B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

*Submitted by:***ARJUN PRABHAKARAN**

1BM22CS053

Department of Computer Science and Engineering B.M.S
College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019 2023-
2024

B.M.S COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I, ARJUN PRABHAKARN (1BM22CS053) student in 2nd Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this LAB REPORT for OBJECT ORIENTED JAVA PROGRAMMING has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester December march 2024.

Signature of the Candidate

ARJUN PRABHAKARN (1BM22CS053)

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the lab report for OBJECT ORIENTED JAVA PROGRAMMING has been carried out by ARJUN PRABHAKARAN (1BM22CS053) during the academic year 2023-2024.

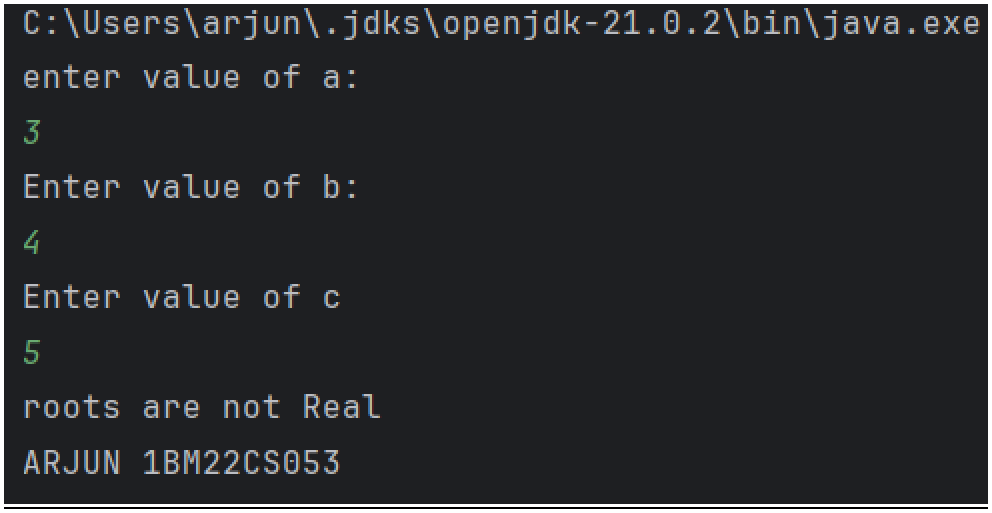
Signature of the Faculty in Charge

Table of Contents

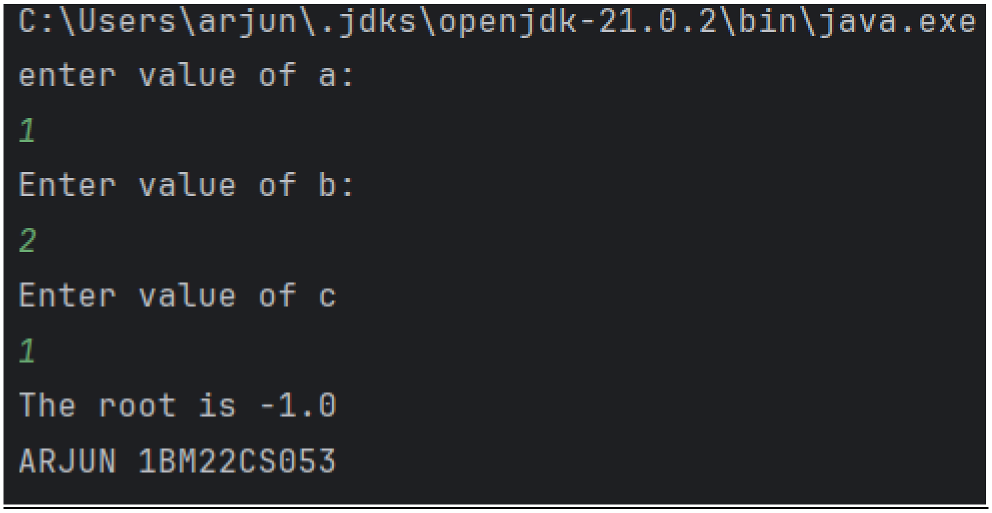
Program 1: Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c= 0. Read in a, b, c and use the quadratic formula. If the discriminant b2-4ac is negative, display a message stating that there are no real solutions.

```
public class Quadratic {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("enter value of a:");
        double a = s.nextDouble();
        System.out.println("Enter value of b:");
        double b = s.nextDouble();
        System.out.println("Enter value of c");
        double c = s.nextDouble();

        double d = b*b - 4.0*a*c;

        if(d>0.0)
        {
            double r1 = (-b+Math.pow(d,0.5))/(2.0+a);
            double r2 = (-b-Math.pow(d,0.5))/(2.0+a);
            System.out.println("The roots are"+r1+"and"+r2);
        }
        if(d==0)
        {
            double r1 = -b/(2.0*a);
            System.out.println("The root is "+r1);
        }
    else
    {
      System.out.println("roots are not Real");
    }
    System.out.println("ARJUN 1BM22CS053");
  }
}
```

Output:



```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
enter value of a:
3
Enter value of b:
4
Enter value of c
5
roots are not Real
ARJUN 1BM22CS053
```

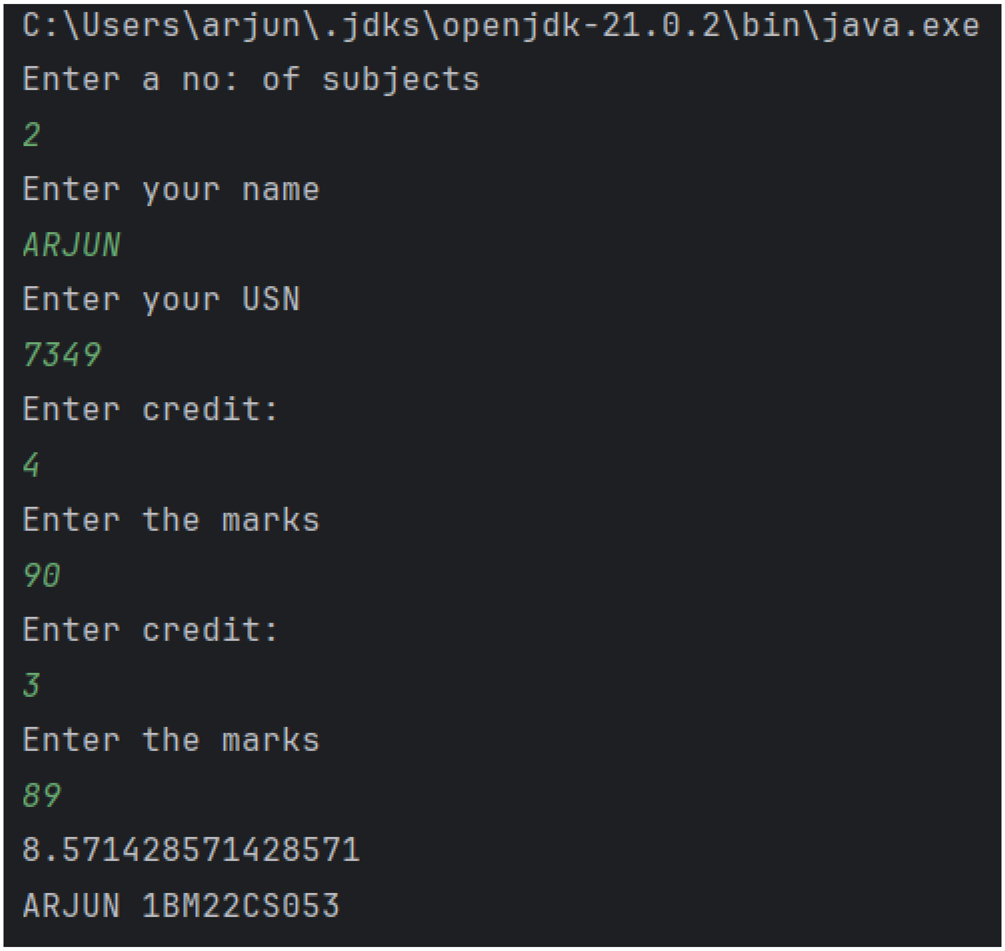


```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
enter value of a:
1
Enter value of b:
2
Enter value of c
1
The root is -1.0
ARJUN 1BM22CS053
```

Program 2: Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.*;
class SS
{
   String usn, name;
   int sum;
   int [] credit = new int[10];
   int [] marks = new int[10];
   double sgpa = 0.0;

   void Accept(int n) {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter your name");
      name = s.next();
      System.out.println("Enter your USN");
      usn = s.next();

      for (int i = 0; i < n; i++) {
         System.out.println("Enter credit: ");
         credit[i] = s.nextInt();
         System.out.println("Enter the marks");
         marks[i] = s.nextInt();
      }
   }
   void calculate (int n) {
      for (int i = 0; i < n; i++) {
         sgpa = sgpa + credit[i] * (int) (marks[i] / 10);
         sum = sum + credit[i];
      }
      sgpa = sgpa / sum;
      System.out.println(sgpa);
   }
}
public class SSS {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter a no: of subjects ");
        int n = s.nextInt();
        SS S = new SS();
        S.Accept(n);
        S.calculate(n);
        System.out.println("ARJUN 1BM22CS053");
    }
}
```

Output:



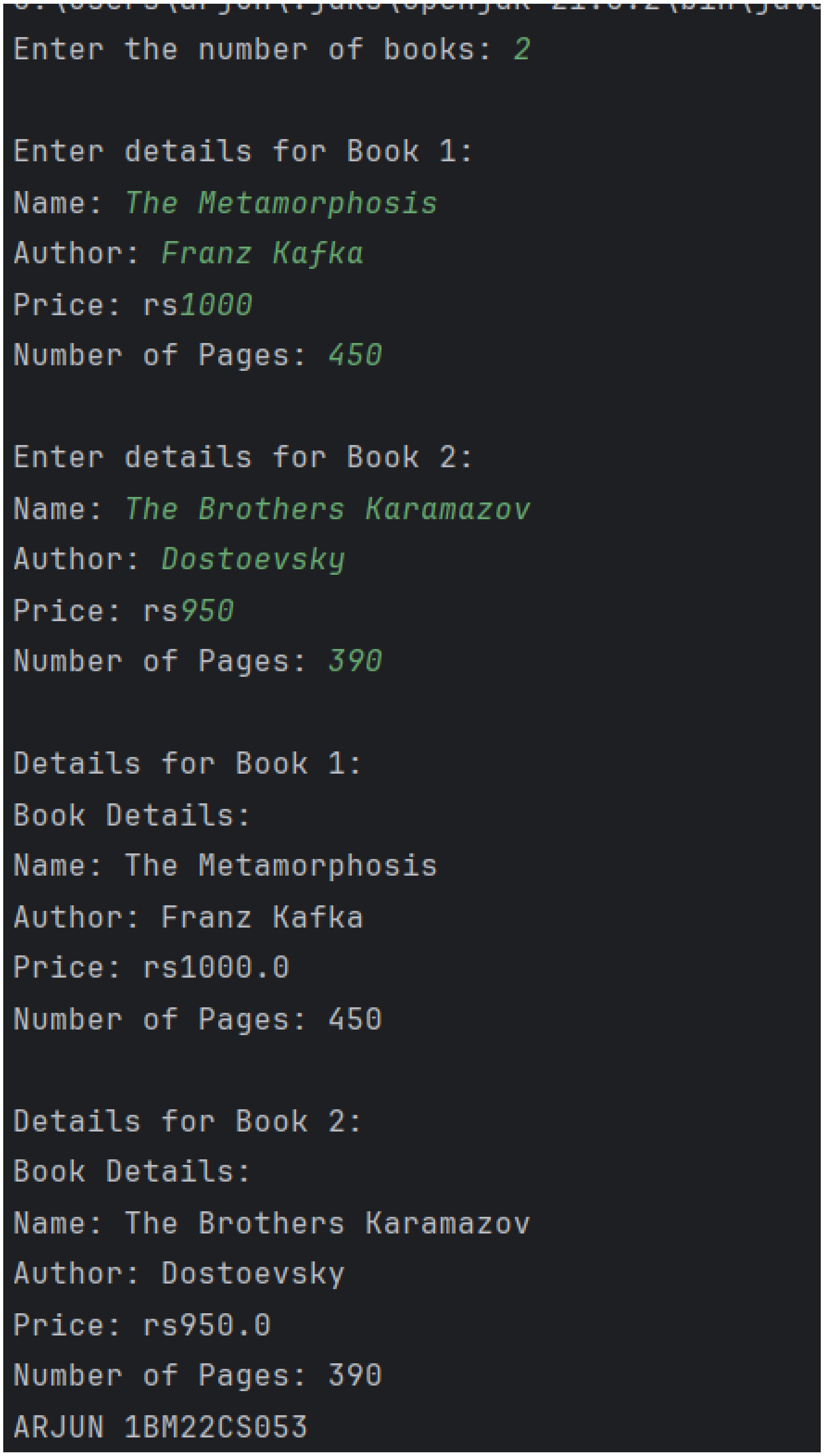
```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Enter a no: of subjects
2
Enter your name
ARJUN
Enter your USN
7349
Enter credit:
4
Enter the marks
90
Enter credit:
3
Enter the marks
89
8.571428571428571
ARJUN 1BM22CS053
```

Program 3: Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;

class Book {
   private String name;
   private String 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;
   }


   public void setName(String name) {
      this.name = name;
   }

   public String getName() {
      return name;
   }

   public void setAuthor(String author) {
      this.author = author;
```

```
    }

    public String getAuthor() {
        return author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getPrice() {
        return price;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public int getNumPages() {
        return numPages;
    }



    public String toString() {
        return "Book Details: \nName: " + name + "\nAuthor: " + author + "\nPrice:
rs" + price + "\nNumber of Pages: " + numPages;
    }
}

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

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

        Book[] books = new Book[n];

        // Input details for each book
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            s.nextLine(); // Consume the newline character
            System.out.print("Name: ");
            String name = s.nextLine();
            System.out.print("Author: ");
            String author = s.nextLine();
            System.out.print("Price: $");
            double price = s.nextDouble();
            System.out.print("Number of Pages: ");
            int numPages = s.nextInt();

            books[i] = new Book(name, author, price, numPages);
        }

        // Display details of each book
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails for Book " + (i + 1) + ":");
            System.out.println(books[i]);
        }
        System.out.println("ARJUN 1BM22CS053");
    }
}
```

Output:

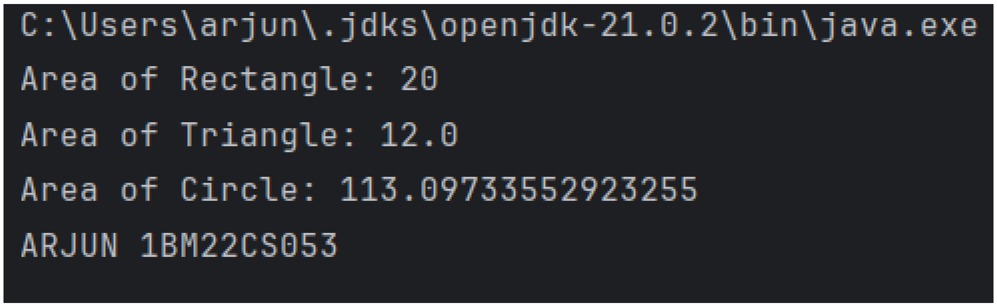


```
Enter the number of books: 2

Enter details for Book 1:
Name: The Metamorphosis
Author: Franz Kafka
Price: rs1000
Number of Pages: 450

Enter details for Book 2:
Name: The Brothers Karamazov
Author: Dostoevsky
Price: rs950
Number of Pages: 390

Details for Book 1:
Book Details:
Name: The Metamorphosis
Author: Franz Kafka
Price: rs1000.0
Number of Pages: 450

Details for Book 2:
Book Details:
Name: The Brothers Karamazov
Author: Dostoevsky
Price: rs950.0
Number of Pages: 390
ARJUN 1BM22CS053
```

Program 4: Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shape.

```
abstract class Shape {
   protected int side1;
   protected int side2;

   public Shape(int side1, int side2) {
      this.side1 = side1;
      this.side2 = side2;
   }

   public abstract void printArea();
}

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


   public void printArea() {
      int area = side1 * side2;
      System.out.println("Area of Rectangle: " + area);
   }
}

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


    public void printArea() {
        double area = 0.5 * side1 * side2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0); // Considering only one side for the radius
    }


    public void printArea() {
        double area = Math.PI * side1 * side1;
        System.out.println("Area of Circle: " + area);
    }
}

public class ShapeTest {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(4, 5);
        Triangle triangle = new Triangle(3, 8);
        Circle circle = new Circle(6);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
        System.out.println("ARJUN 1BM22CS053");
    }
}
```

Output :



```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Area of Rectangle: 20
Area of Triangle: 12.0
Area of Circle: 113.09733552923255
ARJUN 1BM22CS053
```

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. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.
Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance

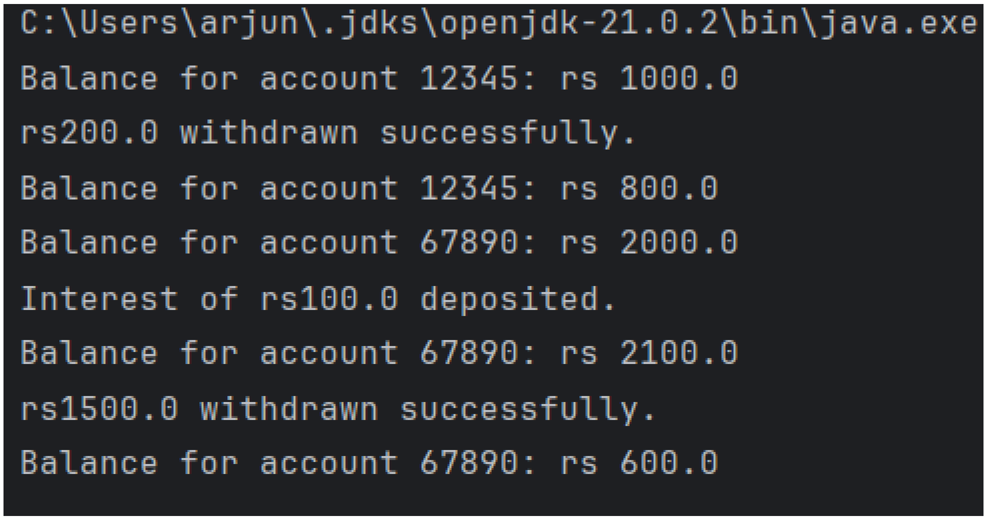
Check for the minimum balance, impose penalty if necessary and update the balance.

Code:

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

    Account(String name, int accNumber, String accType, double initialBalance) {
        customerName = name;
        accountNumber = accNumber;
        accountType = accType;
        balance = initialBalance;
    }
    void displayBalance() {
        System.out.println("Balance for account " + accountNumber + ": $" +
balance);
    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    CurAcct(String name, int accNumber, double initialBalance, double
minBalance, double serviceCharge) {
        super(name, accNumber, "Current", initialBalance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    void withdraw(double amount) {
        if (balance - amount < minBalance) {
            System.out.println("Withdrawal not permitted. Minimum balance not
maintained.");
            imposeServiceCharge();
        } else {
            balance -= amount;
            System.out.println("$" + amount + " withdrawn successfully.");
            displayBalance();
        }
    }

    private void imposeServiceCharge() {
        System.out.println("Service charge of $" + serviceCharge + " imposed.");
        balance -= serviceCharge;
        displayBalance();
    }
}
```

```
class SavAcct extends Account {
    double interestRate;

    SavAcct(String name, int accNumber, double initialBalance, double
interestRate) {
        super(name, accNumber, "Savings", initialBalance);
        this.interestRate = interestRate;
    }

    void depositInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of $" + interest + " deposited.");
        displayBalance();
    }

    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("$" + amount + " withdrawn successfully.");
            displayBalance();
        } else {
            System.out.println("Withdrawal not permitted. Insufficient funds.");
        }
    }
}

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

        CurAcct currentAccount = new CurAcct("John Doe", 12345, 1000.0, 500.0,
10.0);
        SavAcct savingsAccount = new SavAcct("Jane Smith", 67890, 2000.0, 5.0);

        // Example transactions
        currentAccount.displayBalance();
        currentAccount.withdraw(200.0);

        savingsAccount.displayBalance();
        savingsAccount.depositInterest();
        savingsAccount.withdraw(1500.0);
    }
}
```

Output:



```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Balance for account 12345: rs 1000.0
rs200.0 withdrawn successfully.
Balance for account 12345: rs 800.0
Balance for account 67890: rs 2000.0
Interest of rs100.0 deposited.
Balance for account 67890: rs 2100.0
rs1500.0 withdrawn successfully.
Balance for account 67890: rs 600.0
```

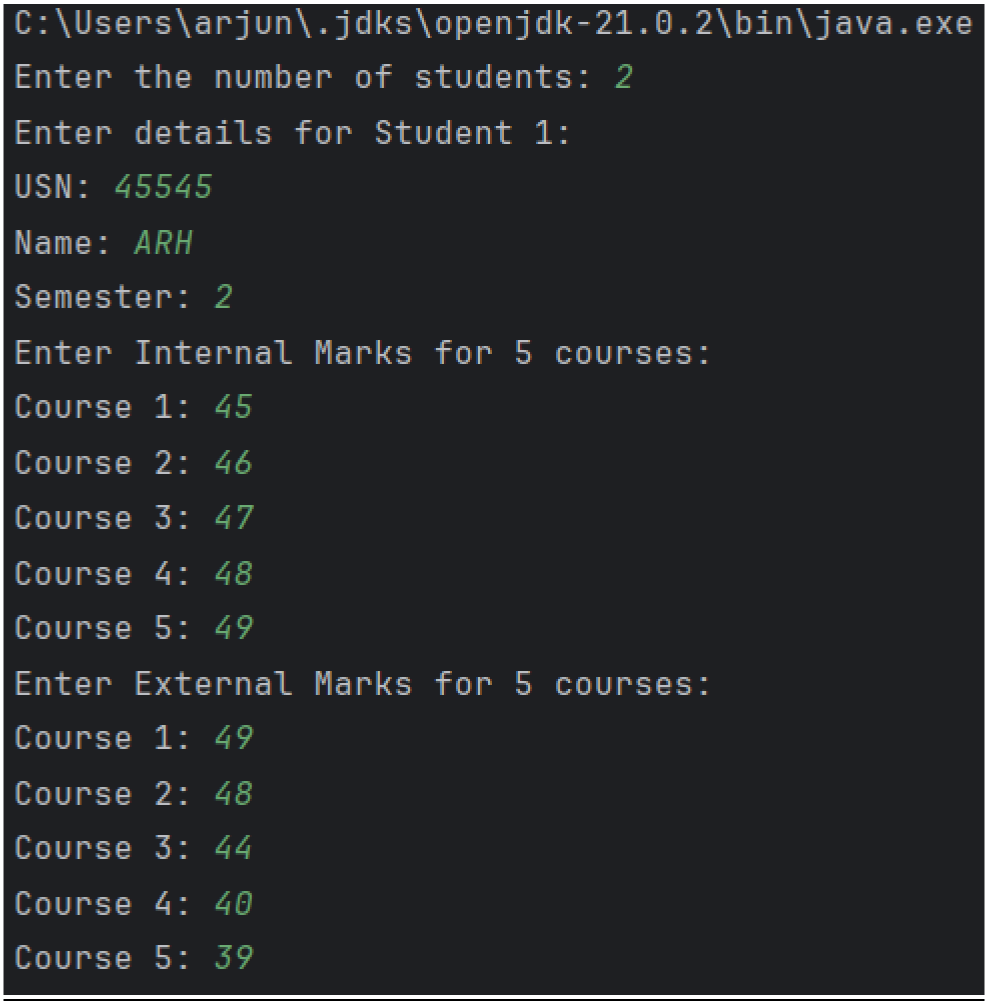
Program 6: Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Code:

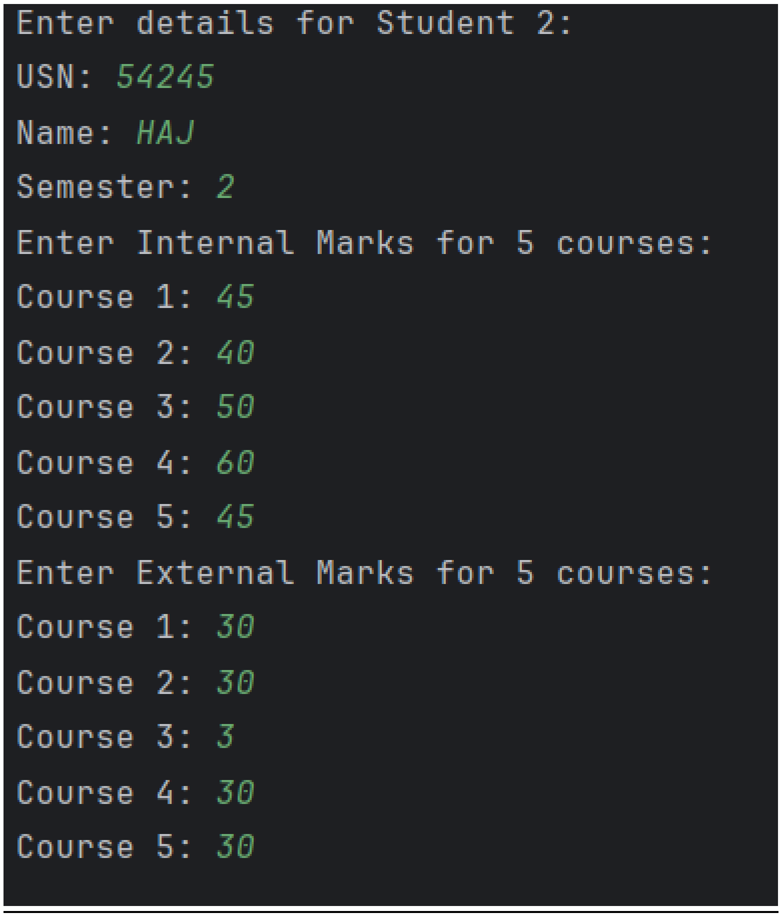
```
package CIE;

public class Internals extends Student {
   public int[] internalMarks = new int[5];
}

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

```
package SEE;

import CIE.Student;

public class External extends Student {
   public int[] seeMarks = new int[5];
}
```

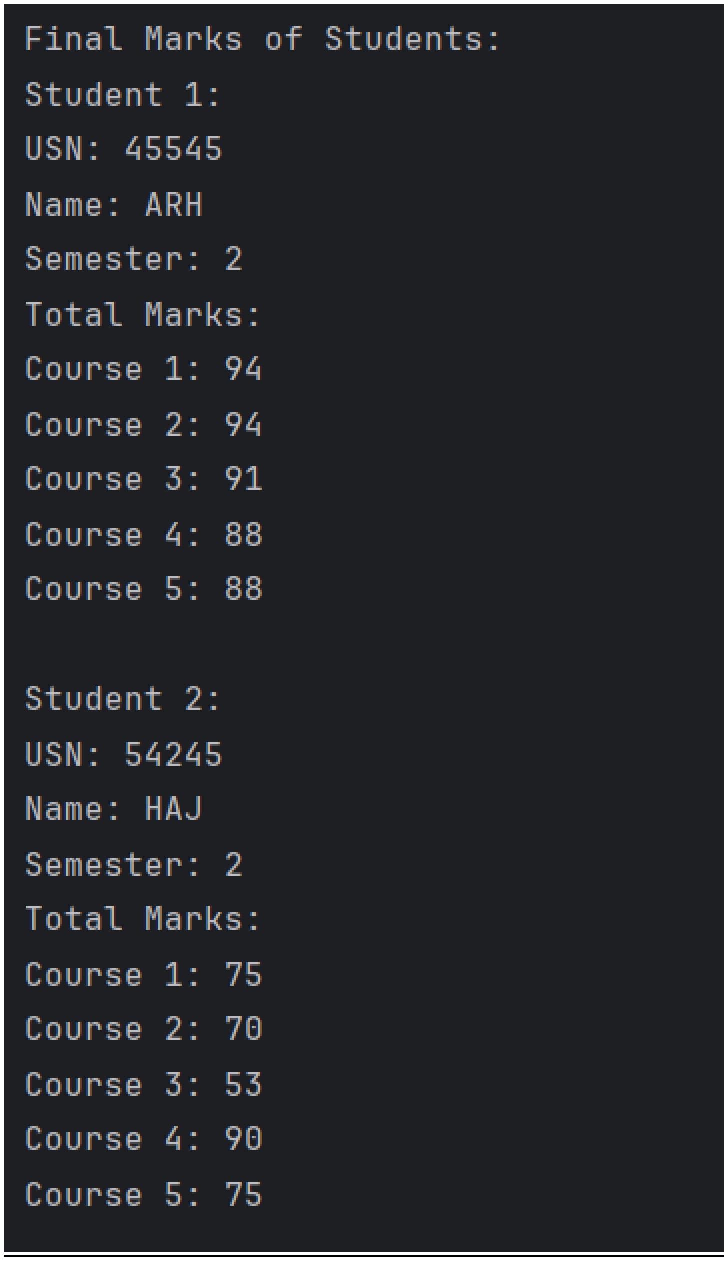
```
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);

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

        Internals[] cieStudents = new Internals[n];
        External[] seeStudents = new External[n];
        for (int i = 0; i < n; i++) {
            cieStudents[i] = new Internals();
            seeStudents[i] = new External();

            System.out.println("Enter details for Student " + (i + 1) + ":");
            System.out.print("USN: ");
            cieStudents[i].usn = seeStudents[i].usn = sc.next();
            System.out.print("Name: ");
            cieStudents[i].name = seeStudents[i].name = sc.next();
            System.out.print("Semester: ");
            cieStudents[i].sem = seeStudents[i].sem = sc.nextInt();

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

            System.out.println("Enter External Marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                seeStudents[i].seeMarks[j] = sc.nextInt();
            }
        }

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < 5; j++) {
                seeStudents[i].seeMarks[j] += cieStudents[i].internalMarks[j];
            }
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("Student " + (i + 1) + ":");
            System.out.println("USN: " + cieStudents[i].usn);
            System.out.println("Name: " + cieStudents[i].name);
            System.out.println("Semester: " + cieStudents[i].sem);

            System.out.println("Total Marks:");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course " + (j + 1) + ": " +
seeStudents[i].seeMarks[j]);
            }

            System.out.println();
        }
    }
}
```



```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Enter the number of students: 2
Enter details for Student 1:
USN: 45545
Name: ARH
Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 45
Course 2: 46
Course 3: 47
Course 4: 48
Course 5: 49
Enter External Marks for 5 courses:
Course 1: 49
Course 2: 48
Course 3: 44
Course 4: 40
Course 5: 39
```



```
Enter details for Student 2:
USN: 54245
Name: HAJ
Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 45
Course 2: 40
Course 3: 50
Course 4: 60
Course 5: 45
Enter External Marks for 5 courses:
Course 1: 30
Course 2: 30
Course 3: 3
Course 4: 30
Course 5: 30
```



```
Final Marks of Students:
Student 1:
USN: 45545
Name: ARH
Semester: 2
Total Marks:
Course 1: 94
Course 2: 94
Course 3: 91
Course 4: 88
Course 5: 88

Student 2:
USN: 54245
Name: HAJ
Semester: 2
Total Marks:
Course 1: 75
Course 2: 70
Course 3: 53
Course 4: 90
Course 5: 75
```

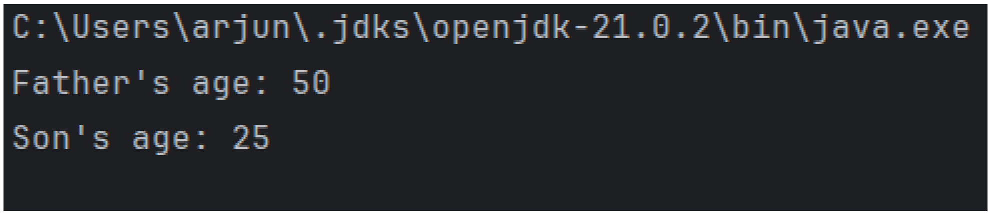
Program 7: Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

Code:

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

class Father {
   private int age;


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


   public int getAge() {
      return age;
   }
}
```

```java
class Son extends Father {
    private int sonAge;


    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge); // Call constructor of base class
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age should be less than Father's age");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class Main1 {
    public static void main(String[] args) {
        try {
            Father father = new Father(50); // Creating Father object with age 50
            Son son = new Son(father.getAge(), 25); // Creating Son object with age 25

            System.out.println("Father's age: " + father.getAge());
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Output:



```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Father's age: 50
Son's age: 25
```

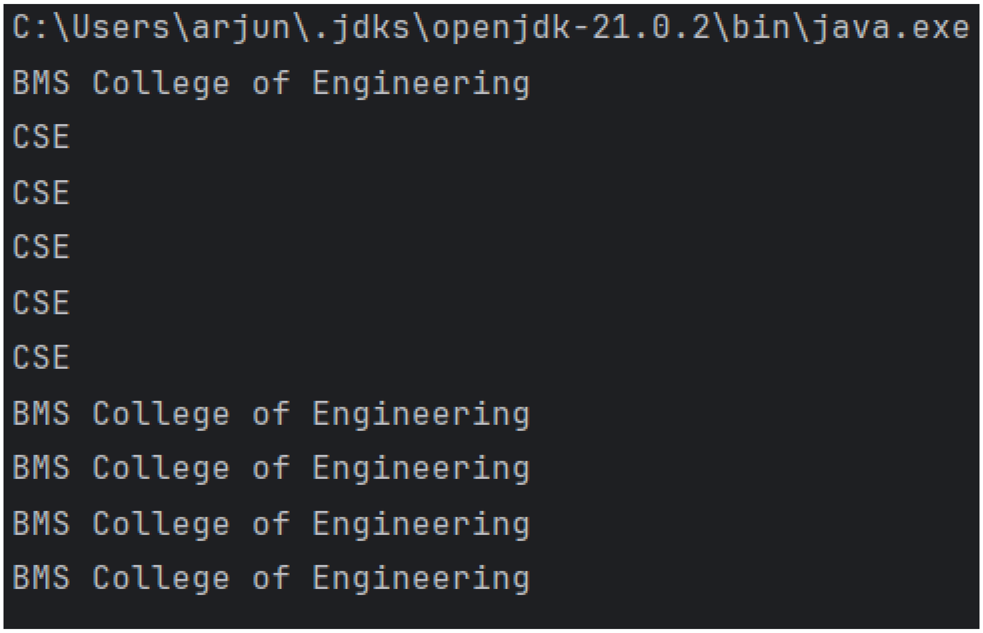
Program 8: Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Code:

```
class ps1 implements Runnable
{
   public void run()
   {
      for(int i=0;i<5;i++)
      {
         System.out.println("BMS College of Engineering");
         try{
            Thread.sleep(10000);
         } catch (Exception e)
         {
            e.printStackTrace();
         }
      }
   }
}
 class ps2 implements Runnable {
   public void run()
   {
      for(int i=0;i<5;i++)
      {
         System.out.println("CSE");
         try {
            Thread.sleep(2000);
         } catch (Exception e) {
            e.printStackTrace();
         }
```

```
        }
    }
}
public class MAIN2 {
    public static void main(String[] args) {
        ps1 p1 = new ps1();
        ps2 p2 = new ps2();

        Thread t1 = new Thread(p1);
        Thread t2 = new Thread(p2);
        t1.start();
        t2.start();

    }
}
```

Output:



```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

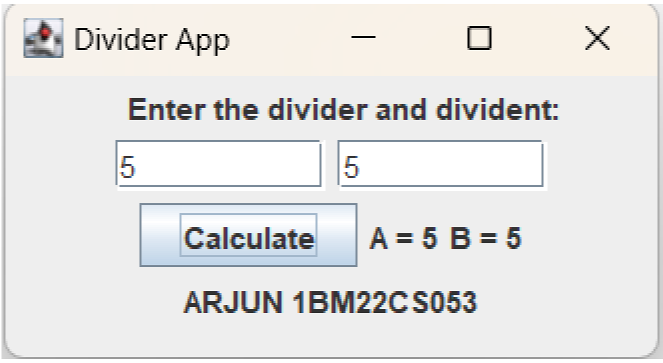
<u>Program 9:</u> Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo1 {
  SwingDemo1(){

    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());

    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JLabel jlab = new JLabel("Enter the divider and divident:");

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

    JButton button = new JButton("Calculate");

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

    jfrm.add(err); // to display error bois
    jfrm.add(jlab);
    jfrm.add(ajtf);
    jfrm.add(bjtf);
    jfrm.add(button);
    jfrm.add(alab);
    jfrm.add(blab);
    jfrm.add(anslab);
    ActionListener l = new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            System.out.println("Action event from a text field");
        }
    };
    ajtf.addActionListener(l);
    bjtf.addActionListener(l);
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            try{
                int a = Integer.parseInt(ajtf.getText());
                int b = Integer.parseInt(bjtf.getText());
                int ans = a/b;
                alab.setText("\nA = " + a);
                blab.setText("\nB = " + b);
                anslab.setText("\nAns = "+ ans);
                anslab.setText("ARJUN 1BM22CS053");
            }
            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 SwingDemo1();
            }
        });
    }
}
```

OUTPUT :

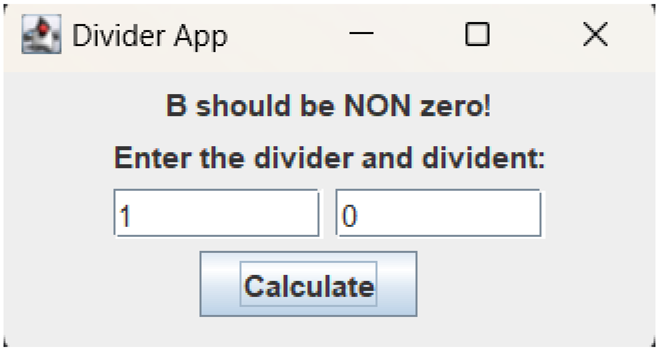


Enter the divider and divident:

5 5

A = 5 B = 5

ARJUN 1BM22CS053



B should be NON zero!

Enter the divider and divident:

1 0