VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Shrey Pandarathil (1BF24CS285)

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 **Shrey Pandarathil(1BF24CS285),** who is 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/shreypandarathilcs24/JAVA

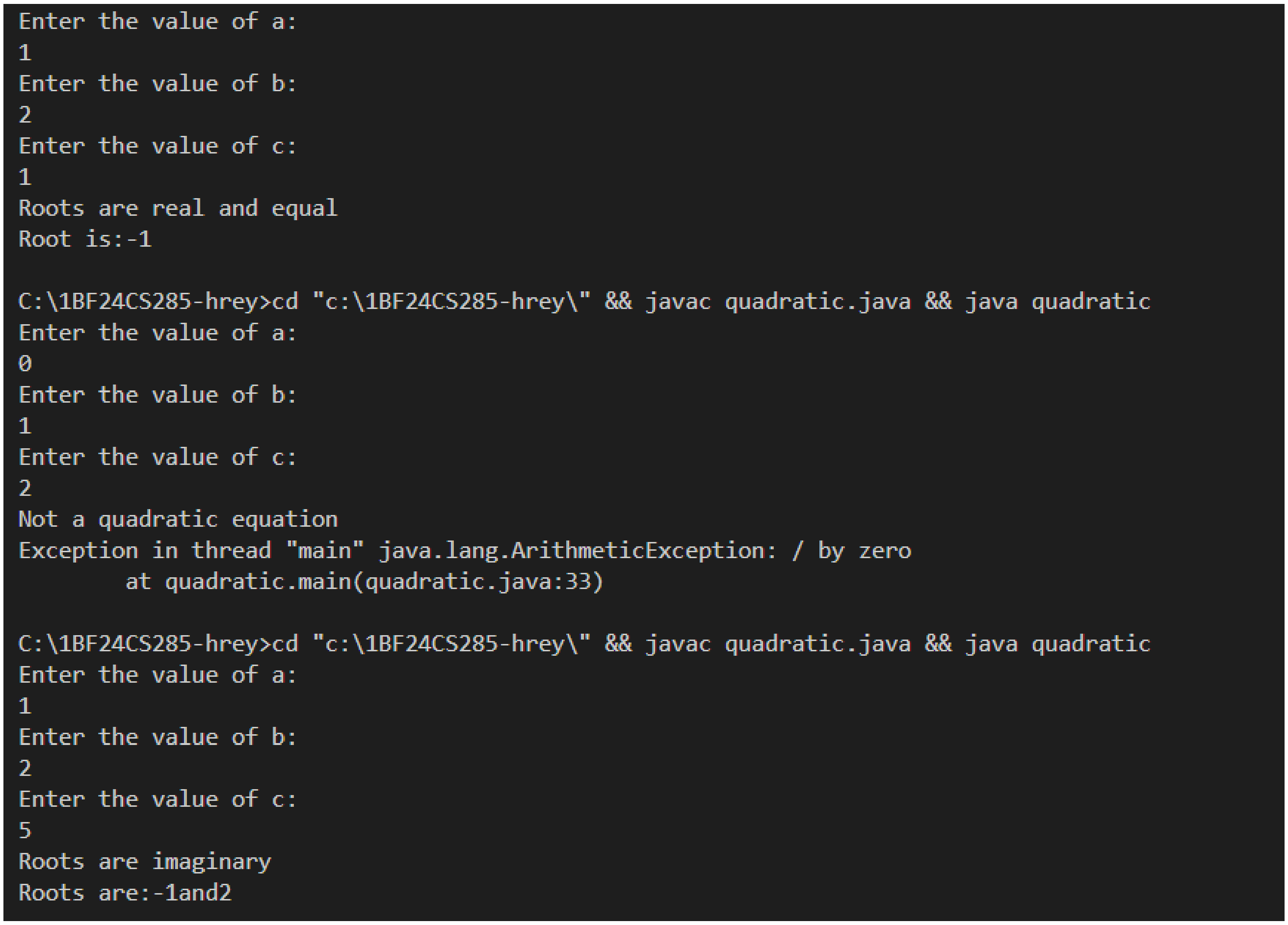
Program 1

Implement Quadratic Equation
Code:

```
import java.util.Scanner;

class quadratic
{
   public static void main (String args[])
   {
      int a,b,c,d,r1,r2;
      Scanner in = new Scanner(System.in);

      System.out.println("Enter the value of a:");
      a=in.nextInt();

      System.out.println("Enter the value of b:");
      b=in.nextInt();

      System.out.println("Enter the value of c:");
      c=in.nextInt();

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

      if (a==0)
      {
         System.out.println("Not a quadratic equation");
      }
      if (d==0)
      {
         r1=(-b)/(2*a);
         System.out.println("Roots are real and equal");
         System.out.println("Root is:" +r1);
      }
      else if (d>0)
      {
         r1 = ((-b) + (int) Math.pow(d, 1/2))/(2*a);
         r2 = ((-b) + (int) Math.pow(d, 1/2))/(2*a);
         System.out.println("Roots are real and distinct");
         System.out.println("Roosts are:" +r1+ "and" +r2);
      }
      else if (d<0)
      {
         System.out.println("Roots are imaginary");
         r1 = (-b)/(2*a);
```

```
            r2 = (int)Math.sqrt(-d)/(2*a);
            System.out.println("Roots are:" +r1+ "and" +r2);
        }
        else
        {
            System.out.println("Invalid Number");
        }
    }
}
```

Output:



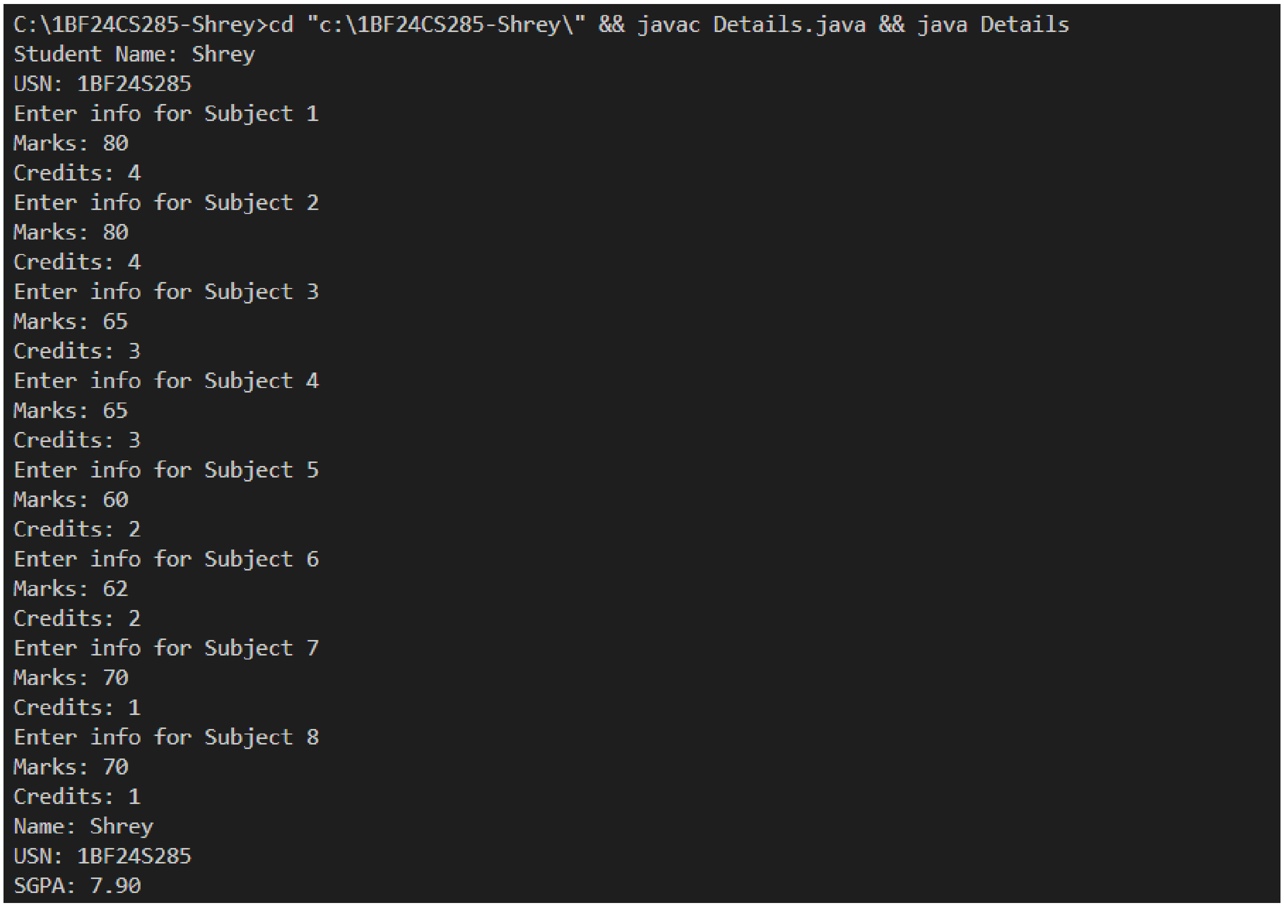
```
Enter the value of a:
1
Enter the value of b:
2
Enter the value of c:
1
Roots are real and equal
Root is:-1

C:\1BF24CS285-hrey>cd "c:\1BF24CS285-hrey\" && javac quadratic.java && java quadratic
Enter the value of a:
0
Enter the value of b:
1
Enter the value of c:
2
Not a quadratic equation
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at quadratic.main(quadratic.java:33)

C:\1BF24CS285-hrey>cd "c:\1BF24CS285-hrey\" && javac quadratic.java && java quadratic
Enter the value of a:
1
Enter the value of b:
2
Enter the value of c:
5
Roots are imaginary
Roots are:-1and2
```

Program 2: SGPA Calculator

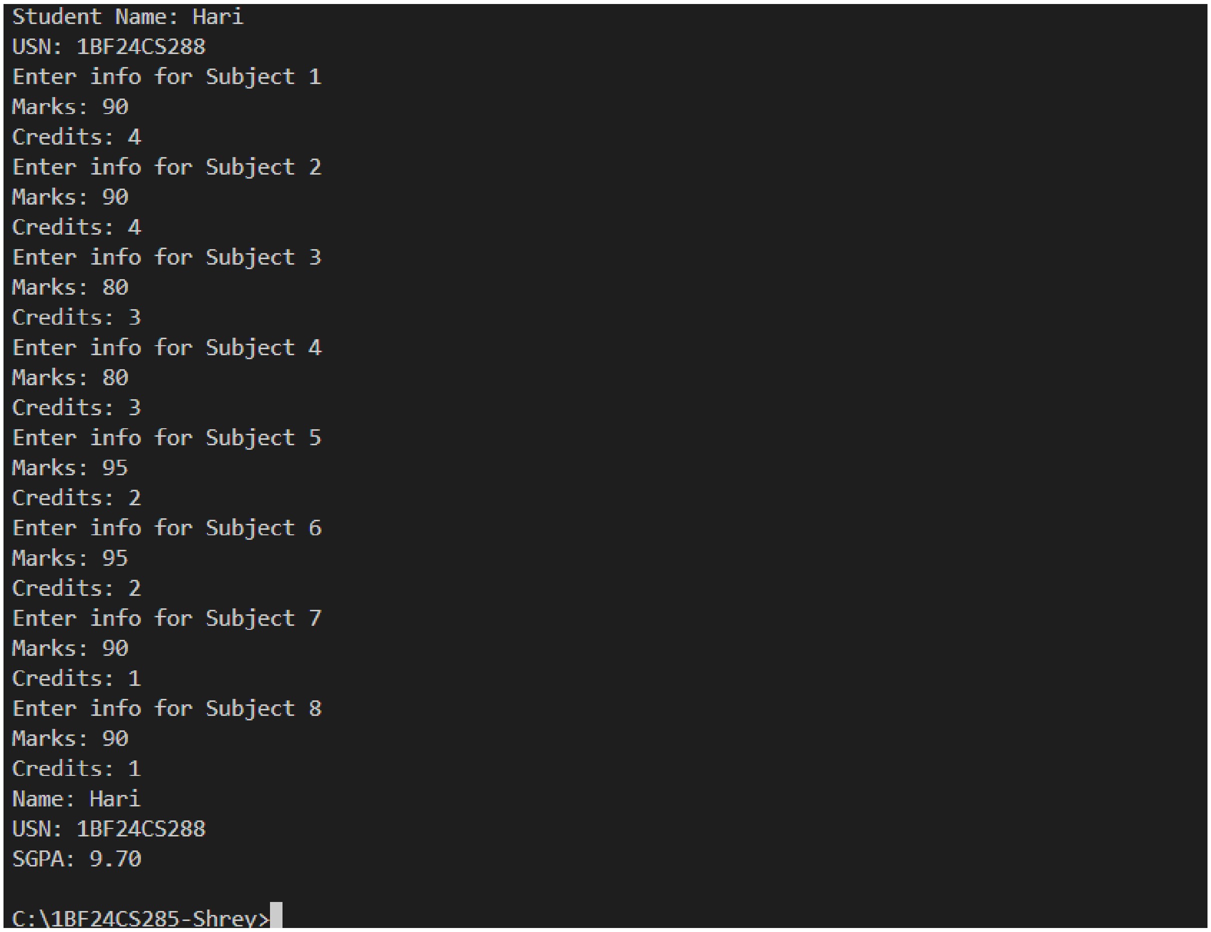
Code:

```
import java.util.Scanner;

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

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

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

    // Method to get student's name and USN
    void StudentDetails() {
        System.out.print("Student Name: ");
        name = s.nextLine();
        System.out.print("USN: ");
        usn = s.nextLine();
    }

    // Method to read marks and credits and calculate grades
    void Marks() {
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter info for Subject " + (i + 1));

            // Input marks
            System.out.print("Marks: ");
            subject[i].subjectMarks = s.nextInt();
            while(subject[i].subjectMarks < 0 || subject[i].subjectMarks > 100) {
                System.out.println("Invalid marks, Enter between 0-100");
                System.out.print("Marks: ");
                subject[i].subjectMarks = s.nextInt();
            }

            // Input credits
            System.out.print("Credits: ");
            subject[i].credits = s.nextInt();
            while(subject[i].credits <= 0) {
                System.out.println("Please enter positive credits.");
                System.out.print("Credits: ");
                subject[i].credits = s.nextInt();
            }

            // Calculate grade based on marks
            subject[i].grade = (subject[i].subjectMarks / 10) + 1;

            if (subject[i].grade == 11)
                subject[i].grade = 10;

            if (subject[i].subjectMarks < 40)
                subject[i].grade = 0;
        }
    }

    // Calculate SGPA
    void SGPA() {
        int Score = 0;
        int totalCredits = 0;

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

        SGPA = (double) Score / (double) totalCredits;
    }

    // Display student details and SGPA
    void display() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.printf("SGPA: %.2f\n", SGPA);
    }
}

public class Details {
    public static void main(String[] args) {
        Student student[] = new Student[2];
        for (int i=0;i<2;i++)
        {
          student[i]=new Student();
          student[i].StudentDetails();
          student[i].Marks();
          student[i].SGPA();
          student[i].display();
        }

    }
}
```

Output:



```
C:\1BF24CS285-Shrey>cd "c:\1BF24CS285-Shrey\" && javac Details.java && java Details
Student Name: Shrey
USN: 1BF24S285
Enter info for Subject 1
Marks: 80
Credits: 4
Enter info for Subject 2
Marks: 80
Credits: 4
Enter info for Subject 3
Marks: 65
Credits: 3
Enter info for Subject 4
Marks: 65
Credits: 3
Enter info for Subject 5
Marks: 60
Credits: 2
Enter info for Subject 6
Marks: 62
Credits: 2
Enter info for Subject 7
Marks: 70
Credits: 1
Enter info for Subject 8
Marks: 70
Credits: 1
Name: Shrey
USN: 1BF24S285
SGPA: 7.90
```



```
Student Name: Hari
USN: 1BF24CS288
Enter info for Subject 1
Marks: 90
Credits: 4
Enter info for Subject 2
Marks: 90
Credits: 4
Enter info for Subject 3
Marks: 80
Credits: 3
Enter info for Subject 4
Marks: 80
Credits: 3
Enter info for Subject 5
Marks: 95
Credits: 2
Enter info for Subject 6
Marks: 95
Credits: 2
Enter info for Subject 7
Marks: 90
Credits: 1
Enter info for Subject 8
Marks: 90
Credits: 1
Name: Hari
USN: 1BF24CS288
SGPA: 9.70

C:\1BF24CS285-Shrey>
```

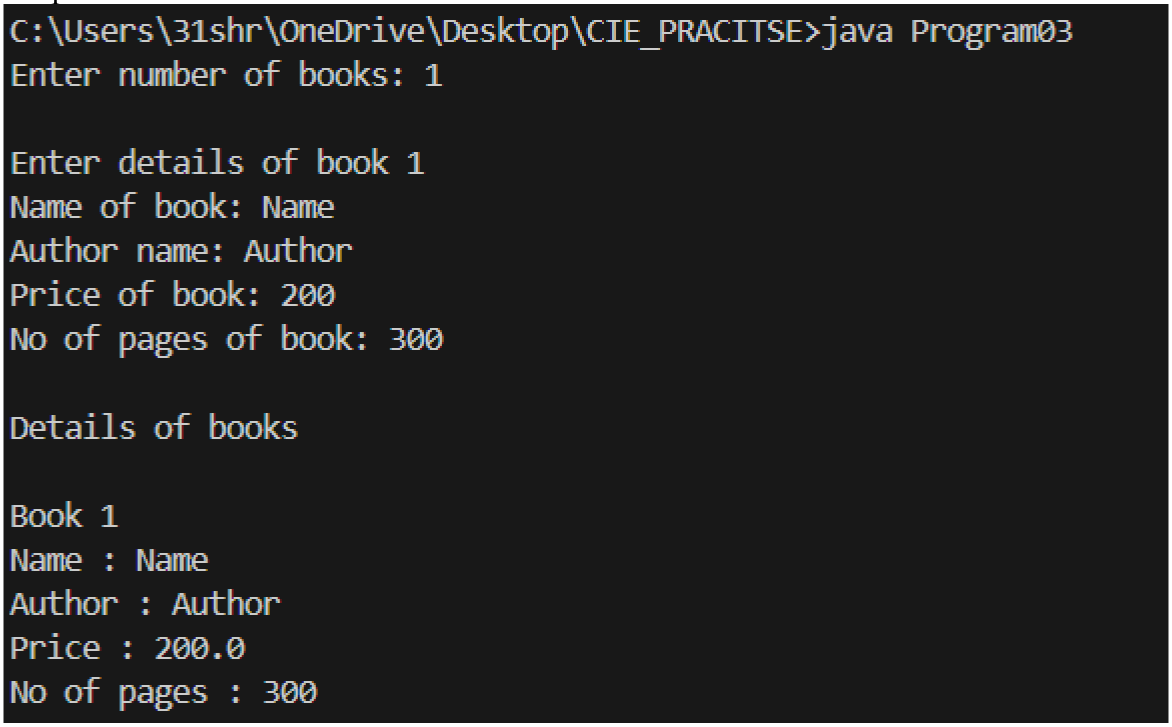
Program 3: Bookstore Program

Code:

```
import java.util.Scanner;

class Book {
   String name;
   String author;
   double price;
   int numpages;

   // Parameterized constructor
   Book(String name, String author, double price, int numpages) {
      this.name = name;
      this.author = author;
      this.price = price;
      this.numpages = numpages;
   }

   // Overriding toString() method
   public String toString() {
      return "Name : " + name + "\n"
          + "Author : " + author + "\n"
          + "Price : " + price + "\n"
          + "No of pages : " + numpages + "\n";
   }
}

public class Program03 {
   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

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

      Book b[] = new Book[n];

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

         System.out.print("Name of book: ");
         String name = sc.nextLine();

         System.out.print("Author name: ");
         String author = sc.nextLine();
```

```
            System.out.print("Price of book: ");
            double price = sc.nextDouble();

            System.out.print("No of pages of book: ");
            int numpages = sc.nextInt();
            sc.nextLine(); // consume newline

            b[i] = new Book(name, author, price, numpages);
        }

        System.out.println("\nDetails of books");

        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1));
            System.out.println(b[i]);
        }

        sc.close();
    }
}
```

Output:



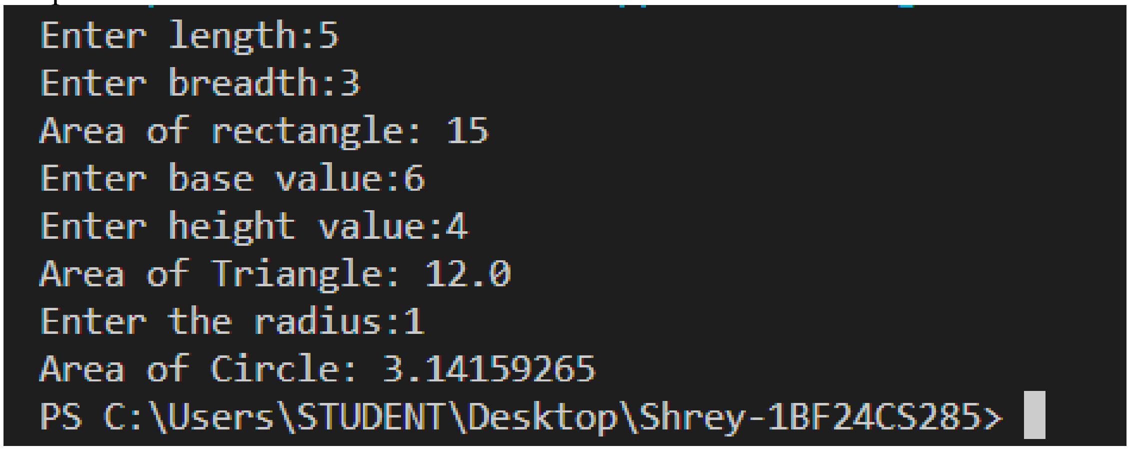
```
C:\Users\31shr\OneDrive\Desktop\CIE_PRACITSE>java Program03
Enter number of books: 1

Enter details of book 1
Name of book: Name
Author name: Author
Price of book: 200
No of pages of book: 300

Details of books

Book 1
Name : Name
Author : Author
Price : 200.0
No of pages : 300
```

Program 4: Shapes Program

Code:

```
import java.util.Scanner;

class Input {
   Scanner sc = new Scanner(System.in);

   int getInt(String message) {
      System.out.print(message);
      return sc.nextInt();
   }
}

abstract class Shape extends Input {
   int a, b;
   abstract void printArea();
}

class Rectangle extends Shape {
   void input() {
      a = getInt("Enter length:");
      b = getInt("Enter breadth:");
   }

   void printArea() {
      System.out.println("Area of rectangle: " + (a * b));
   }
}

class Triangle extends Shape {
   void input() {
      a = getInt("Enter base value:");
      b = getInt("Enter height value:");
   }

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

class Circle extends Shape {
   void input() {
      a = getInt("Enter the radius:");
   }

   void printArea() {
      System.out.println("Area of Circle: " + (3.14159265 * a * a));
   }
}

public class abstractClass {
   public static void main(String args[]) {
      Rectangle r = new Rectangle();
      r.input();
      r.printArea();

      Triangle t = new Triangle();
      t.input();
      t.printArea();

      Circle c = new Circle();
      c.input();
      c.printArea();
   }
}
```

Output:



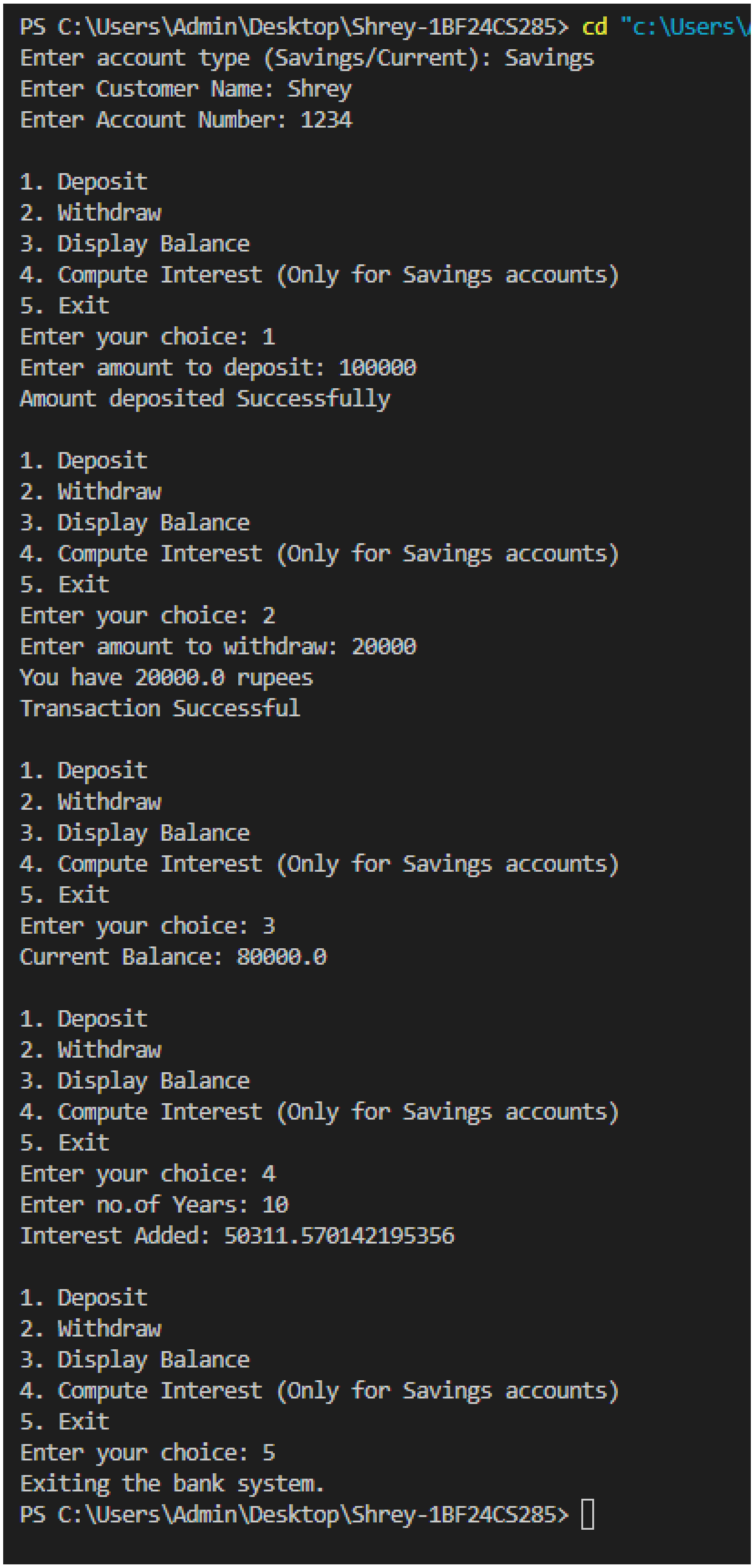
```
Enter length:5
Enter breadth:3
Area of rectangle: 15
Enter base value:6
Enter height value:4
Area of Triangle: 12.0
Enter the radius:1
Area of Circle: 3.14159265
PS C:\Users\STUDENT\Desktop\Shrey-1BF24CS285>
```

Program 5: Bank Program

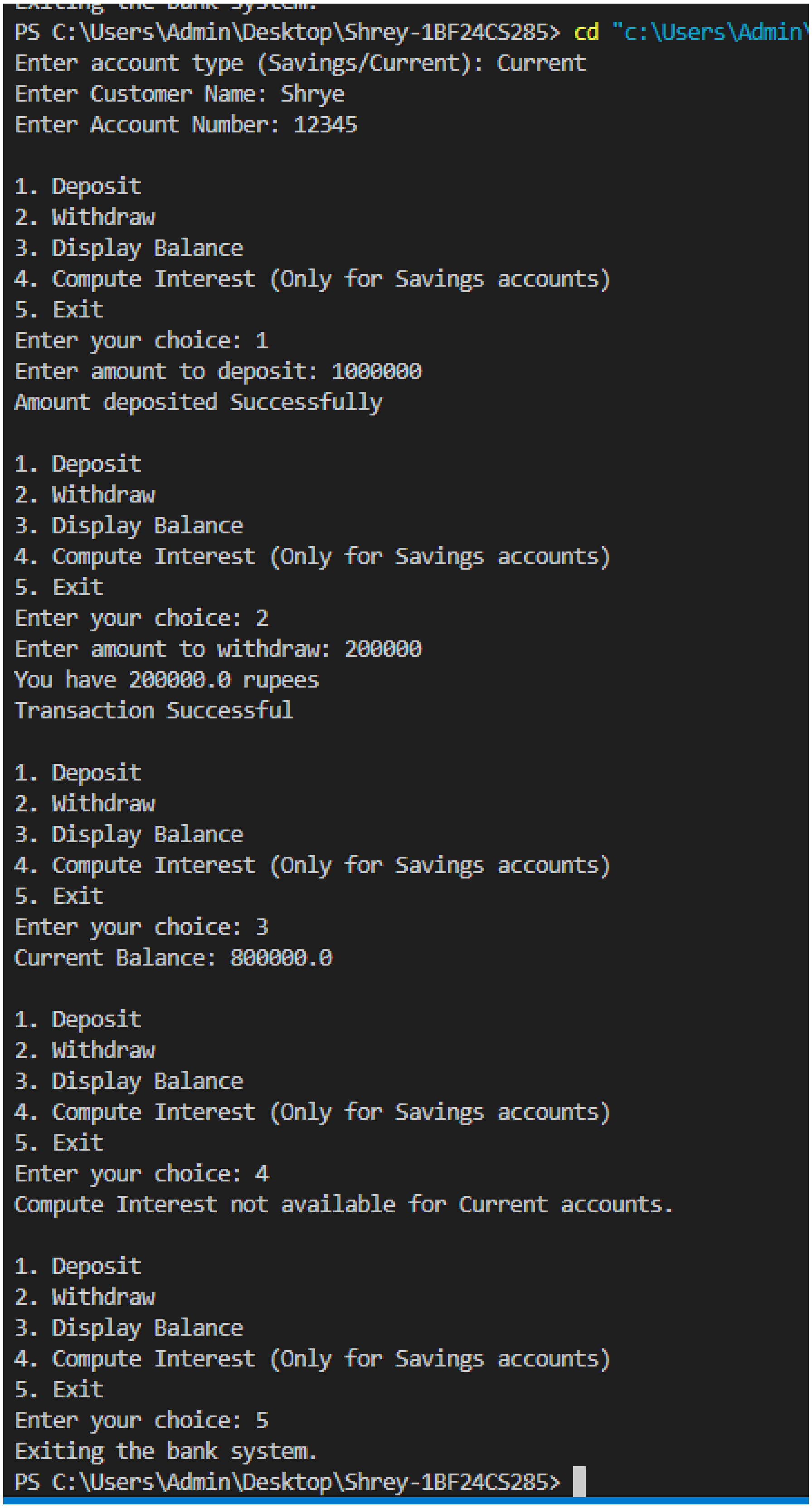
Code:

```
import java.util.Scanner;

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

    Scanner sc = new Scanner(System.in);

    void input(){
        System.out.print("Enter Customer Name: ");
        customerName = sc.nextLine();

        System.out.print("Enter Account Number: ");
        accountNumber = sc.nextInt();
        sc.nextLine();
    }


    void deposit(){
        System.out.print("Enter amount to deposit: ");
        double amount = sc.nextDouble();

        balance += amount;
        System.out.println("Amount deposited Successfully");
    }


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

    void withdraw(){
        System.out.print("Enter amount to withdraw: ");
        double amount = sc.nextDouble();

        if(amount <= balance){
            balance -= amount;
            System.out.println("You have " + amount + " rupees");
            System.out.println("Transaction Successful");
            checkMinBalance();
        }
        else{
            System.out.println("Insufficient balance");
        }
    }

    void checkMinBalance(){

    }

}



class Savings extends Account{
    final double interestRate = 0.05;
    Savings() {
        accountType = "Savings";
    }

    void computeInterest(){
        System.out.print("Enter no.of Years: ");
        double years = sc.nextDouble();

        double interest = balance * Math.pow((1 + interestRate), years) - balance;
        balance += interest;

        System.out.println("Interest Added: " + interest);
    }

}


class Current extends Account{
    final double serviceCharge = 100;
    final double minBalance = 1000;

    Current() {
        accountType = "Current";
    }


    @Override
    void checkMinBalance(){
        if (balance < minBalance){
            balance -= serviceCharge;
            System.out.println("Balance < minBalance");
            System.out.print("Penalty of " + serviceCharge + " is imposed");
        }
    }

}


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

        System.out.print("Enter account type (Savings/Current): ");
        String accountType = sc.nextLine();

        Account account = null;

        if (accountType.trim().equalsIgnoreCase("Savings")) {
            account = new Savings();
        } else if (accountType.equalsIgnoreCase("Current")) {
            account = new Current();
        } else {
            System.out.println("Invalid account type.");
            return;
        }

        account.input();

        while (true) {
            printMenu();
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    account.deposit();
                    break;
                case 2:
                    account.withdraw();
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                    if (account instanceof Savings) {
                        ((Savings) account).computeInterest();
                    }

                    else {
                        System.out.println("Compute Interest not available for Current accounts.");
                    }
                    break;
                case 5:
                    System.out.println("Exiting the bank system.");
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }

    public static void printMenu() {
        System.out.println("\n1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Display Balance");
        System.out.println("4. Compute Interest (Only for Savings accounts)");
        System.out.println("5. Exit");
    }

}
```

Output:



```
PS C:\Users\Admin\Desktop\Shrey-1BF24CS285> cd "c:\Users\
Enter account type (Savings/Current): Savings
Enter Customer Name: Shrey
Enter Account Number: 1234

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 1
Enter amount to deposit: 100000
Amount deposited Successfully

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 20000
You have 20000.0 rupees
Transaction Successful

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 3
Current Balance: 80000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 4
Enter no.of Years: 10
Interest Added: 50311.570142195356

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 5
Exiting the bank system.
PS C:\Users\Admin\Desktop\Shrey-1BF24CS285> 
```



```
Exiting the bank system.
PS C:\Users\Admin\Desktop\Shrey-1BF24CS285> cd "c:\Users\Admin\
Enter account type (Savings/Current): Current
Enter Customer Name: Shrye
Enter Account Number: 12345

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 1
Enter amount to deposit: 1000000
Amount deposited Successfully

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 200000
You have 200000.0 rupees
Transaction Successful

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 3
Current Balance: 800000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 4
Compute Interest not available for Current accounts.

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 5
Exiting the bank system.
PS C:\Users\Admin\Desktop\Shrey-1BF24CS285>
```

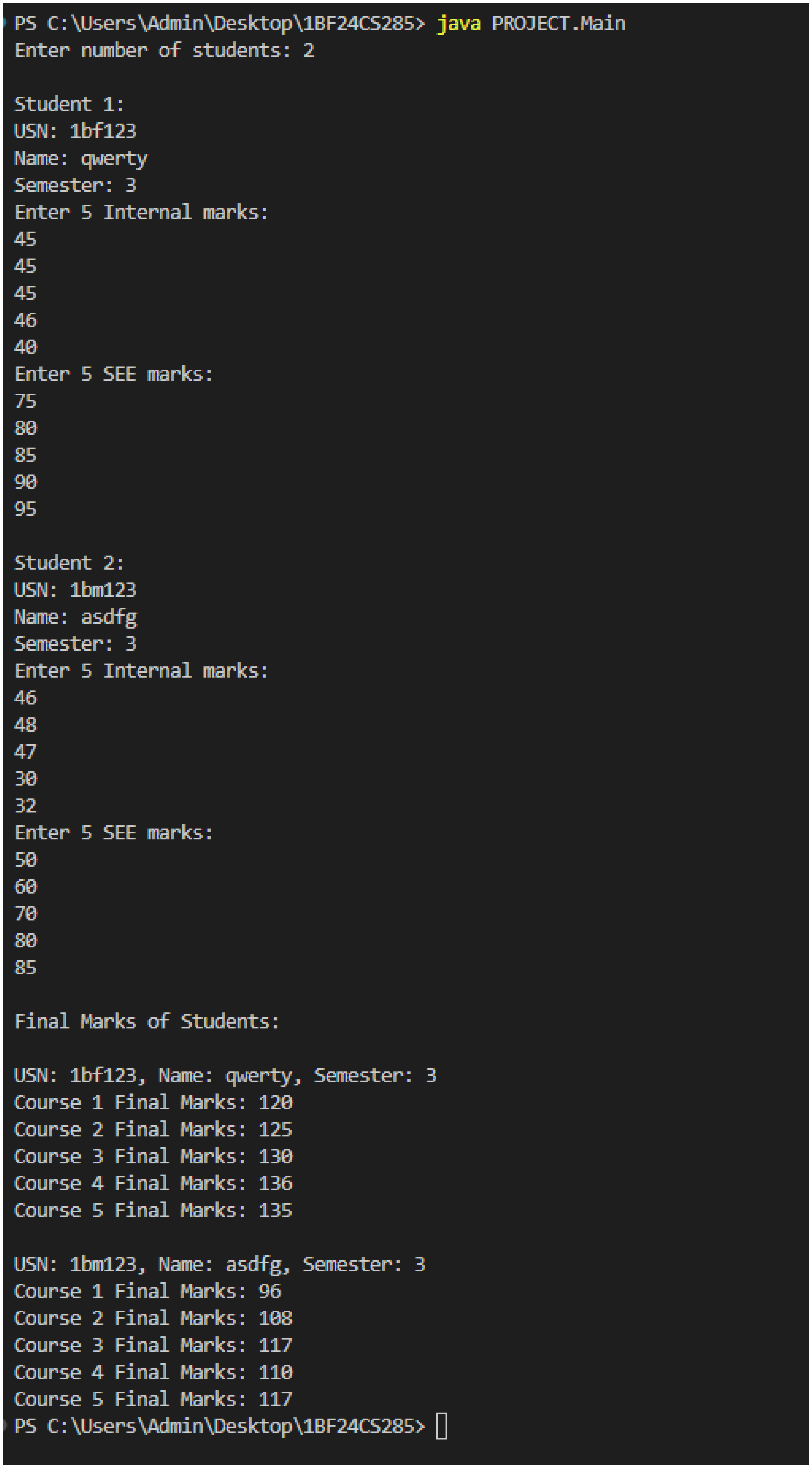
Program 6: Packages

Code:

```
package CIE;

public class Internal {
   public int[] marks;

   public Internal(int[] marks) {
      this.marks = marks;
   }
}
package SEE;

import CIE.Student;

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

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

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

   public Student(String usn, String name, int sem) {
      this.usn = usn;
      this.name = name;
      this.sem = sem;
   }
}
package PROJECT;

import java.util.Scanner;

import CIE.Student;
import CIE.Internal;
import SEE.External;

public class Main {

   public static void main(String[] args) {
```

```
Scanner sc = new Scanner(System.in);

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

Student[] personalData = new Student[n];
Internal[] internalMarks = new Internal[n];
External[] externalMarks = new External[n];

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

    System.out.println("\nStudent " + (i + 1) + ":");

    System.out.print("USN: ");
    String usn = sc.next();

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

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

    personalData[i] = new Student(usn, name, sem);

    System.out.println("Enter 5 Internal marks:");
    int[] internals = new int[5];
    for (int j = 0; j < 5; j++) {
        internals[j] = sc.nextInt();
    }
    internalMarks[i] = new Internal(internals);

    // External SEE marks
    System.out.println("Enter 5 SEE marks:");
    int[] see = new int[5];
    for (int j = 0; j < 5; j++) {
        see[j] = sc.nextInt();
    }
    externalMarks[i] = new External(usn, name, sem, see);
}

// Display final results
System.out.println("\nFinal Marks of Students:");

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

    System.out.println("\nUSN: " + personalData[i].usn +
                    ", Name: " + personalData[i].name +
                    ", Semester: " + personalData[i].sem);

            for (int j = 0; j < 5; j++) {

                int finalMark = internalMarks[i].marks[j] + externalMarks[i].seeMarks[j];

                System.out.println("Course " + (j + 1) + " Final Marks: " + finalMark);
            }
        }

        sc.close();
    }
}
```

Output:



```
PS C:\Users\Admin\Desktop\1BF24CS285> java PROJECT.Main
Enter number of students: 2

Student 1:
USN: 1bf123
Name: qwerty
Semester: 3
Enter 5 Internal marks:
45
45
45
46
40
Enter 5 SEE marks:
75
80
85
90
95

Student 2:
USN: 1bm123
Name: asdfg
Semester: 3
Enter 5 Internal marks:
46
48
47
30
32
Enter 5 SEE marks:
50
60
70
80
85

Final Marks of Students:

USN: 1bf123, Name: qwerty, Semester: 3
Course 1 Final Marks: 120
Course 2 Final Marks: 125
Course 3 Final Marks: 130
Course 4 Final Marks: 136
Course 5 Final Marks: 135

USN: 1bm123, Name: asdfg, Semester: 3
Course 1 Final Marks: 96
Course 2 Final Marks: 108
Course 3 Final Marks: 117
Course 4 Final Marks: 110
Course 5 Final Marks: 117
PS C:\Users\Admin\Desktop\1BF24CS285> 
```

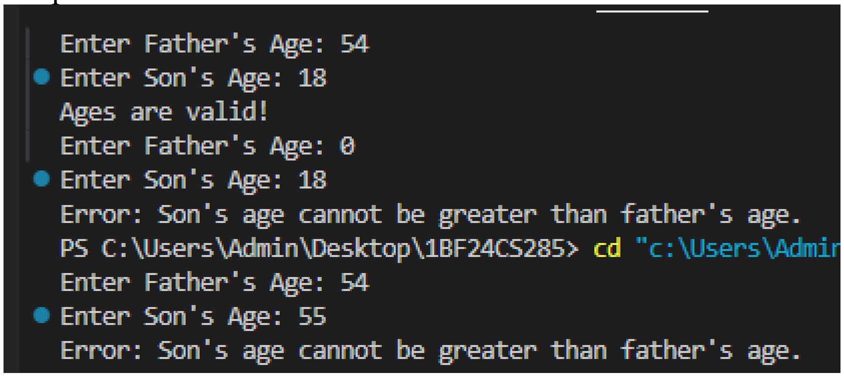
Program 7: Errors

Code:

```
import java.util.Scanner;

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

class Father {
   int father_age;

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

class Son extends Father {
   int son_age;

   Son(int father_age, int son_age) throws WrongAgeException {
      super(father_age);

      if (son_age > father_age) {
         throw new WrongAgeException("Son's age cannot be greater than father's age.");
      }
      this.son_age = son_age;
   }
}

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

      System.out.print("Enter Father's Age: ");
      int father_age = sc.nextInt();

      System.out.print("Enter Son's Age: ");
      int son_age = sc.nextInt();

      try {
         Son s = new Son(father_age, son_age);
```

```
            System.out.println("Ages are valid!");
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }

        sc.close();
    }
}
```

Output:



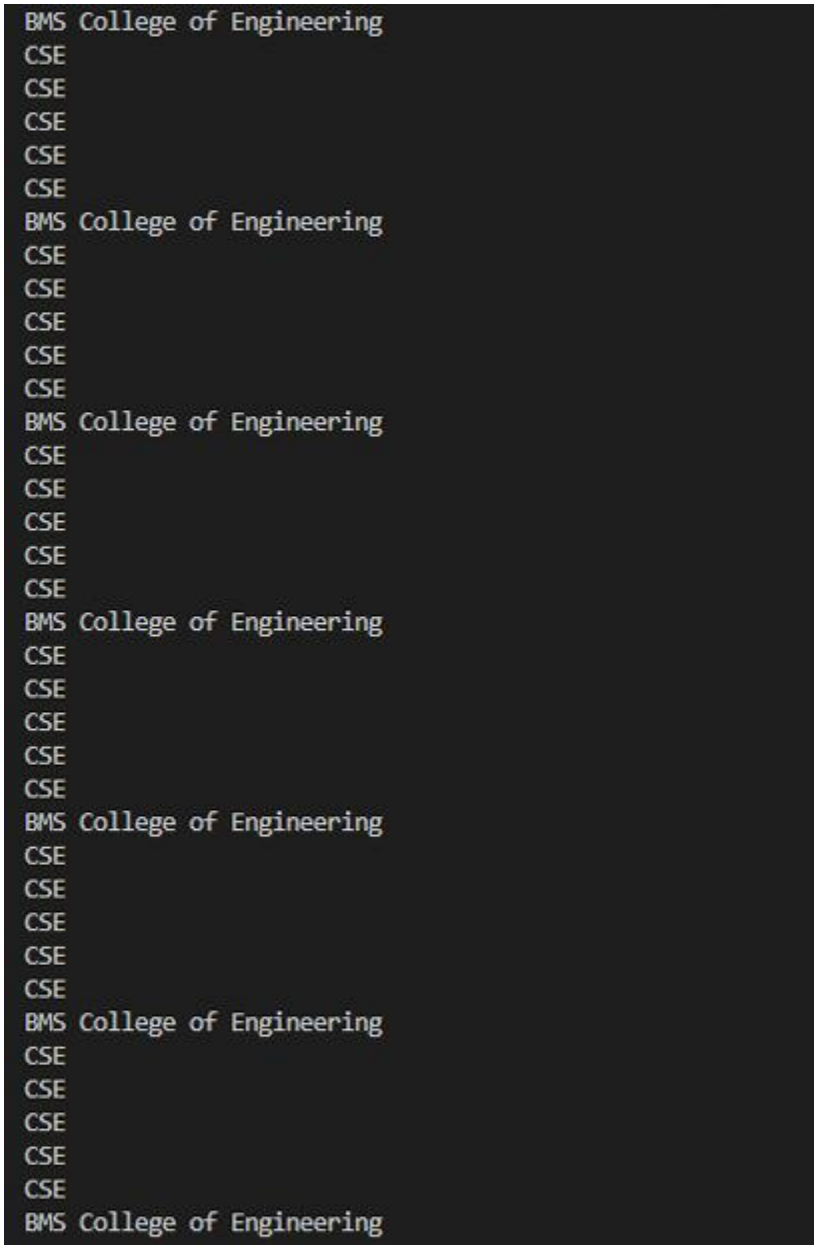
```
Enter Father's Age: 54
Enter Son's Age: 18
Ages are valid!
Enter Father's Age: 0
Enter Son's Age: 18
Error: Son's age cannot be greater than father's age.
PS C:\Users\Admin\Desktop\1BF24CS285> cd "c:\Users\Admin
Enter Father's Age: 54
Enter Son's Age: 55
Error: Son's age cannot be greater than father's age.
```

Program 8: Multi Threading

Code:

```
class MsgThread extends Thread
{
   private String msg;
   private int interval;

   public MsgThread(String msg, int interval)
   {
      this.msg = msg;
      this.interval = interval;
   }

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

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

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

Output:



```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

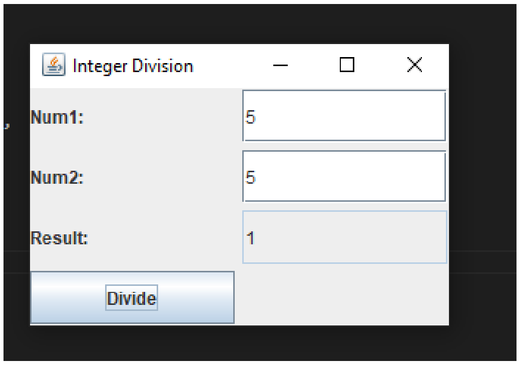
<u>Program 9:</u> Open Ended Question 1

Code:

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

public class IntegerDivisionUI extends JFrame {

   private JTextField num1Field, num2Field, resultField;
   private JButton divideButton;

   public IntegerDivisionUI() {
      setTitle("Integer Division");
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLocationRelativeTo(null);

      // Layout
      setLayout(new GridLayout(4, 2, 5, 5));

      add(new JLabel("Num1:"));
      num1Field = new JTextField();
      add(num1Field);

      add(new JLabel("Num2:"));
      num2Field = new JTextField();
      add(num2Field);

      add(new JLabel("Result:"));
      resultField = new JTextField();
      resultField.setEditable(false);
      add(resultField);

      divideButton = new JButton("Divide");
      add(divideButton);

      // Action listener
      divideButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            try {
               // Parse integers (may throw NumberFormatException)
               int num1 = Integer.parseInt(num1Field.getText());
               int num2 = Integer.parseInt(num2Field.getText());

               // Perform division (may throw ArithmeticException)
```

```
                    int result = num1 / num2;

                    // Display result
                    resultField.setText(Integer.toString(result));

                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(
                            null,
                            "Please enter valid integers!",
                            "Number Format Error",
                            JOptionPane.ERROR_MESSAGE
                    );
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(
                            null,
                            "Division by zero is not allowed!",
                            "Arithmetic Error",
                            JOptionPane.ERROR_MESSAGE
                    );
                }
            }
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new IntegerDivisionUI().setVisible(true);
        });
    }
}
```

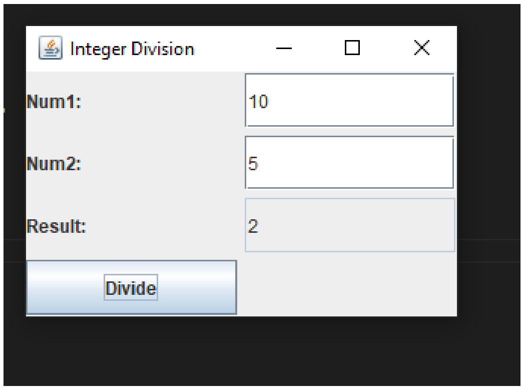
Output:



Num1: 5

Num2: 5

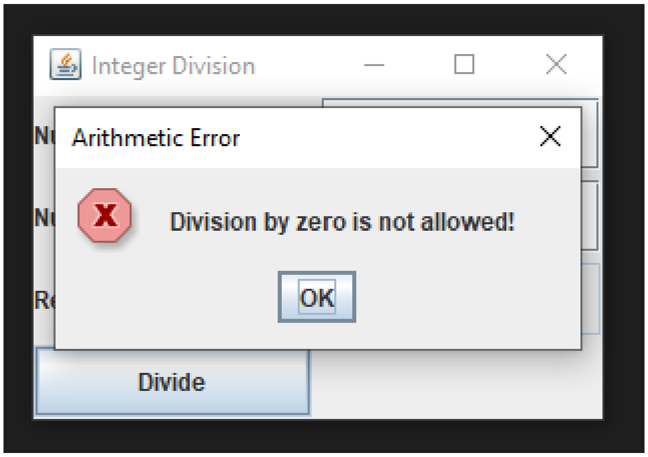
Result: 1



Num1: 10

Num2: 5

Result: 2



Division by zero is not allowed!

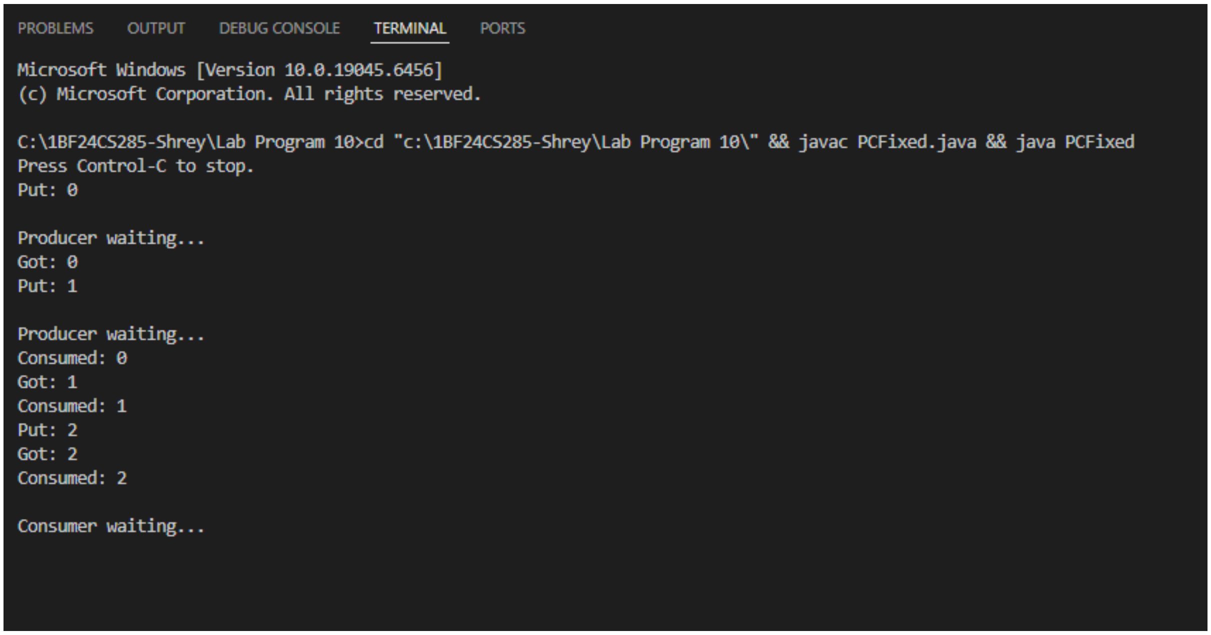
Program 10: Open Ended Question 2

Code:

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

   synchronized int get() {
      // Wait if there is no value to consume
      while (!valueSet) {
         try {
            System.out.println("\nConsumer waiting...");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      }
      // Once a value is available, consume it
      System.out.println("Got: " + n);
      valueSet = false;
      // Notify the producer that space is available
      notify();
      return n;
   }

   synchronized void put(int n) {
      // Wait if there is already a value to consume (can't put another one)
      while (valueSet) {
         try {
            System.out.println("\nProducer waiting...");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      }
      // Once space is available, produce the value
      this.n = n;
      valueSet = true;
      System.out.println("Put: " + n);
      // Notify the consumer that a new value is available
      notify();
   }
}

class Producer implements Runnable {
   Q q;
```

```
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start(); // Start the producer thread
    }

    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(); // Start the consumer thread
    }

    public void run() {
        int r = 0;
        while (r < 3) {
            r = q.get();
            System.out.println("Consumed: " + r);
        }
    }
}

class PCFixed {
    public static void main(String args[]) {
        Q q = new Q(); // Create a shared queue object
        new Producer(q); // Start the producer
        new Consumer(q); // Start the consumer
        System.out.println("Press Control-C to stop.");
    }
}
```

Output:



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

C:\1BF24CS285-Shrey\Lab Program 10>cd "c:\1BF24CS285-Shrey\Lab Program 10\" && javac PCFixed.java && java PCFixed
Press Control-C to stop.
Put: 0

Producer waiting...
Got: 0
Put: 1

Producer waiting...
Consumed: 0
Got: 1
Consumed: 1
Put: 2
Got: 2
Consumed: 2

Consumer waiting...
```