B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY RECORD

ON

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted By :

Ayman Amjad

1BM22CS061

In partial fulfilment of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2023-24

Faculty-In-Charge

Shravya A R

Assistant Professor

Department of Computer Science and Engineering

INDEX

Program 1:

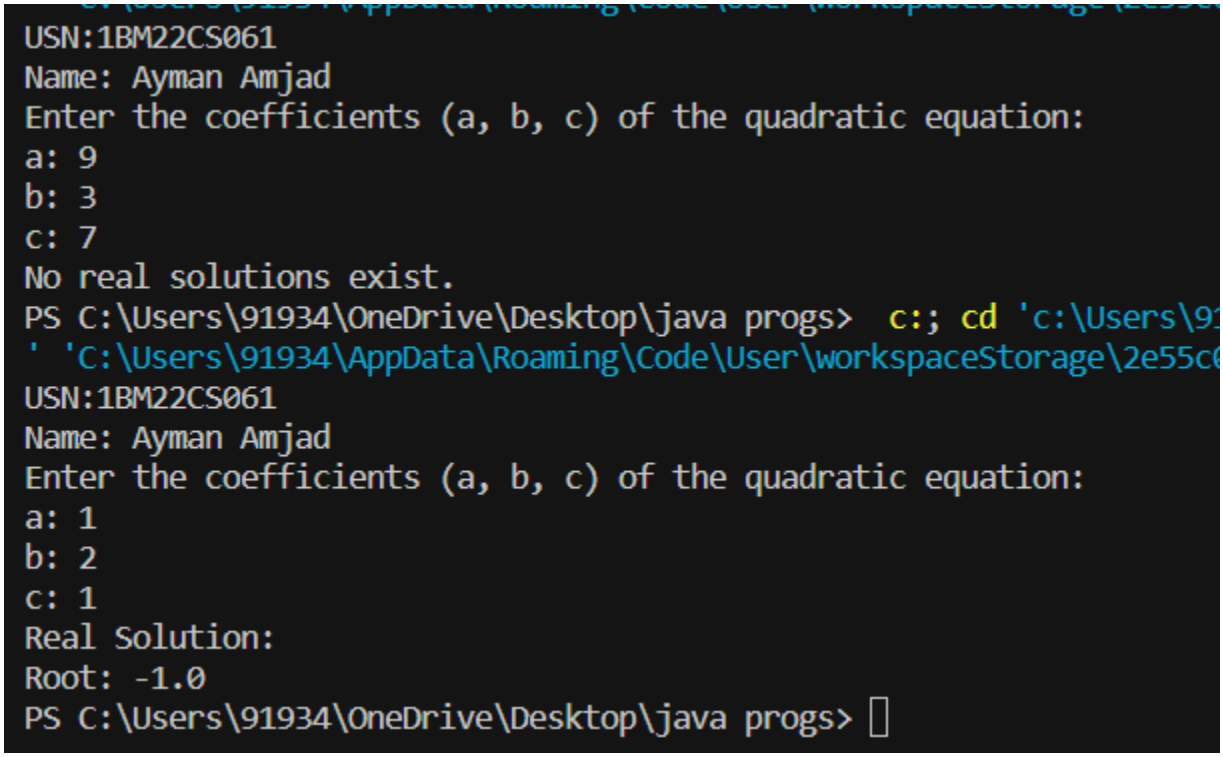
Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 - 4ac is negative, display a message stating that there are no real solutions.

CODE:

```
import java.util.*;
class QuadraticEquationSolver {
    double a, b, c;
    QuadraticEquationSolver(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }
    void takeInput() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the coefficients (a, b, c) of the quadratic equation:");
        System.out.print("a: ");
        this.a = scanner.nextDouble();
        System.out.print("b: ");
        this.b = scanner.nextDouble();
        System.out.print("c: ");
        this.c = scanner.nextDouble();
    }
    void calculateRoots() {
        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("Real Solutions:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        }
        else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("Real Solution:");
            System.out.println("Root: " + root);
        }
        else {
            System.out.println("No real solutions exist.");
        }
    }
}
```

```
public class prog1 {
    public static void main(String[] args) {
        QuadraticEquationSolver solver = new QuadraticEquationSolver(0,0, 0);
        System.out.println("USN:1BM22CS061");
        System.out.println("Name: Ayman Amjad");
        solver.takeInput();
        solver.calculateRoots();
    }
}
```

OUTPUT:



Program 2:

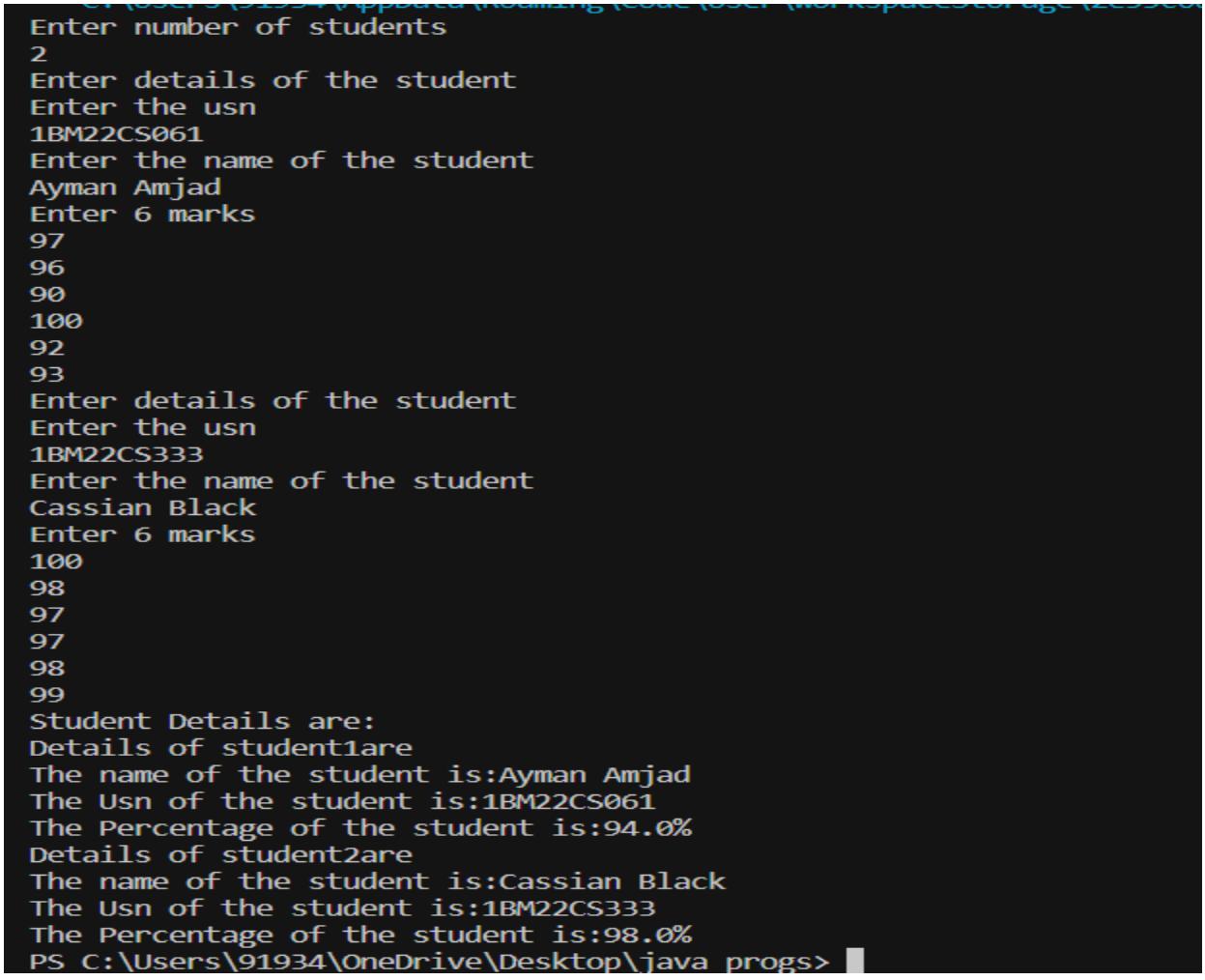
Write a Java program to create a class Student with members USN, name, marks(6 subjects). Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object to be created)

CODE:

```
import java.util.Scanner;

class Student {
    String name, usn;
    int[] marks = new int[6];
    Scanner sc = new Scanner(System.in);

    void acceptdetails() {
        System.out.println("Enter the usn");
        usn = sc.nextLine();
        System.out.println("Enter the name of the student");
        name = sc.nextLine();
        System.out.println("Enter 6 marks");
        for (int i = 0; i < 6; i++) {
            marks[i] = sc.nextInt();
        }
    }

    void calculate() {
        int sum = 0;
        double percentage;
        for (int i = 0; i < 6; i++) {
            sum = sum + marks[i];
        }
        percentage = sum / 6;
        System.out.println("The name of the student is:" + name);
        System.out.println("The Usn of the student is:" + usn);
        System.out.println("The Percentage of the student is:" + percentage + "%");
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of students");
        int stu = sc.nextInt();
        Student[] students = new Student[stu];
```

```
        for (int i = 0; i < stu; i++) {
            students[i] = new Student();
            System.out.println("Enter details of the student");
            students[i].acceptdetails();
        }
        System.out.println("Student Details are:");
        for (int i = 0; i < stu; i++) {
            System.out.println("Details of student" + (i + 1) + "are");
            students[i].calculate();
        }
        sc.close();
    }
}
```

OUTPUT:



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.

CODE:

```
import java.util.Scanner;

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

    // Constructor to set values for the members
    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    // Method to set details of the book
    public void setDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter book name: ");
        this.name = scanner.nextLine();

        System.out.println("Enter author name: ");
        this.author = scanner.nextLine();

        System.out.println("Enter price: ");
        this.price = scanner.nextDouble();

        System.out.println("Enter number of pages: ");
        this.numPages = scanner.nextInt();
    }
```

```
    // Method to get details of the book
    public void getDetails() {
        System.out.println("Book Name: " + name);
        System.out.println("Author: " + author);
        System.out.println("Price: $" + price);
        System.out.println("Number of Pages: " + numPages);
    }

    // toString method to display complete details of the book
    public String toString() {
        return "Book Details:\n" +
                "Name: " + name + "\n" +
                "Author: " + author + "\n" +
                "Price: $" + price + "\n" +
                "Number of Pages: " + numPages;
    }
}

public class prog3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("USN:1BM22CS061");
        System.out.println("Name: Ayman Amjad");
        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();

        // Creating an array to store n book objects
        Book[] books = new Book[n];

        // Creating n book objects and setting their details
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            books[i] = new Book("", "", 0.0, 0);
            books[i].setDetails();
        }

        // Displaying details of all the books
        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":");
            books[i].getDetails();
        }

        // Using the toString method to display complete details of the books
```

```
        System.out.println("\nComplete details of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toString());
        }
    }
}
```

OUTPUT:

```
USN:1BM22CS061
Name: Ayman Amjad
Enter the number of books: 2

Enter details for Book 1:
Enter book name:
AA
Enter author name:
Quro
Enter price:
345.66
Enter number of pages:
400

Enter details for Book 2:
Enter book name:
BB
Enter author name:
Buto
Enter price:
456
Enter number of pages:
234

Details of all books:

Book 1:
Book Name: AA
Author: Quro
Price: $345.66
Number of Pages: 400
```

```
Book 2:
Book Name: BB
Author: Buto
Price: $456.0
Number of Pages: 234

Complete details of all books:

Book 1:
Book Details:
Name: AA
Author: Quro
Price: $345.66
Number of Pages: 400

Book 2:
Book Details:
Name: BB
Author: Buto
Price: $456.0
Number of Pages: 234
```

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 contain only the method printArea( ) that prints the area of the given shape.

CODE:

```
abstract class Shape {
    protected int length;
    protected int width;

    public Shape(int length, int width) {
        this.length = length;
        this.width = width;
    }

    public abstract void printArea();
}

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

    @Override
    public void printArea() {
        int area = length * width;
        System.out.println("Rectangle Area: " + area);
    }
}

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

    @Override
    public void printArea() {
        double area = 0.5 * length * width;
        System.out.println("Triangle Area: " + area);
    }
}
```

```
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0); // Using width to represent radius
    }

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

public class prog4 {
    public static void main(String[] args) {
        System.out.println("USN:1BM22CS061");
        System.out.println("Name: Ayman Amjad");
        Rectangle rectangle = new Rectangle(5, 10);
        rectangle.printArea();

        Triangle triangle = new Triangle(4, 6);
        triangle.printArea();

        Circle circle = new Circle(3);
        circle.printArea();
    }
}
```

OUTPUT:

```
USN:1BM22CS061
Name: Ayman Amjad
Rectangle Area: 50
Triangle Area: 12.0
Circle Area: 28.274333882308138
```

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.

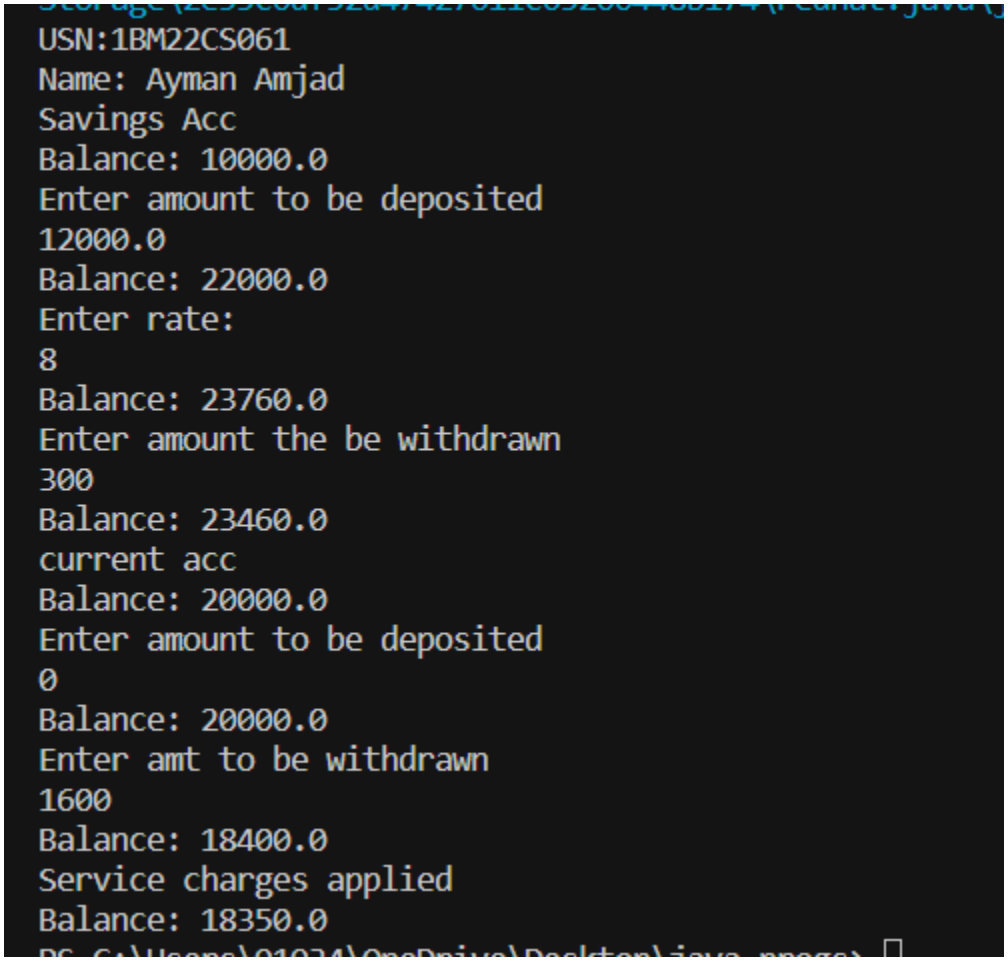
CODE:

```
import java.util.Scanner;
class account{
   boolean check;
   String name;
   long accno;
   String acctype;
   double bal;
   Scanner sc = new Scanner(System.in);
   account(String name, long accno, String acctype, double bal, boolean cheque)
   {
      this.name = name;
      this.accno=accno;
      this.acctype= acctype;
      this.bal= bal;
      this.check = cheque;
   }

   void Dep (){
      System.out.println("Enter amount to be deposited");
      double amt=sc.nextDouble();
      bal += amt;
   }
   void Displaybal() {
      System.out.println("Balance: " + bal);
   }
   public void interest()
   {
```

```
        System.out.println("Enter rate: ");
        Double rate=sc.nextDouble();
        if ("Savings". equals (acctype))
        {
            double interest=bal*(rate/100);
            bal += interest;
        }
    }
    public void withdraw () {
        System.out.println("Enter amount the be withdrawn");
        double amt=sc.nextDouble();
        if (amt <= bal){
            bal-=amt;
        }
        else{
            System.out.println("Insufficient funds");
        }
    }
}

class savacc extends account{
    public savacc(String name, long accno, double bal, boolean cheque)
    {
        super(name, accno, "Savings", bal, cheque);
 }
}

class curracct extends account{
    double minbal;
    double serv;
    public curracct(String name, long accno, double bal, double minbal, double serv, boolean
cheque)
    {
        super(name, accno, "current", bal, cheque);
        this.minbal=minbal;
        this.serv=serv;
    }
    public void withdraw() {
        System.out.println("Enter amt to be withdrawn");
        double amt=sc.nextDouble();
        if (amt<= bal-minbal) {
            bal-=amt;
        }
        else{
```

```
            System.out.println("Insufficient funds");
        }
    }
    public void check() {
        if (bal > minbal) {
            bal -= serv;
            System.out.println("Service charges applied");
        }
    }
}

class bank{

    public static void main(String[] args) {
        System.out.println("USN:1BM22CS061");
        System.out.println("Name: Ayman Amjad");
        savacc s = new savacc ("joseph",123456,10000.0,false);
        curracct c= new curracct ("John Mike",938462,20000.0,500.0, 50.0,true);
        System.out.println("Savings Acc");
        s.Displaybal();
        s.Dep();
        s.Displaybal();
        s.interest();
        s.Displaybal();
        s.withdraw();
        s.Displaybal();
        System.out.println("current acc");
        c.Displaybal();
        c.Dep();
        c.Displaybal();
        c.withdraw();
        c.Displaybal();
        c.check();
        c.Displaybal();
    }
}
```

OUTPUT:


USN:1BM22CS061
Name: Ayman Amjad
Savings Acc
Balance: 10000.0
Enter amount to be deposited
12000.0
Balance: 22000.0
Enter rate:
8
Balance: 23760.0
Enter amount the be withdrawn
300
Balance: 23460.0
current acc
Balance: 20000.0
Enter amount to be deposited
0
Balance: 20000.0
Enter amt to be withdrawn
1600
Balance: 18400.0
Service charges applied
Balance: 18350.0

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 student {
    public String usn;
    public String name;
    public int sem;
    public student(String usn, String name, int sem) {

        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

package CIE;

public class internal extends student {
    public double[] internalMarks=new double[5];
    internal(String usn, String name, int sem, double[] internalMarks)
    {
        super(usn,name,sem);
        this.internalMarks = internalMarks;

    }
}

package SEE;
import CIE.student;

public class external extends student{
    public double[] externalMarks;
    public external(String usn, String name, int sem, double[] externalMarks) {
        super(usn, name, sem);
        this.externalMarks = externalMarks;
```

```
    }
}

import CIE.student;
import CIE.internal;
import SEE.external;

import java.util.Scanner;
public class result {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();
        student[] students = new student[n];
        double[][] result = new double[n][5];
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Student " + (i + 1) + ":");
            scanner.nextLine();
            System.out.print("USN: ");
            String usn = scanner.nextLine();
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();
            System.out.println("Enter Internal marks for Student " + (i + 1) +":");
            double[] internalMarks = new double[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Internal marks for Course " + (j + 1) + ": ");
                internalMarks[j] = scanner.nextInt();
            }
            System.out.println("Enter External marks for Student " + (i + 1) +":");
            double[] externalMarks = new double[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("External marks for Course " + (j + 1) + ": ");
                externalMarks[j] = scanner.nextInt();
            }
            students[i] = new external(usn, name, sem, externalMarks);
            for (int j = 0; j < 5; j++) {
                result[i][j] = internalMarks[j] + (externalMarks[j]/2);
            }
        }
        System.out.println("\nFinal Marks of all Students:");
        for (int i = 0; i < n; i++) {
```

```
        System.out.println("\nStudent " + (i + 1) + " - USN: " + students[i].usn + ", Name: " +
students[i].name + ", Semester: " + students[i].sem);
        System.out.println("Final Marks:");
        for (int j = 0; j < 5; j++) {
            System.out.println("Course " + (j + 1) + ": " + result[i][j]);
        }
    }
    scanner.close();
  }
}
```

OUTPUT:

```
Enter the number of students: 2

Enter details for Student 1:
USN: 1bm22cs061
Name: Ayman Amjad
Semester: 3
Enter Internal marks for Student 1:
Internal marks for Course 1: 45
Internal marks for Course 2: 46
Internal marks for Course 3: 42
Internal marks for Course 4: 49
Internal marks for Course 5: 40
Enter External marks for Student 1:
External marks for Course 1: 90
External marks for Course 2: 94
External marks for Course 3: 96
External marks for Course 4: 93
External marks for Course 5: 99

Enter details for Student 2:
USN: 1bm22cs333
Name: Cassian Black
Semester: 3
Enter Internal marks for Student 2:
Internal marks for Course 1: 49
Internal marks for Course 2: 48
Internal marks for Course 3: 50
Internal marks for Course 4: 48
Internal marks for Course 5: 49
Enter External marks for Student 2:
External marks for Course 1: 99
External marks for Course 2: 100
External marks for Course 3: 100
External marks for Course 4: 99
External marks for Course 5: 97
```

```
Final Marks of all Students:

Student 1 - USN: 1bm22cs061, Name: Ayman Amjad, Semester: 3
Final Marks:
Course 1: 90.0
Course 2: 93.0
Course 3: 90.0
Course 4: 95.5
Course 5: 89.5

Student 2 - USN: 1bm22cs333, Name: Cassian Black, Semester: 3
Final Marks:
Course 1: 98.5
Course 2: 98.0
Course 3: 100.0
Course 4: 97.5
Course 5: 97.5
```

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() {
        super("Age cannot be negative");
    }
}

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge, IllegalArgumentException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new IllegalArgumentException("Son's age must be less than father's age");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
```

```
    }
}

public class prog7{
    public static void main(String[] args) {
        System.out.println("USN:1BM22CS061");
        System.out.println("Name: Ayman Amjad");
        try {
            Father father = new Father(50);
            Son son = new Son(50, 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());
        } catch (IllegalArgumentException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

OUTPUT:

```
' 'C:\Users\91934\AppData\Roaming\Code\User\worksp
USN:1BM22CS061
Name: Ayman Amjad
Father's age: 50
Son's age: 25
PS C:\Users\91934\OneDrive\Desktop\java progs>
```

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 DisplayMessage extends Thread {
    private String message;
    private int interval;

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

    @Override
    public void run() {
            try {
                while (!Thread.interrupted()) {
                    System.out.println(message);
                    Thread.sleep(interval);
            } }catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

public class hello{
    public static void main(String[] args) {
        System.out.println("USN:1BM22CS061");
        System.out.println("Name: Ayman Amjad");
        DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10000);
        DisplayMessage thread2 = new DisplayMessage("CSE", 2000);

        thread1.start();
        thread2.start();
        try{
            Thread.sleep(20000);
            thread1.interrupt();
            thread2.interrupt();

        } catch(InterruptedException e){
```

```
            e.printStackTrace();
        }
    }
}
```

OUTPUT:

```
PS C:\Users\91934\OneDrive\Desktop\java progs>  & 'C:\Prograr
Storage\2e55c0df52d47427611c65200448b174\redhat.java\jdt_ws\j
USN:1BM22CS061
Name: Ayman Amjad
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
java.lang.InterruptedException: sleep interrupted
        at java.lang.Thread.sleep(Native Method)
        at DisplayMessage.run(hello.java:15)
java.lang.InterruptedException: sleep interrupted
        at java.lang.Thread.sleep(Native Method)
        at DisplayMessage.run(hello.java:15)
PS C:\Users\91934\OneDrive\Desktop\java progs>
```