VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Tanmay Agarwal (1WA23CS010)

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 **Tanmay Agarwal(1WA23CS010),** who is a 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.

Syed Akram Assistant Professor Department of CSE, BMSCE	Dr. Kavita Sooda Professor & HOD Department of CSE, BMSCE

Index

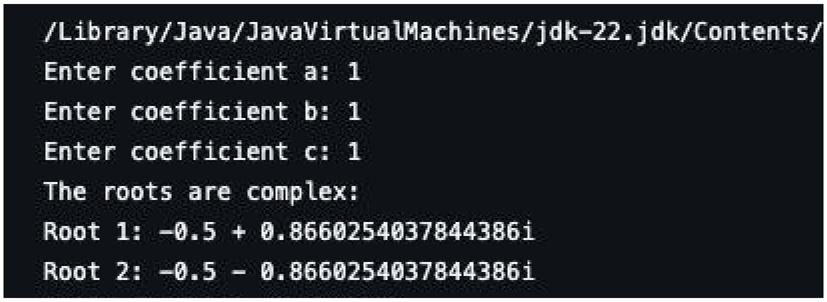
Program 1

Implement Quadratic Equation

<u>Code:</u>

```
import java.util.Scanner;
  //"TAnmay 1wa23cs010”

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();

        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();

        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The roots are real and different:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("The roots are real and the same:");
            System.out.println("Root: " + root);
        } else {
            System.out.println("The roots are complex:");
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");

        }

        scanner.close();
    }
}
```

Output :



```
/Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/
Enter coefficient a: 1
Enter coefficient b: 1
Enter coefficient c: 1
The roots are complex:
Root 1: -0.5 + 0.8660254037844386i
Root 2: -0.5 - 0.8660254037844386i
```

Program 2 :

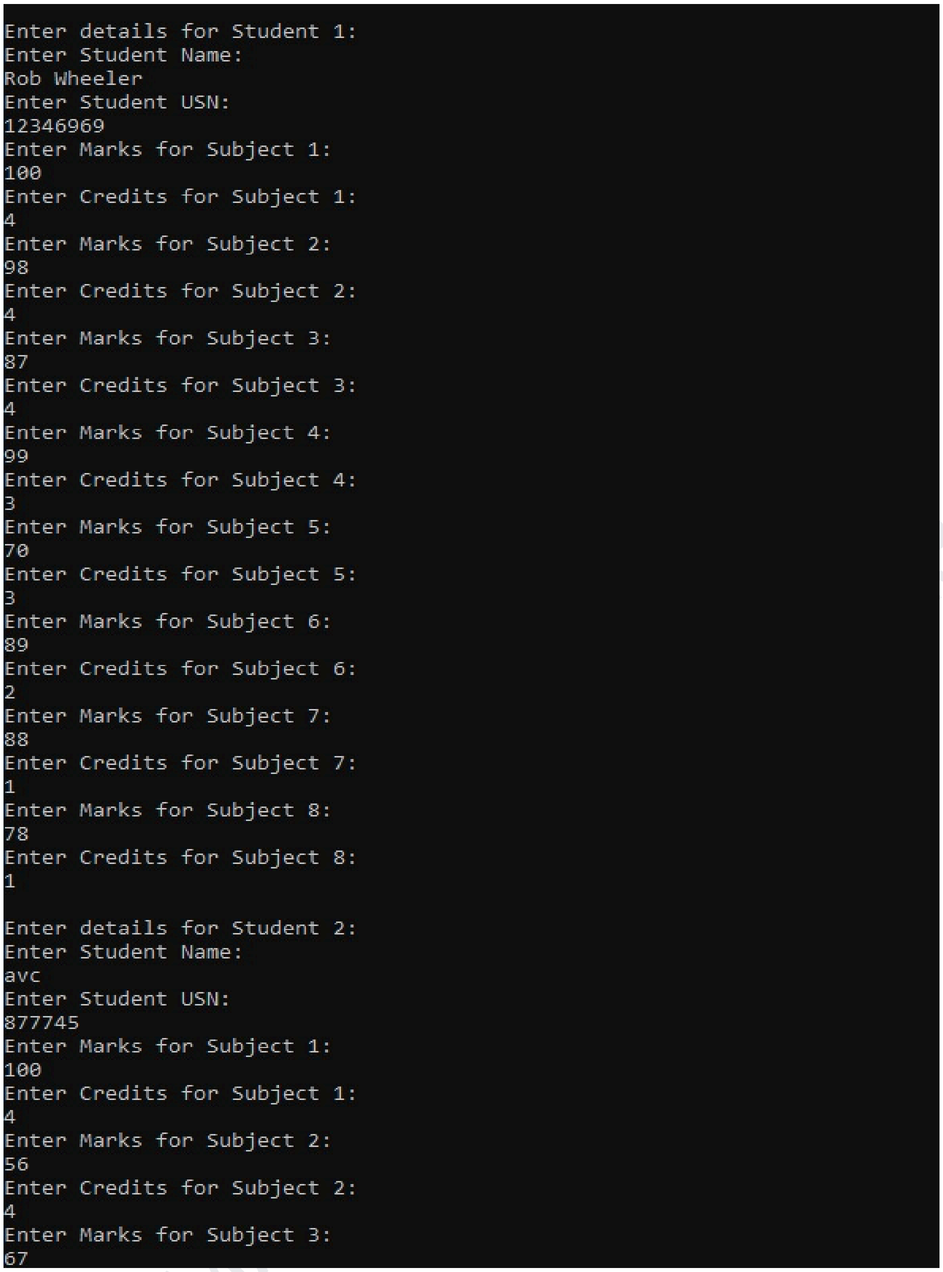
Implement SGPA Calculator

Code :

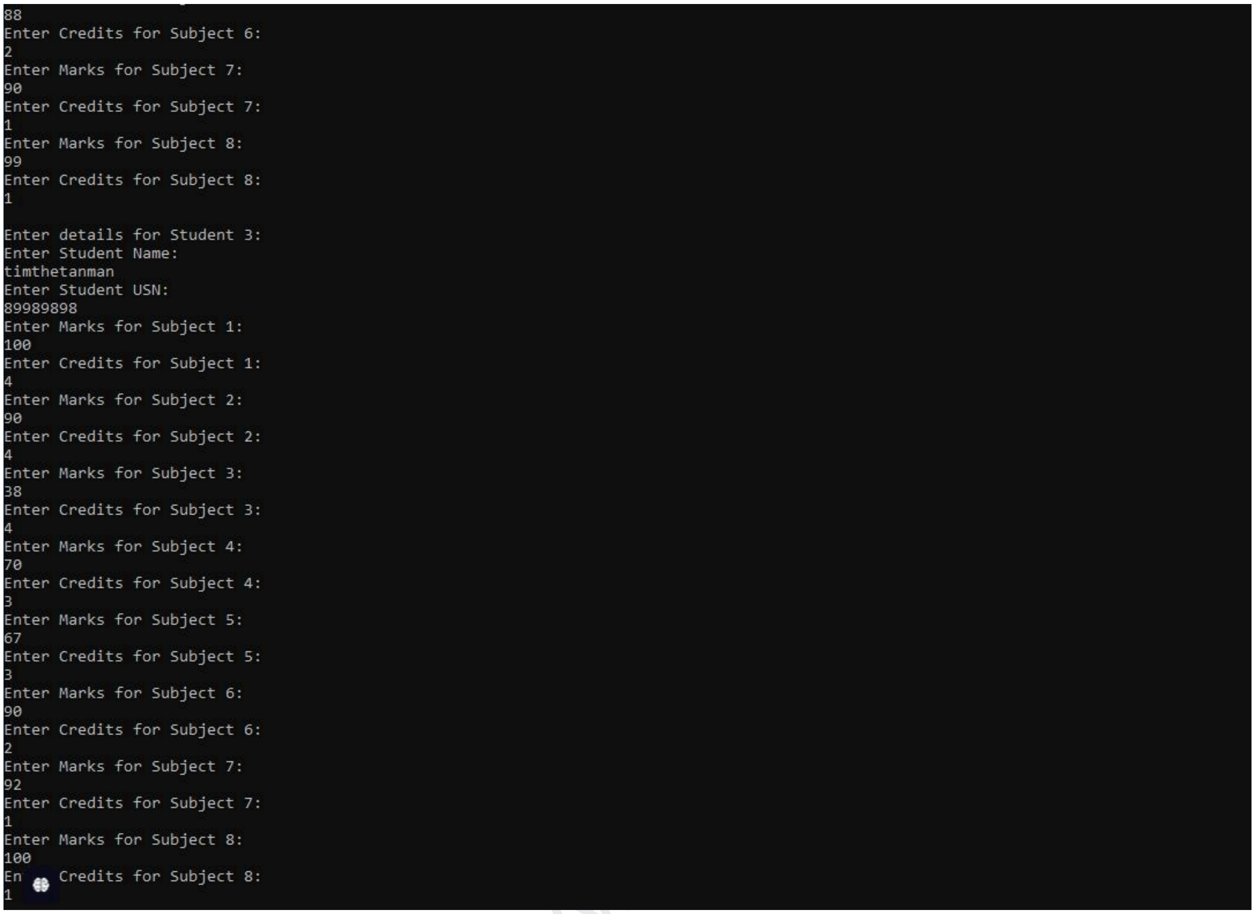
```
import java.util.Scanner;
  //"TAnmay 1wa23cs010”
class Subject {
    int subjectMarks;
    int credits;
    int grade;

    // Method to calculate grade based on marks
    public void calculateGrade() {
        if (subjectMarks >= 90 && subjectMarks <= 100) {
            grade = 10;
        } else if (subjectMarks >= 80) {
            grade = 9;
        } else if (subjectMarks >= 70) {
            grade = 8;
        } else if (subjectMarks >= 60) {
            grade = 7;
        } else if (subjectMarks >= 50) {
            grade = 6;
        } else if (subjectMarks >= 40) {
            grade = 5;
        } else {
            grade = 0;  // Failed the subject
        }
    }
}

class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subject = new Subject[8];
    Scanner s = new Scanner(System.in);


    public Student() {
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject(); // Create subject objects
        }
    }


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


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


            if (subject[i].subjectMarks > 100 || subject[i].subjectMarks < 0) {
                System.out.println("Invalid marks! Please enter again.");
                i--;
                continue;
            }

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

            subject[i].calculateGrade(); // Calculate grade based on marks
        }
    }


    public void computeSGPA() {
        int totalCredits = 0;
        int effectiveScore = 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 displayResult() {
        System.out.println("\nStudent Name: " + name);
        System.out.println("Student USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }
}
```

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




      Student[] students = new Student[3];


      for (int i = 0; i < 3; i++) {
         System.out.println("\nEnter details for Student " + (i + 1) + ": ");
         students[i] = new Student(); // Create a new student object
         students[i].getStudentDetails(); // Get name and usn
         students[i].getMarks(); // Get marks and credits for subjects
         students[i].computeSGPA(); // Compute SGPA
      }


      System.out.println("\n\nResults for all students:");
      for (int i = 0; i < 3; i++) {
         students[i].displayResult();

      }
   }
}
```

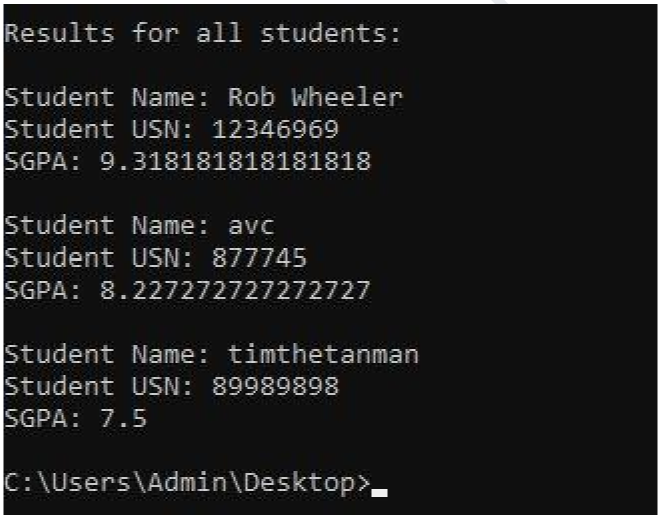
Output :



```
Enter details for Student 1:
Enter Student Name:
Rob Wheeler
Enter Student USN:
12346969
Enter Marks for Subject 1:
100
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
98
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
87
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
99
Enter Credits for Subject 4:
3
Enter Marks for Subject 5:
70
Enter Credits for Subject 5:
3
Enter Marks for Subject 6:
89
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
88
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
78
Enter Credits for Subject 8:
1

Enter details for Student 2:
Enter Student Name:
avc
Enter Student USN:
877745
Enter Marks for Subject 1:
100
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
56
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
67
```



```
88
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
90
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
99
Enter Credits for Subject 8:
1

Enter details for Student 3:
Enter Student Name:
timthetanman
Enter Student USN:
89989898
Enter Marks for Subject 1:
100
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
90
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
38
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
70
Enter Credits for Subject 4:
3
Enter Marks for Subject 5:
67
Enter Credits for Subject 5:
3
Enter Marks for Subject 6:
90
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
92
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
100
En    Credits for Subject 8:
1
```



```
Results for all students:

Student Name: Rob Wheeler
Student USN: 12346969
SGPA: 9.318181818181818

Student Name: avc
Student USN: 877745
SGPA: 8.227272727272727

Student Name: timthetanman
Student USN: 89989898
SGPA: 7.5

C:\Users\Admin\Desktop>
```

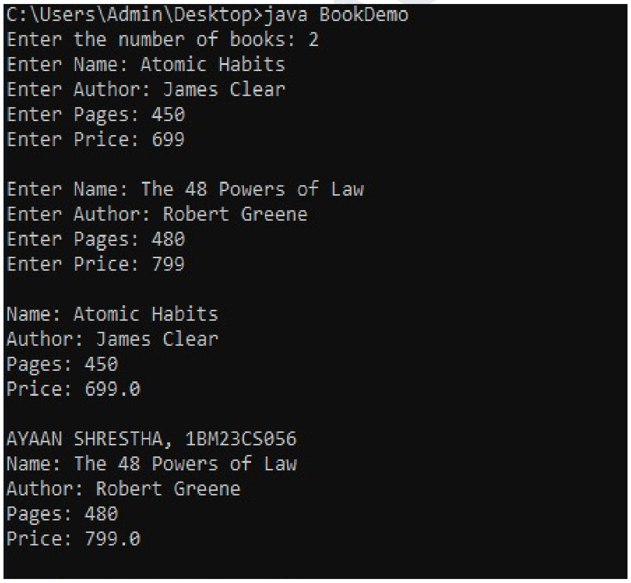
Program 3 :

Create Objects for Books

Code :

```
import java.util.Scanner;
  //"TAnmay 1wa23cs010”

class Book {
    String name, author;
    int num_pages;
    double price;

    void setDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name: ");
        this.name = sc.nextLine();
        System.out.print("Enter Author: ");
        this.author = sc.nextLine();
        System.out.print("Enter Pages: ");
        while (!sc.hasNextInt()) {
            System.out.print("Please enter a valid number for pages: ");
            sc.next();
        }
        this.num_pages = sc.nextInt();
        System.out.print("Enter Price: ");
        while (!sc.hasNextDouble()) {
            System.out.print("Please enter a valid number for price: ");
            sc.next();
        }
        this.price = sc.nextDouble();
        sc.nextLine();     }

    void getDetails() {
        System.out.println(this);
    }

    public String toString() {
            return "Name: " + name + "\nAuthor: " + author + "\nPages: " +
num_pages + "\nPrice: " + price;
    }
}

class BookDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        while (!sc.hasNextInt()) {
            System.out.print("Please enter a valid number: ");
            sc.next();
        }
        int bookNum = sc.nextInt();
        sc.nextLine();

        Book[] bookArray = new Book[bookNum];

        for (int i = 0; i < bookNum; i++) {
            bookArray[i] = new Book();
            bookArray[i].setDetails();
            System.out.println();
        }

        for (int i = 0; i < bookNum; i++) {
            bookArray[i].getDetails();
            System.out.println();

        }


    }
}
```

Output:



```
C:\Users\Admin\Desktop>java BookDemo
Enter the number of books: 2
Enter Name: Atomic Habits
Enter Author: James Clear
Enter Pages: 450
Enter Price: 699

Enter Name: The 48 Powers of Law
Enter Author: Robert Greene
Enter Pages: 480
Enter Price: 799

Name: Atomic Habits
Author: James Clear
Pages: 450
Price: 699.0

AYAAN SHRESTHA, 1BM23CS056
Name: The 48 Powers of Law
Author: Robert Greene
Pages: 480
Price: 799.0
```

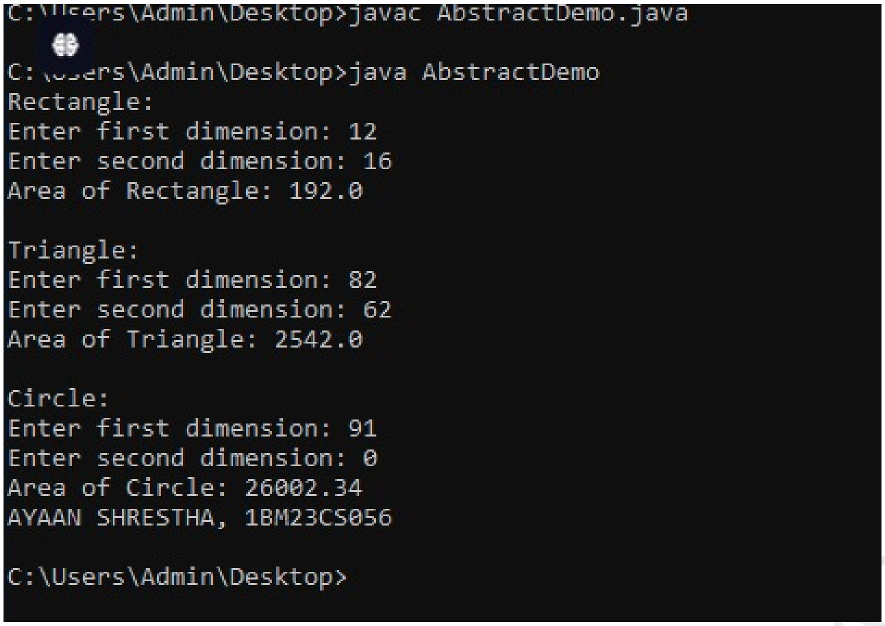
Program 4 :

Implement Abstract Class

Code :

```
import java.util.Scanner;

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

abstract class Shape extends InputScanner {
    double dim1, dim2;

    Shape() {
        System.out.print("Enter first dimension: ");
        this.dim1 = sc.nextDouble();
        System.out.print("Enter second dimension: ");
        this.dim2 = sc.nextDouble();
    }

    abstract double printArea();
}

class Rectangle extends Shape {
    Rectangle() {
        super();
    }

    double printArea() {
        return dim1 * dim2;
    }
  //"TAnmay 1wa23cs010”
}

class Triangle extends Shape {
    Triangle() {
        super();
    }

    double printArea() {
        return 0.5 * dim1 * dim2;
    }
}

class Circle extends Shape {
    Circle() {
        super();
        this.dim2 = 0;      }

    double printArea() {
        return 3.14 * dim1 * dim1;
    }
}

class AbstractDemo {
    public static void main(String[] args) {
        System.out.println("Rectangle:");
        Shape figref = new Rectangle();
        System.out.println("Area of Rectangle: " + figref.printArea());

        System.out.println("\nTriangle:");
        figref = new Triangle();
        System.out.println("Area of Triangle: " + figref.printArea());

        System.out.println("\nCircle:");
        figref = new Circle();
        System.out.println("Area of Circle: " + figref.printArea());
                //"TAnmay 1wa23cs010"
    }
}
```

Output:



```
C:\Users\Admin\Desktop>javac AbstractDemo.java

C:\Users\Admin\Desktop>java AbstractDemo
Rectangle:
Enter first dimension: 12
Enter second dimension: 16
Area of Rectangle: 192.0

Triangle:
Enter first dimension: 82
Enter second dimension: 62
Area of Triangle: 2542.0

Circle:
Enter first dimension: 91
Enter second dimension: 0
Area of Circle: 26002.34
AYAAN SHRESTHA, 1BM23CS056

C:\Users\Admin\Desktop>
```

Program 5 :

Bank Account Management

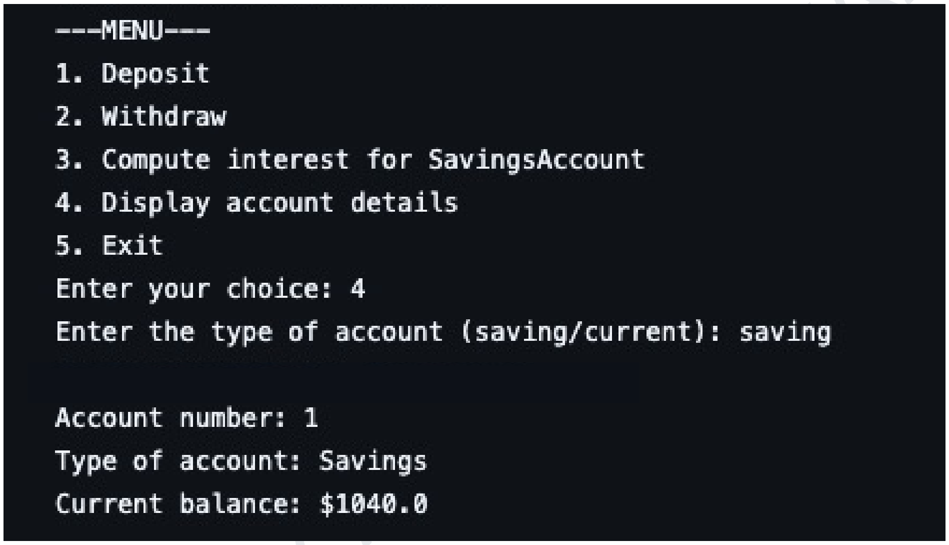
Code :

```
import java.util.Scanner;

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

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

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. New balance: $" + balance);
    }
    void displayBalance() {
        System.out.println("Current balance: $" + balance);
    }
}
  //"TAnmay 1wa23cs010"

class SavingsAccount extends Account {
    private static final double INTEREST_RATE = 0.04;

        SavingsAccount(String customerName, int accountNumber, double
initialBalance) {
        super(customerName, accountNumber, "Savings", initialBalance);
    }

    void computeInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest added. New balance: $" + balance);
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds.");
        } else {
            balance -= amount;
                System.out.println("Withdrawal successful. New balance: $" +
balance);
        }
    }
}

class CurrentAccount extends Account {
    private static final double MIN_BALANCE = 500.00;
    private static final double SERVICE_CHARGE = 25.00;

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

    void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
             System.out.println("Below minimum balance. Service charge of $"
+ SERVICE_CHARGE + " applied. New balance: $" + balance);
        } else {
            System.out.println("Balance is above the minimum required.");
        }
    }

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

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


           SavingsAccount savingsAccount = new SavingsAccount("Tanmay",
1, 1000.00);
         CurrentAccount currentAccount = new CurrentAccount("Tanmay", 2,
800.00);

        while (true) {
            System.out.println("\n---MENU---");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute interest for SavingsAccount");
            System.out.println("4. Display account details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            Account selectedAccount = null;
            System.out.print("Enter the type of account (saving/current): ");
            String accountType = sc.next();

            if (accountType.equalsIgnoreCase("saving")) {
                selectedAccount = savingsAccount;
            } else if (accountType.equalsIgnoreCase("current")) {
                selectedAccount = currentAccount;
            } else {
                System.out.println("Invalid account type. Try again.");
                continue;
            }
```

```
 //"TAnmay 1wa23cs010”

          switch (choice) {
             case 1:
                System.out.print("Enter the deposit amount: ");
                double depositAmount = sc.nextDouble();
                selectedAccount.deposit(depositAmount);
                break;
             case 2:
                System.out.print("Enter the withdrawal amount: ");
                double withdrawalAmount = sc.nextDouble();
                if (selectedAccount instanceof SavingsAccount) {
                                                              ((SavingsAccount)
selectedAccount).withdraw(withdrawalAmount);
                } else if (selectedAccount instanceof CurrentAccount) {
                                                              ((CurrentAccount)
selectedAccount).withdraw(withdrawalAmount);
                }
                break;
             case 3:
                if (selectedAccount instanceof SavingsAccount) {
                   ((SavingsAccount) selectedAccount).computeInterest();
                } else {
                    System.out.println("Interest computation is only for savings
accounts.");
                }
                break;
             case 4:
                                    System.out.println("Account  name:  "  +
selectedAccount.customerName);
                                 System.out.println("Account  number:  "  +
selectedAccount.accountNumber);
                                 System.out.println("Type  of  account:  "  +
selectedAccount.accountType);
                selectedAccount.displayBalance();
                break;
             case 5:
                System.out.println("Exiting program...");
                sc.close();
                return;
             default:
                System.out.println("Invalid choice. Try again.");

          }
       }
   }
```

```
}
```

Output:



```
---MENU---
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 4
Enter the type of account (saving/current): saving

Account number: 1
Type of account: Savings
Current balance: $1040.0
```

Program No: 6

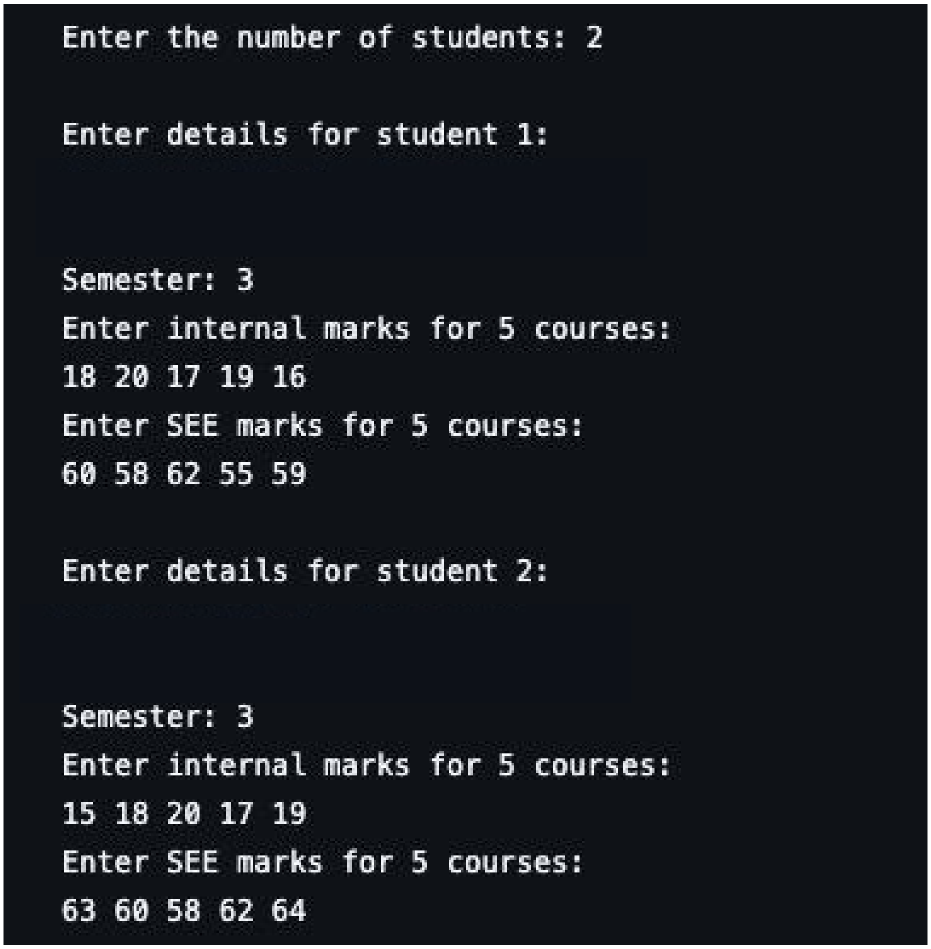
Implement Packages

Code :

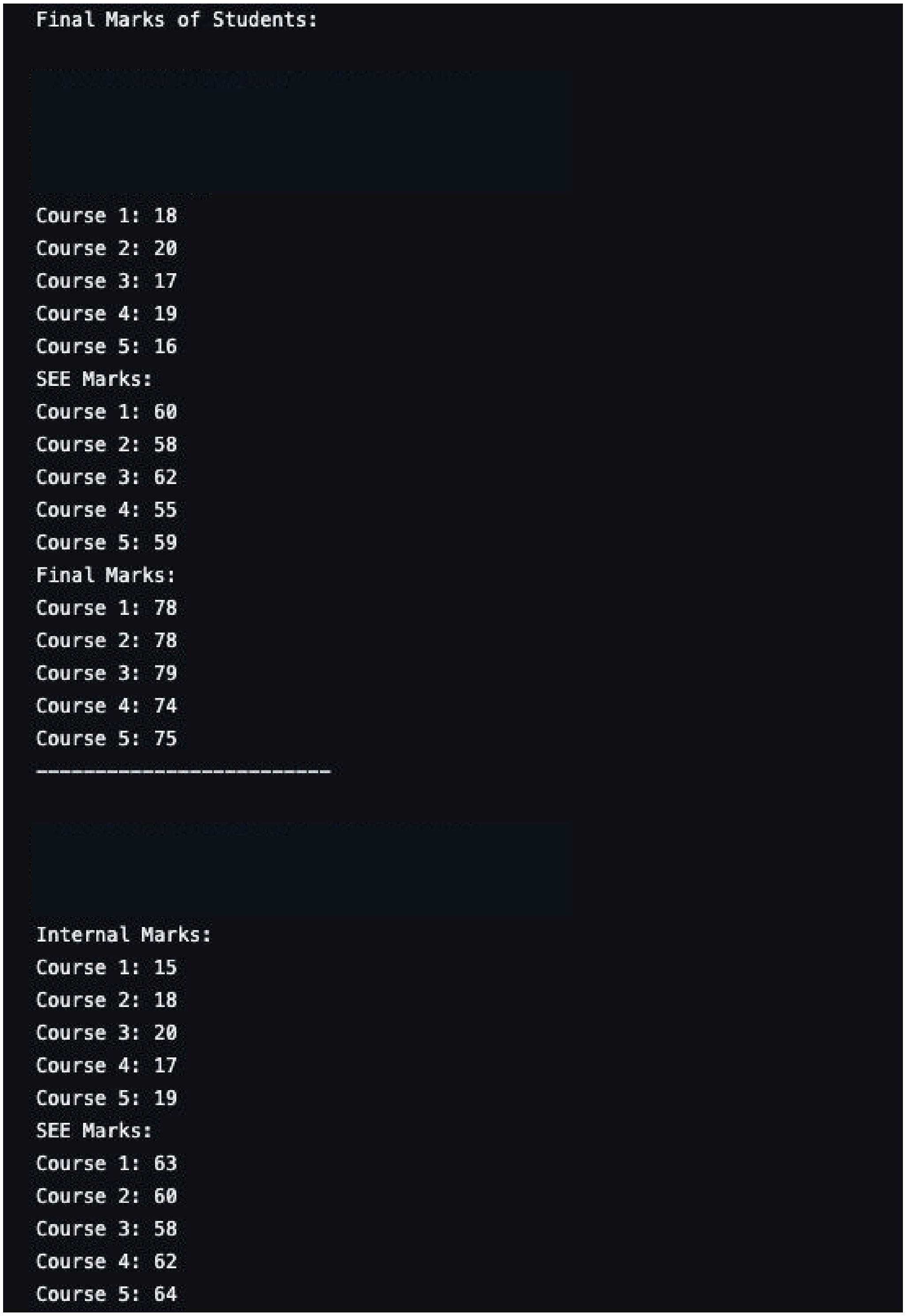
```
package SEE;

import CIE.Student;

public class External extends Student {
    public int[] externalMarks;


    public External(String usn, String name, int sem, int[] externalMarks) {
        super(usn, name, sem);
        this.externalMarks = externalMarks;
    }


    public void printExternalMarks() {
        System.out.println("External Marks for " + name + " (" + usn + "):");
        for (int i = 0; i < externalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
        }
    }
}


package CIE;

public class Internals extends Student {
    public int[] internalMarks;


    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }


    public void printInternalMarks() {
        System.out.println("Internal Marks for " + name + " (" + usn + "):");
        for (int i = 0; i < internalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + internalMarks[i]);
        }
    }
}
```

```
import CIE.Internals;
import SEE.External;

import java.util.Scanner;
  //"TAnmay 1wa23cs010”

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();

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

            System.out.println("\nEnter details for Student " + (i + 1) + ":");

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

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

            System.out.print("Enter Semester: ");
            int sem = scanner.nextInt();


            int[] internalMarks = new int[5];
            System.out.println("Enter Internal Marks for 5 Courses:");
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                internalMarks[j] = scanner.nextInt();
            }


            int[] externalMarks = new int[5];
            System.out.println("Enter External Marks for 5 Courses:");
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                externalMarks[j] = scanner.nextInt();
            }
            scanner.nextLine();


            Internals internalStudent = new Internals(usn, name, sem, internalMarks);
            External externalStudent = new External(usn, name, sem, externalMarks);


            internalStudent.printInternalMarks();
            externalStudent.printExternalMarks();


            printFinalMarks(internalStudent, externalStudent);
        }

        scanner.close();
    }


    public static void printFinalMarks(Internals internal, External external) {
        int[] internalMarks = internal.internalMarks;
        int[] externalMarks = external.externalMarks;
        int totalMarks;

            System.out.println("Final Marks for " + internal.getName() + " (" + internal.getUsn() +
"):");
        for (int i = 0; i < internalMarks.length; i++) {
            totalMarks = internalMarks[i] + externalMarks[i];
            System.out.println("Course " + (i + 1) + ": " + totalMarks);
        }
    }
}

package CIE;

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


    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
    // Getter methods
    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {
        return sem;
  //"TAnmay 1wa23cs010”
    }
}
```

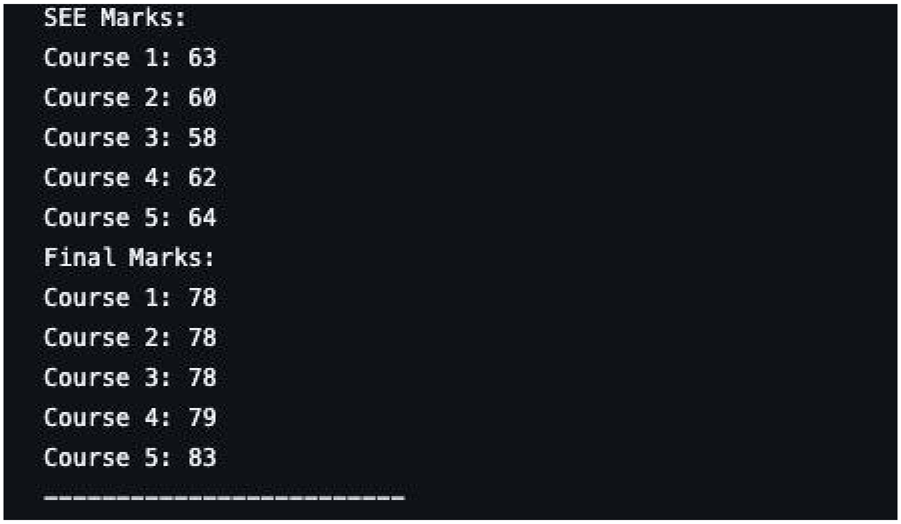
Output :



```
Enter the number of students: 2

Enter details for student 1:
[redacted]

Semester: 3
Enter internal marks for 5 courses:
18 20 17 19 16
Enter SEE marks for 5 courses:
60 58 62 55 59

Enter details for student 2:
[redacted]

Semester: 3
Enter internal marks for 5 courses:
15 18 20 17 19
Enter SEE marks for 5 courses:
63 60 58 62 64
```



```
Final Marks of Students:

[redacted]

Course 1: 18
Course 2: 20
Course 3: 17
Course 4: 19
Course 5: 16
SEE Marks:
Course 1: 60
Course 2: 58
Course 3: 62
Course 4: 55
Course 5: 59
Final Marks:
Course 1: 78
Course 2: 78
Course 3: 79
Course 4: 74
Course 5: 75
-------------------------

[redacted]

Internal Marks:
Course 1: 15
Course 2: 18
Course 3: 20
Course 4: 17
Course 5: 19
SEE Marks:
Course 1: 63
Course 2: 60
Course 3: 58
Course 4: 62
Course 5: 64
```



```
SEE Marks:
Course 1: 63
Course 2: 60
Course 3: 58
Course 4: 62
Course 5: 64
Final Marks:
Course 1: 78
Course 2: 78
Course 3: 78
Course 4: 79
Course 5: 83
-------------------------
```

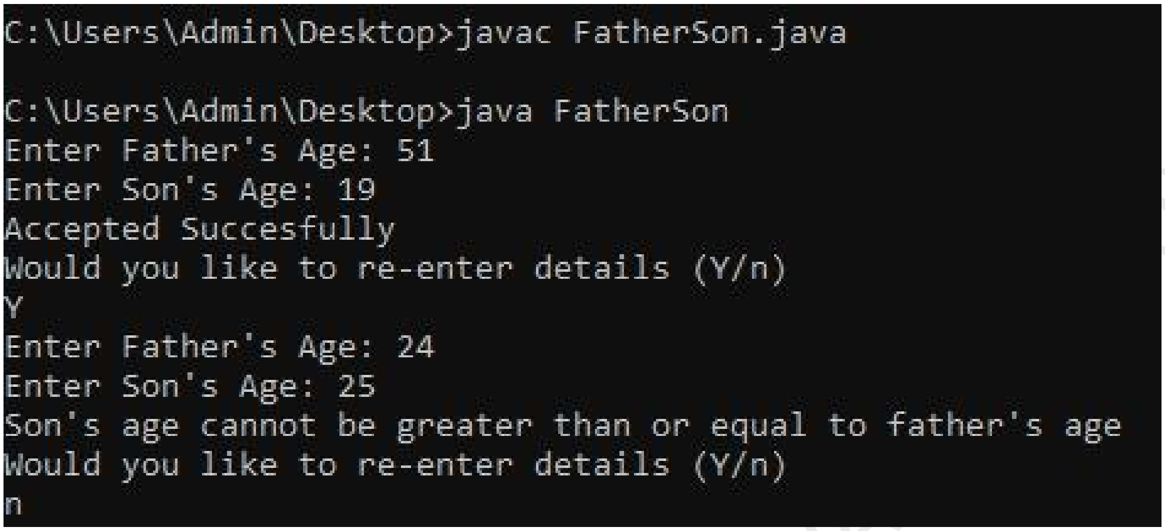
Program No : 7

Implement Exception Handling

Code :

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
  //"TAnmay 1wa23cs010"
class SonAgeException extends Exception {
    public SonAgeException(String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
     public Son(int fatherAge, int sonAge) throws WrongAgeException,
SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
                throw new SonAgeException("Son's age cannot be greater than or
equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class FatherSon{
    public static void main(String[] args) {
        while(true){
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Successfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re-enter details (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}
```

Output:



```
C:\Users\Admin\Desktop>javac FatherSon.java

C:\Users\Admin\Desktop>java FatherSon
Enter Father's Age: 51
Enter Son's Age: 19
Accepted Succesfully
Would you like to re-enter details (Y/n)
Y
Enter Father's Age: 24
Enter Son's Age: 25
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
n
```

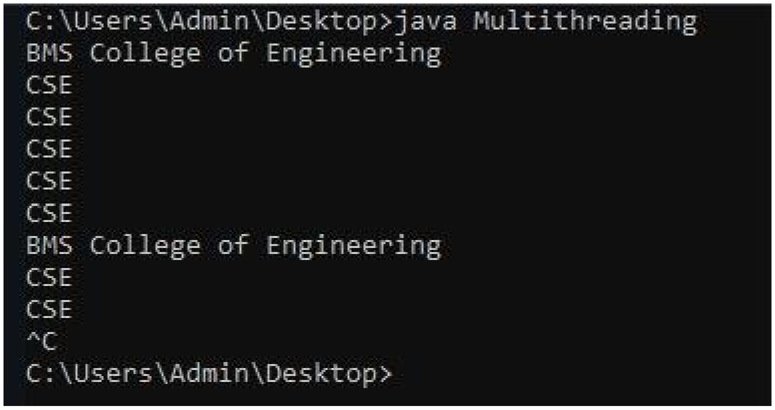
Program No : 8

Multithreading, Creating Threads in Java

Code :

```
class BMS extends Thread {
   public void run() {
      try {
         while (true) {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000); // Sleep for 10 seconds
         }
      }catch (InterruptedException e) {}
   }
}
class CSE extends Thread {
   public void run() {
      try {
         while (true) {
            System.out.println("CSE");
            Thread.sleep(2000); // Sleep for 2 seconds
         }  //"TAnmay 1wa23cs010”
      }catch (InterruptedException e) {}
   }
}

public class Multithreading{
   public static void main(String[] args) {
      BMS bms = new BMS();
      CSE cse = new CSE();
      bms.start();
      cse.start();
   }
}
```

Output:



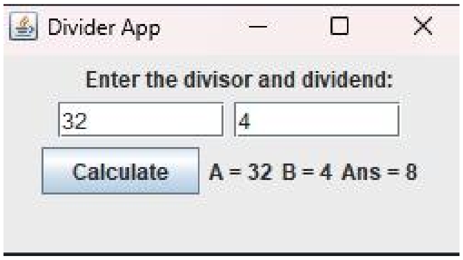
```
C:\Users\Admin\Desktop>java Multithreading
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
^C
C:\Users\Admin\Desktop>
```

Program No : 9

Interface to Perform Integer Division

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

class SwingDemo {
   SwingDemo() {
      //Tanmay1wa23cs010
      // Creating the Frame using AWT
      Frame frame = new Frame("Divider App");
      frame.setSize(275, 150);//Tanmay1wa23cs010
      frame.setLayout(new FlowLayout());

      frame.setVisible(true);

      // Creating components using AWT//Tanmay1wa23cs010
      Label jlab = new Label("Enter the divisor and dividend:");

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

      Button button = new Button("Calculate");

      Label err = new Label();
      Label alab = new Label();
      Label blab = new Label();
      Label anslab = new Label();

      frame.add(err);
      frame.add(jlab);
      frame.add(ajtf);
      frame.add(bjtf);
      frame.add(button);
      frame.add(alab);   //Tanmay1wa23cs010
      frame.add(blab);
      frame.add(anslab);   //Tanmay1wa23cs010

      button.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent evt) {
            try {
               //Tanmay1wa23cs010
               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!");
            }
         }
      });

      // Setting window close behavior
      frame.addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent we) {
            System.exit(0);
         }
      });
   }

   public static void main(String args[]) {
      // Initialize the GUI
      SwingDemo demo = new SwingDemo();
   }
}
```

Output :



Enter the divisor and dividend:

```
32    4
A = 32  B = 4  Ans = 8
```

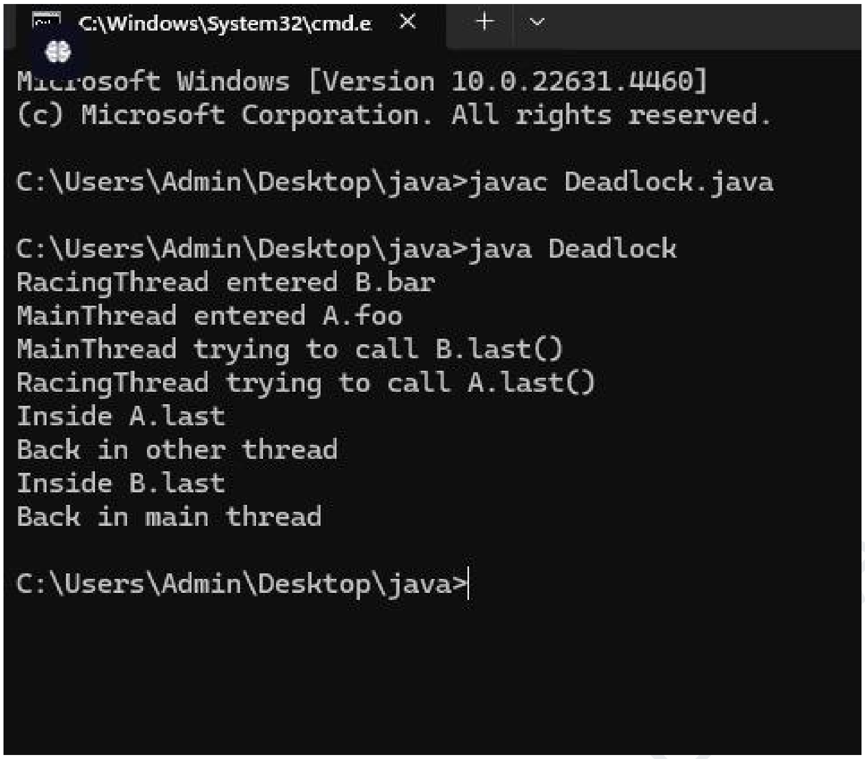
Program 10.1

Implement Deadlock

Code :

```
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");
        }


        synchronized (b) {
            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");
        }


        synchronized (a) {
            System.out.println(name + " trying to call A.last()");
            a.last();
        }
    }
    void last() {
        System.out.println("Inside B.last");  //"TAnmay 1wa23cs010”
    }
}

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

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



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\java>javac Deadlock.java

C:\Users\Admin\Desktop\java>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in other thread
Inside B.last
Back in main thread

C:\Users\Admin\Desktop\java>
```

Program 10.2 :

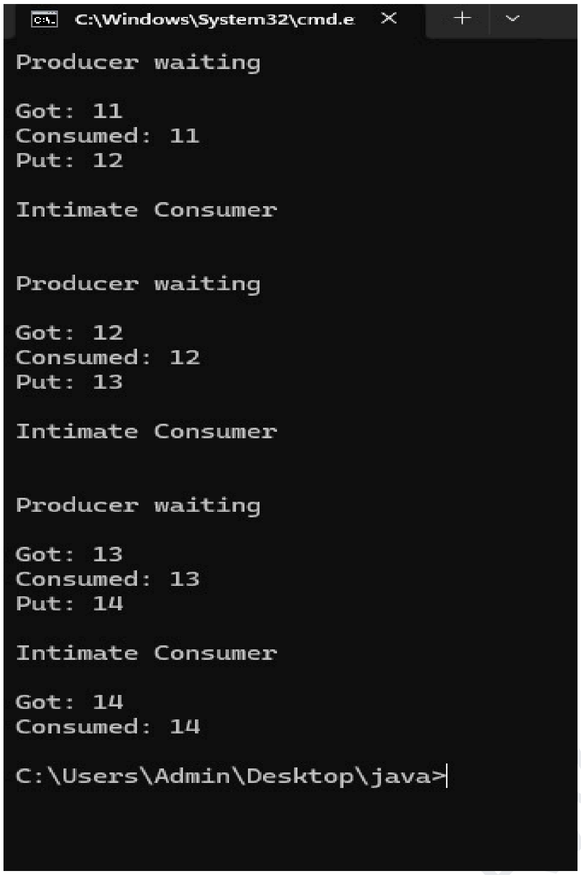
Implement Inter-process Communication

Code :

```
//Tanmay1wa23cs010

class Money {
   int amt = 0;
   boolean accountEmpty = true;

   public synchronized void putMoney() {
      while (!accountEmpty) {
         try {
            wait(); // Wait if the account is not empty
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught in putMoney");
         }
      }//Tanmay1wa23cs010
      amt++; // Increment the amount
      System.out.println("Put: " + amt);
      accountEmpty = false; // Set account to not empty
      notify(); // Notify waiting threads
   }

   public synchronized void getMoney() {
      while (accountEmpty) {
         try {
            wait(); // Wait if the account is empty
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught in getMoney");
         }
      }
      System.out.println("Get: " + amt);
      accountEmpty = true; // Set account to empty
      notify(); // Notify waiting threads
   }
} //Tanmay1wa23cs010

class Put implements Runnable {
   Money m;

   public Put(Money m) {
      this.m = m;
   }
```

```
    public void run() {
        for (int i = 0; i < 5; i++) {
            m.putMoney(); // Put money into the account
            try {
                Thread.sleep(500); // Sleep to simulate delay
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught in Put");
            }
        }
    }
} //Tanmay1wa23cs010

class Get implements Runnable {
    Money m;

    public Get(Money m) {
        this.m = m;
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            m.getMoney(); // Get money from the account
            try {
                Thread.sleep(500); // Sleep to simulate delay
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught in Get");
            }
        }
    }
}

public class MultiThreading {
    public static void main(String[] args) {
        Money m = new Money();
        Put p = new Put(m);
        Get g = new Get(m);

        Thread t1 = new Thread(p);
        Thread t2 = new Thread(g);

        t1.start();
        t2.start();
    }
}
//Tanmay1wa23cs010
```

OutPut:



```
Producer waiting

Got: 11
Consumed: 11
Put: 12

Intimate Consumer


Producer waiting

Got: 12
Consumed: 12
Put: 13

Intimate Consumer


Producer waiting

Got: 13
Consumed: 13
Put: 14

Intimate Consumer

Got: 14
Consumed: 14

C:\Users\Admin\Desktop\java>
```