of Triangle: 12.0
Area of Circle: 153.93804002589985
```

Program 5

Create a Class Bank

Algorithm:

Lab Program 5

Develop a Java program to create a class Bank. that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility 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 name, 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 & update the balance.

```java
import java.util.Scanner;

abstract class Account {
        String name, accNo;
        double balance;
```

```java
Account (String name, String accNo, double balance){
                this.name = name;
                this.accNo = accNo;
                this.balance = balance;
    }

    void deposit (double amount){
            balance += amount;
            System.out.println("Deposited: $" + amount);
    }
    void displayBalance (){
            System.out.println("Balance: $" + balance);
    }
    abstract void withdraw (double amount);

}

class SavAcct extends Account {
    SavAcct (String name, String accNo, double balance){
            super (name, accNo, balance);
    }
    void computeInterest (){
            balance += balance * 0.05;
            System.out.println("Interest added.");
    }
    void withdraw (double amount){
            if (amount <= balance) balance -= amount;
            else System.out.println("Insufficient balance.");
    }

}
```

```java
class CurAcct extends Account {
    CurAcct (String name, String accNo, double balance) {
        super (name, accNo, balance);
    }
    void withdraw (double amount) {
        if (amount <= balance) {
            balance -= amount;
            if (balance < 1000) balance -= 50;
        } else System.out.println ("Insufficient balance.");
    }
}

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

        System.out.print ("Enter name & balance for Savings Account:");
        SavAcct sa = new SavAcct (sc.next(), "SA001", sc.nextDouble());
        System.out.print ("Enter name & balance for Current Account:");
        CurAcct ca = new CurAcct (sc.next(), "CA001", sc.nextDouble());

        System.out.println ("\n Savings Account Operations: ");
        sa.deposit (500);
        sa.computeInterest ();
        sa.withdraw (300);
        sa.displayBalance ();

        System.out.println ("\n Current Account Operations: ");
        ca.deposit (200);
        ca.withdraw (400);
        ca.withdraw (1200);
        ca.displayBalance ();

        sc.close ();
    }
}
```

Output:

Enter custo...
John
Enter initi...
2000
Enter cust...
Alice
Enter ini...
1500

Performi...
Balance
Deposite...
Balance
Interest
Balan...
Withd...
Balan...

Perf...
Ba...
De...
Ba...
W...
B...

Output:

Enter customer name for the Savings Account:
John
Enter initial balance for the Savings Account:
2000
Enter customer name for the Current Account:
Alice
Enter initial balance for the Current Account:
1500

Performing operations on Savings Account
Balance: $2000.0
Deposited: $500
Balance: $2500.0
Interest of $125.0 deposited
Balance: $2625.0
Withdrawn: $300
Balance: $2325.0

Performing operations on Current Account
Balance: $1500.0
Deposited: $200
Balance: $1700.0
Withdrawn: $400
Balance: $1300.0
Withdrawn: $1200
Balance below minimum, penalty of $50.0 imposed.
Withdrawn: $1200
Balance: $50.0

Code:

```java
import java.util.Scanner;

abstract class Account {
    String name, accNo;
    double balance;

    Account(String name, String accNo, double balance) {
        this.name = name;
        this.accNo = accNo;
        this.balance = balance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: $" + amount);
    }

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

    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    SavAcct(String name, String accNo, double balance) {
        super(name, accNo, balance);
    }

    void computeInterest() {
        balance += balance * 0.05;
        System.out.println("Interest added.");
    }

    void withdraw(double amount) {
        if (amount <= balance) balance -= amount;
        else System.out.println("Insufficient balance.");
    }
```

```java
}

class CurAcct extends Account {
    CurAcct(String name, String accNo, double balance) {
        super(name, accNo, balance);
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            if (balance < 1000) balance -= 50; // Penalty
        } else System.out.println("Insufficient balance.");
    }
}

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

        System.out.print("Enter name and balance for Savings Account: ");
        SavAcct sa = new SavAcct(sc.next(), "SA001", sc.nextDouble());
        System.out.print("Enter name and balance for Current Account: ");
        CurAcct ca = new CurAcct(sc.next(), "CA001", sc.nextDouble());

        System.out.println("\nSavings Account Operations:");
        sa.deposit(500);
        sa.computeInterest();
        sa.withdraw(300);
        sa.displayBalance();

        System.out.println("\nCurrent Account Operations:");
        ca.deposit(200);
        ca.withdraw(400);
        ca.withdraw(1200);
        ca.displayBalance();

        sc.close();
    }
}
```

Output:

```
Enter customer name for the Savings Account:
John
Enter initial balance for the Savings Account:
2000
Enter customer name for the Current Account:
Alice
Enter initial balance for the Current Account:
1500

Performing operations on Savings Account
Balance: $2000.0
Deposited: $500
Balance: $2500.0
Interest of $125.0 deposited.
Balance: $2625.0
Withdrawn: $300
Balance: $2325.0

Performing operations on Current Account
Balance: $1500.0
Deposited: $200
Balance: $1700.0
Withdrawn: $400
Balance: $1300.0
Withdrawn: $1200
Balance below minimum, penalty of $50.0 imposed.
Withdrawn: $1200
Balance: $50.0
```

Program 6

<u>Packages</u>

Algorithm:

Create a package CIE which has two classes - Personal 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 Personal. 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.

CIE/Internals.java

```
package CIE;
public class Internals {
    public int[] marks = new int[5];
    public Internals(int[] marks) {
        System.arraycopy(marks, 0, this.marks, 0, 5);
    }
}
```

CIE/Personal.java

```
package CIE;
public class Personal {
    public String usn, name, sem;
    public Personal(String usn, String name, String sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

```java
SEE/ External.java
package SEE;
import CIE.Personal;
public class External extends Personal {
    public int [] marks = new int [5];
    public External (String usn, String name, String sem, int[] marks){
        super (usn, name, sem);
        System. arraycopy (marks, 0, this.marks, 0, 5);
    }
}


FinalMarks.java

import SEE.*;
import java.util.Scanner;

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

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

        External [] students = new External[n];
        int [] cieMarks = new int [5], seeMarks = new int [5];

        for (int i = 0; i < n; i++){
            System. out. println ("Enter details for Student" + (i+1) + ":");
            System. out. print ("USN: ");
            String usn = sc.next ();
            System. out. print ("Name : ");
            String name = sc. next ();
            System. out. print ("Semester : ");
            String sem = sc.next ();
```

```java
System.out.println ("Enter CIE marks for 5 course:");
for (int j = 0; j < 5; j++)  cieMarks [j] = sc.nextInt();
System.out.println ("Enter SEE marks for 5 courses:");
for (int j = 0; j < 5; j++)  seeMarks [j] = sc.nextInt();

students [i] = new External (usn, name, sem, seeMarks);
System.arraycopy (cieMarks, 0, students[i].marks, 0, 5);
}

System.out.println ("\n Final Marks of students:");
for (int i = 0; i < n; i++) {
    System.out.println ("Student " + (i+1) + " - USN: " +
                                    students [i].usn);
    System.out.println ("Name:" + students[i].name + ",
            Semester: " + students [i].sem);
    System.out.print ("Final Marks: ");
    for (int j = 0; j < 5; j++) {
        int finalMarks = students[i].marks [j] +
                            (students [i].marks [j] / 2);
        System.out.print(finalMarks + " ");
    }
    System.out.println ();
}

sc.close ();
}
}
```

Output:

Enter the number of students : 1

Enter details for Student 1 :
USN : 1BM23ME115
Name : Tanmay
Semester : 3
Enter CIE marks for 5 courses :

18   20   15   19   17

Enter SEE marks for 5 courses :

40   38   45   42   39

Final Marks of Students :
Student 1 - USN : 1BM23ME115
Name : Tanmay , Semester : 3
Final Marks : 38   39   37   40   36

Code:

```
package CIE;

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

   public Internals(int[] marks) {
      System.arraycopy(marks, 0, this.marks, 0, 5);
   }
}

package CIE;

public class Personal {
   public String usn, name, sem;

   public Personal(String usn, String name, String sem) {
      this.usn = usn;
      this.name = name;
      this.sem = sem;
   }
}

package SEE;

import CIE.Personal;

public class External extends Personal {
   public int[] marks = new int[5];

   public External(String usn, String name, String sem, int[] marks) {
      super(usn, name, sem);
      System.arraycopy(marks, 0, this.marks, 0, 5);
   }
}

import SEE.*;
import java.util.Scanner;
```

```java
public class FinalMarks {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

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

      External[] students = new External[n];
      int[] cieMarks = new int[5], seeMarks = new int[5];

      for (int i = 0; i < n; i++) {
         System.out.println("Enter details for Student " + (i + 1) + ":");
         System.out.print("USN: ");
         String usn = sc.next();
         System.out.print("Name: ");
         String name = sc.next();
         System.out.print("Semester: ");
         String sem = sc.next();

         System.out.println("Enter CIE marks for 5 courses:");
         for (int j = 0; j < 5; j++) cieMarks[j] = sc.nextInt();

         System.out.println("Enter SEE marks for 5 courses:");
         for (int j = 0; j < 5; j++) seeMarks[j] = sc.nextInt();

         students[i] = new External(usn, name, sem, seeMarks);
         System.arraycopy(cieMarks, 0, students[i].marks, 0, 5);
      }

      System.out.println("\nFinal Marks of Students:");
      for (int i = 0; i < n; i++) {
         System.out.println("Student " + (i + 1) + " - USN: " + students[i].usn);
         System.out.println("Name: " + students[i].name + ", Semester: " + students[i].sem);
         System.out.print("Final Marks: ");
         for (int j = 0; j < 5; j++) {
            int finalMarks = students[i].marks[j] + (students[i].marks[j] / 2);
            System.out.print(finalMarks + " ");
         }
         System.out.println();
      }
```

```
        sc.close();
    }
}
```

Output:

```
> javac CIE/Personal.java CIE/Internals.java SEE/External.java FinalMarks.java
> java FinalMarks

Enter the number of students: 2

Enter details for Student 1:
USN: 1RV23CS001
Name: Alice
Semester: 5
Enter CIE marks for 5 courses:
18 20 15 19 17
Enter SEE marks for 5 courses:
40 38 45 42 39

Enter details for Student 2:
USN: 1RV23CS002
Name: Bob
Semester: 5
Enter CIE marks for 5 courses:
20 22 18 19 21
Enter SEE marks for 5 courses:
36 40 44 35 38

Final Marks of Students:
Student 1 - USN: 1RV23CS001
Name: Alice, Semester: 5
Final Marks: 38 39 37 40 36

Student 2 - USN: 1RV23CS002
Name: Bob, Semester: 5
Final Marks: 38 42 40 36 40
```

Program 7

Handling of Exceptions

Algorithm:

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 uses both father and son's age and throws an exception if son's age is 7= father's age.

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

class Father {
    int age;
    public Father (int age) throws WrongAge Exception {
        if (age < 0) throw new WrongAge Exception ("Father's age cannot be
                                                    negative.");
        this. age = age ;
        }
}
class Son extends Father {
    int sonAge;
    public Son (int father Age , int sonAge) throws WrongAge Exception{
            super( father Age) ;
        if (sonAge 7= fatherAge) throw new WrongAge Exception (" Son's age
                        cannot be greater than or equal to father's
                        age.");
            this. sonAge = sonAge ;
        }
}
```

47

```java
public class Exception Demo {
    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.age +",
                            Son's Age : " + son. sonAge);
        } catch (Wrong Age Exception e) {
            System.out.println ("Exception : " + e. get Message ());
        }

        try {
            Son invalidSon = new Son (30, 35);
        } catch (Wrong Age Exception e) {
            System.out.print ln (" Exception : " + e.get Message ());
        }
    }
}
```

Output:

Father's Age: 40, Son's Age: 20
Exception: Son's age cannot be greater than or equal to father's age.

Code:

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

// Father class
class Father {
   int age;

   public Father(int age) throws WrongAgeException {
      if (age < 0)
         throw new WrongAgeException("Father's age cannot be negative.");
      this.age = age;
   }
}

// Son class that extends Father
class Son extends Father {
   int sonAge;

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

public class ExceptionDemo {
   public static void main(String[] args) {
      System.out.println("TANMAY - 1BM23ME115");

      try {
         Father father = new Father(40); // Create a Father object
         Son son = new Son(40, 20);     // Create a Son object
         System.out.println("Father's Age: " + father.age + ", Son's Age: " + son.sonAge);
         } catch (WrongAgeException e) {
         System.out.println("Exception: " + e.getMessage());
      }


      try {
```

```
            Son invalidSon = new Son(30, 35); // This will throw an exception
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

E:\LAB-7>javac ExceptionDemo.java

E:\LAB-7>java ExceptionDemo.java
TANMAY - 1BM23ME115
Father's Age: 40, Son's Age: 20
Exception: Son's age cannot be greater than or equal to father's age.

E:\LAB-7>
```
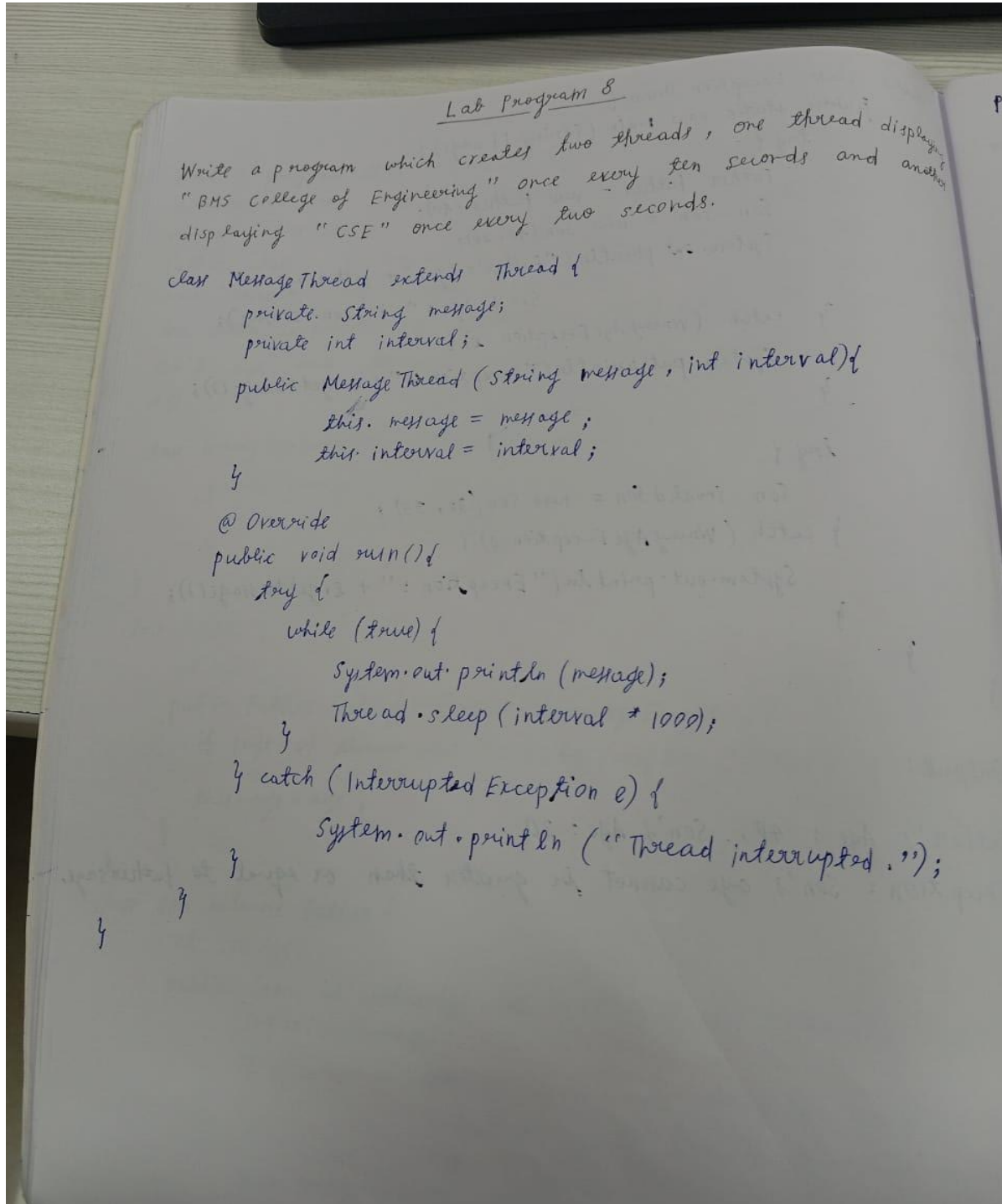
Program 8

Threads

Algorithm:

Lab 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.

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

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

    @Override
    public void run(){
        try {
            while (true){
                System.out.println (message);
                Thread.sleep (interval * 1000);
            }
        } catch (InterruptedException e){
            System.out.println ("Thread interrupted.");
        }
    }
}
```

```
public class MultiThreadDemo{
    public static void main (String[] args){
        new MessageThread ("BMS College of Engineering", 10).start();
        new MessageThread ("CSE", 2).start();
    }
}
```

Output:

CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
...

Code:

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

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

    @Override
    public void run() {
        try {
            while (true) {
                System.out.println(message + " - TANMAY");
                Thread.sleep(interval * 1000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted.");
        }
    }
}

public class MultiThreadDemo {
    public static void main(String[] args) {
        new MessageThread("BMS College of Engineering", 10).start();
        new MessageThread("CSE", 2).start();
    }
}
```

Output

```
Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

E:\LAB-8>javac MultiThreadDemo.java

E:\LAB-8>java MultiThreadDemo.java
BMS College of Engineering - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
BMS College of Engineering - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
BMS College of Engineering - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
BMS College of Engineering - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
BMS College of Engineering - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
BMS College of Engineering - TANMAY
CSE - TANMAY
CSE - TANMAY
CSE - TANMAY
```
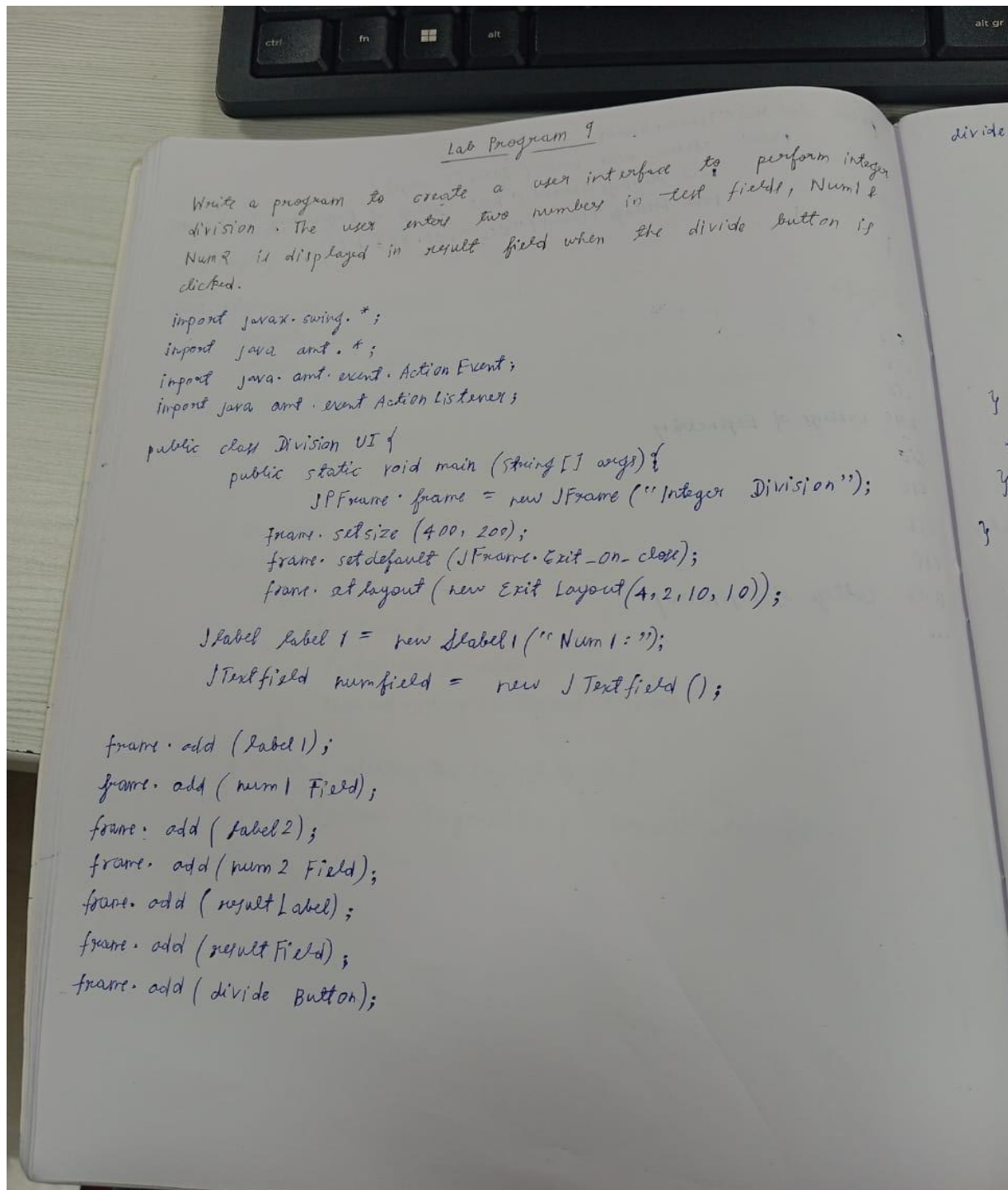
Program 9

SwingDemo

Algorithm:

Lab Program 9

Write a program to create a user interface to perform integer division. The user enters two numbers in test fields, Num1 & Num2 is displayed in result field when the divide button is clicked.

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

public class DivisionUI {
    public static void main(String[] args){
        JPFrame.frame = new JFrame("Integer Division");
        frame.setsize(400, 200);
        frame.setdefault(JFrame.Exit_On_close);
        frame.atlayout(new Exit Layout(4,2,10,10));

        Jlabel label1 = new Jlabel1("Num1:");
        JTextfield numfield = new JTextfield();

        frame.add(label1);
        frame.add(num1 Field);
        frame.add(label2);
        frame.add(num2 Field);
        frame.add(result Label);
        frame.add(result Field);
        frame.add(divide Button);
```

```
divideButton.addActionListener (new ActionListener()){
        try {
            int num1 = Integer.parseInt (num1Field.getText());
            int num2 = Integer.parseInt (num2Field.getText());
        }
        catch ( (Arithmetic Exception ex)
                    J option Plane.showmessageDialogue, frame, setMessage(),
                            "Arithmetic Error", Joption.error_message);
        }
    }

    frame   setVisible (true);
    }
}
```

Output :

Num 1 = 10

Num 2 = 2

5

Num 1 = 10

Num 2 = abc

Please enter valid integers for Num 1 & Num 2

Num 1 = 10

Num 2 = 0

Cannot divide by zero.

Code:

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

class SwingDemo {
    SwingDemo() {

        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 200);
        jfrm.setLayout(new FlowLayout());


        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        JLabel jlab = new JLabel("Enter the divisor and dividend:");


        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);


        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        JLabel nameLabel = new JLabel("TANMAY - 1BM23ME115"); // Your name label

            // Add components in order
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(err);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
        jfrm.add(nameLabel);


        JButton button = new JButton("Calculate");
        jfrm.add(button);


        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
```

```java
            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("");

            } 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!");
            }
         }
      });

      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

Program 10

Demonstrate Inter Process Communication & Deadlock

Algorithm:
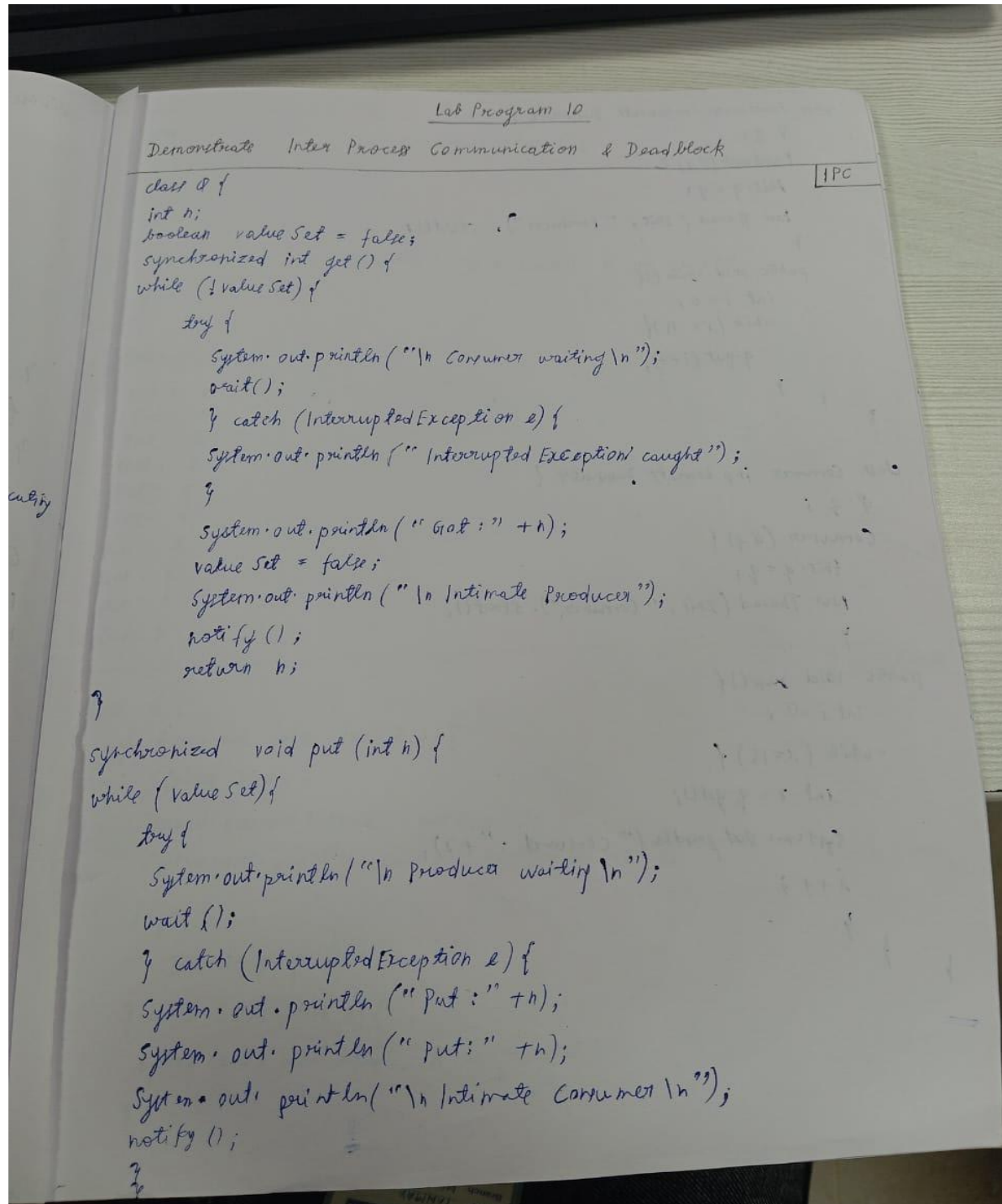
Lab Program 10

Demonstrate Inter Process Communication & Deadlock

```
class Q {
int n;
boolean valueSet = false;
synchronized int get () {
while (!valueSet) {
    try {
        System.out.println ("\n Consumer waiting \n");
        wait ();
    } catch (InterruptedException e) {
        System.out.println (" Interrupted Exception caught");
    }
    System.out.println (" Got :" + n);
    valueSet = false;
    System.out.println ("\n Intimate Producer");
    notify ();
    return n;
}

synchronized void put (int n) {
while (valueSet) {
    try {
        System.out.println ("\n Producer waiting \n");
    wait ();
    } catch (InterruptedException e) {
    System.out.println (" Put :" + n);
    System.out.println (" put:" + n);
    System.out.println ("\n Intimate Consumer \n");
    notify ();
    }
}
```

```java
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 PCFixed {
    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 :

Put : 1
Got : 1
Put : 2
Got : 2
Put : 3
Got : 3

Put : 4
Got : 4
Put : 5
Got : 5

Dead lock

```
class A {
    synchronized void foo (B 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 ");
        }
```

```java
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
    void last() {
        System.out.println(" Inside A.last: ");
    }
}

class B {
    synchronized void bar(A a) {
        String name =
        Thread.currentThread().getName();

        System.out.println(name + " entered B.bar");

        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }

        System.out.println(name + " trying to call A.last()");
        a.last();
    }

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

```java
class Deadlock Implements Runnable {
    A a = new A ();
    B b = new B ();
    Deadlock () {
        Thread.currentThread ().setName ("Main Thread");
        Thread t = new Thread ( this, "Racing Thread");
        t.start ();
        a.foo (b);
        System.out.println (" Back in main thread ");
    }

    public void run () {
        b.bar (a);
        System.out.println (" Back in other thread ");
    }
    public static void main (String args[]) {
        new Deadlock ();
    }
}
```

Output :

Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last ()

Inside A.last
Back in main thread
Racing Thread trying to call A.last ()

Inside A.last B
Back in other thread

Code:

// Deadlock

```
class A {
   synchronized void foo(B b) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered A.foo");
      try {
         Thread.sleep(1000); // Simulating some work
      } catch (Exception e) {
         System.out.println("A Interrupted");
      }
      System.out.println(name + " trying to call B.last()");
      b.last(); // Deadlock occurs here
   }

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

class B {
   synchronized void bar(A a) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered B.bar");
      try {
         Thread.sleep(1000); // Simulating some work
      } catch (Exception e) {
         System.out.println("B Interrupted");
      }
      System.out.println(name + " trying to call A.last()");
      a.last(); // Deadlock occurs here
   }

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

class Deadlock implements Runnable {
   A a = new A();
   B b = new B();

   Deadlock() {
      Thread.currentThread().setName("MainThread");
      Thread t = new Thread(this, "RacingThread");
```

```java
      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[]) {
      System.out.println("TANMAY 1BM23ME115") ;
      new Deadlock();
   }
}


// IPC
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();
```

```java
        } 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);
```
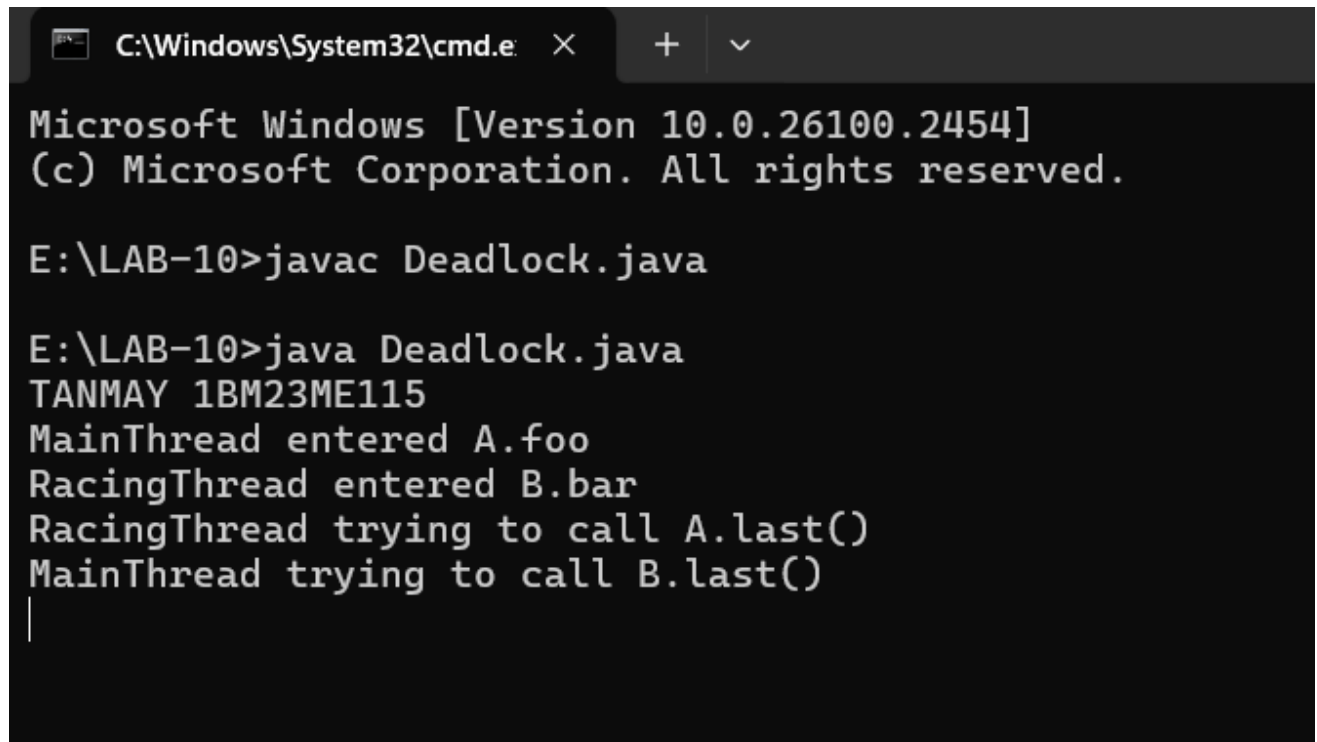
```
        i++;
      }
   }
}

public class PCFixed {
   public static void main(String args[]) {
      System.out.println("TANMAY 1BM23ME115") ;
      Q q = new Q();
      new Producer(q);
      new Consumer(q);
      System.out.println("Press Control-C to stop.");
   }
}
```

```
C:\Windows\System32\cmd.e   ×    +   ∨

Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

E:\LAB-10>javac Deadlock.java

E:\LAB-10>java Deadlock.java
TANMAY 1BM23ME115
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
```

```
Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

E:\LAB-10\IPC>javac PCFixed.java

E:\LAB-10\IPC>java PCFixed.java
TANMAY 1BM23ME115
Put: 0
Press Control-C to stop.

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


Producer waiting

Got: 2

Intimate Producer

Consumed: 2
Put: 3

Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

Consumed: 3
```

69