# SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

**Fall Semester, 2020-21**

**CSE1007 – Java Programming Lab**

**Digital Assingment-2**

Date: 20/08/2020

19BCE0758

R Narayan

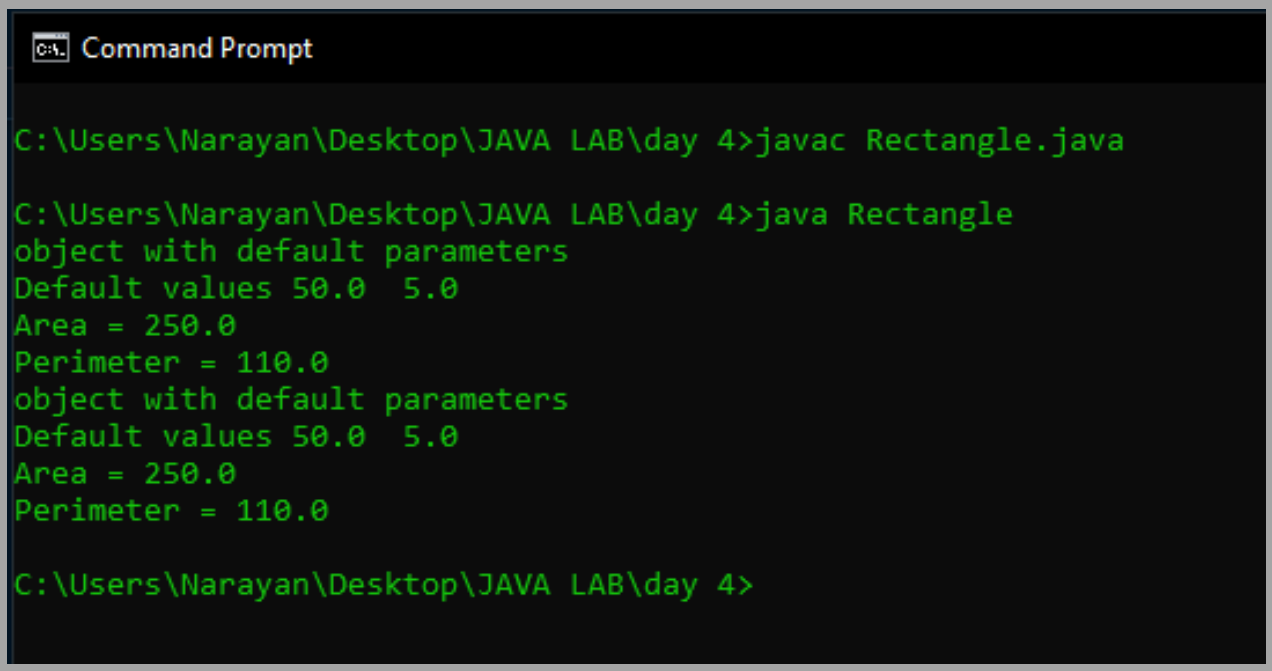
1. Design a class named Rectangle to represent a rectangle. The class contains: Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.
   (i)A default constructor that creates a default rectangle.
   (ii)A constructor that creates a rectangle with the specified width and height.
   (iii)A method named getArea() that returns the area of this rectangle.
   (iv)A method named getPerimeter() that returns the perimeter. Implement the class. Write a test program that creates two Rectangle objects— one with width 5 and height 50 and the other with width 2.5 and height 45.7. Display the width, height, area, and perimeter of each rectangle in this order.

   Code:

```java
//19BCE0758
//R Narayan
import java.util.Scanner;

public class Rectangle {

    private double width;
    private double height;

    public Rectangle() {
        // Default height = 10 , width = 5
        this.width = 5;
        this.height = 50;
    }

    public Rectangle(double w, double h) {
        this.width = w;
        this.height = h;
    }

    public double getArea() {
        return width * height;
```

```
    }

    public double getPerimeter() {
        return 2 * width + 2 * height;
    }

    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        Rectangle rectangle = new Rectangle();
        System.out.println("object with default parameters ");
        System.out.println("Default values " + rectangle.height + "  " + recta
ngle.width);
        System.out.println("Area = " + rectangle.getArea());
        System.out.println("Perimeter = " + rectangle.getPerimeter());
        Rectangle rectangle2 = new Rectangle(2.5, 45.7);
        System.out.println("object with manually passed  parameters ");
        System.out.println("Default values " + rectangle2.height + "  " + rect
angle2.width);
        System.out.println("Area = " + rectangle2.getArea());
        System.out.println("Perimeter = " + rectangle2.getPerimeter());

    }

}
```

Output:



2. Write a Java program to create a class called Student having data members Regno, Name, Course being studied and current CGPA. Include constructor to initialize objects. Create array of objects with at least 10 students and find 9- pointers.
   Code:

```java
//19BCE0758
//R Narayan
import java.util.Scanner;

public class Cgpa {
    private String reg;
    private String name;
    private String Course;
    private float cgp;

    public Cgpa( String r,String n, String c, float cg) {
        this.reg = r;
        this.name = n;
        this.Course = c;
        this.cgp = cg;
    }

    public boolean isNine() {
        if (this.cgp >= 9)
            return true;
        else
            return false;
    }

    public void nameP() {
        System.out.println(this.name);
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Cgpa[] students = new Cgpa[10];
        System.out.println("Enter the number of students ");
        int n = s.nextInt();
        for (int i = 0; i < n; i++) {
            s.nextLine();
            System.out.println("Enter the Reg No");
            String r = s.nextLine();
            System.out.println("Enter the Name");
            String na = s.nextLine();
```

```
            System.out.println("Enter the Course");
            String c = s.nextLine();
            System.out.println("Enter the CGPA");
            float cg = s.nextFloat();
            students[i] = new Cgpa(r, na, c, cg);
        }
        System.out.println("List of Nine Pointers");
        for (int i = 0; i < n; i++) {
            if (students[i].isNine()) {
                students[i].nameP();
            }
        }

    }
}
```

Output:

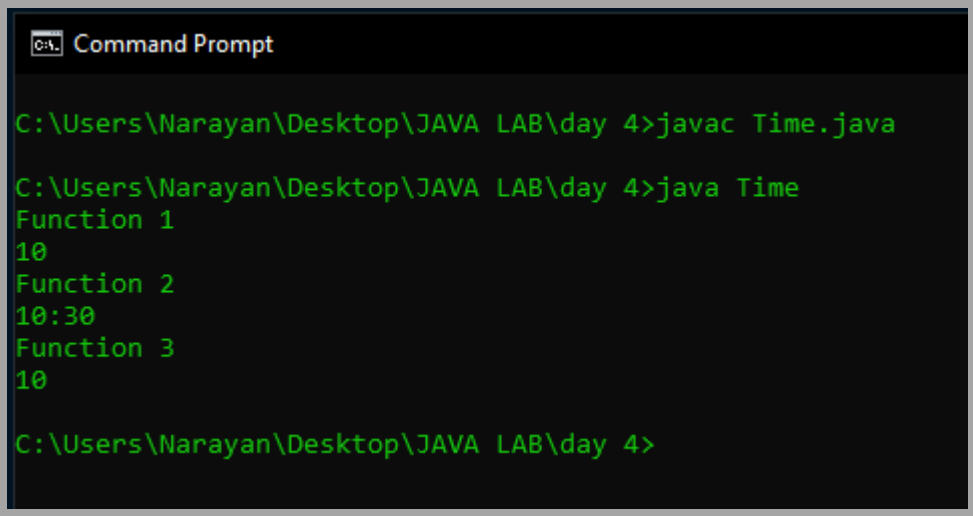
```
Command Prompt
C:\Users\Narayan\Desktop\JAVA LAB\day 4>javac Cgpa.java

C:\Users\Narayan\Desktop\JAVA LAB\day 4>java Cgpa
Enter the number of students
3
Enter the Reg No
19BCE0001
Enter the Name
Abc
Enter the Course
Computer Science
Enter the CGPA
10
Enter the Reg No
19BMI0001
Enter the Name
def
Enter the Course
mechanical
Enter the CGPA
9
Enter the Reg No
19BEC0001
Enter the Name
ghi
Enter the Course
electrical
Enter the CGPA
8
List of Nine Pointers
Abc
```

3. Write a Java program that displays that displays the time in different formats in the form of HH,MM,SS using constructor Overloading.

Code:

```java
//19BCE0758
//R Narayan
import java.util.Scanner;
public class Time {
    public void disp(String hh) {
        System.out.println(hh);
    }

    public void disp(String hh, String mm) {
        System.out.println(hh + ":" + mm);
    }

    public void disp(String hh, String mm, String ss) {
        System.out.println(hh);
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Time t = new Time();
        System.out.println("Function 1");
        t.disp("10");
        System.out.println("Function 2");
        t.disp("10", "30");
        System.out.println("Function 3");
        t.disp("10", "30", "55");
    }
}
```

Output:

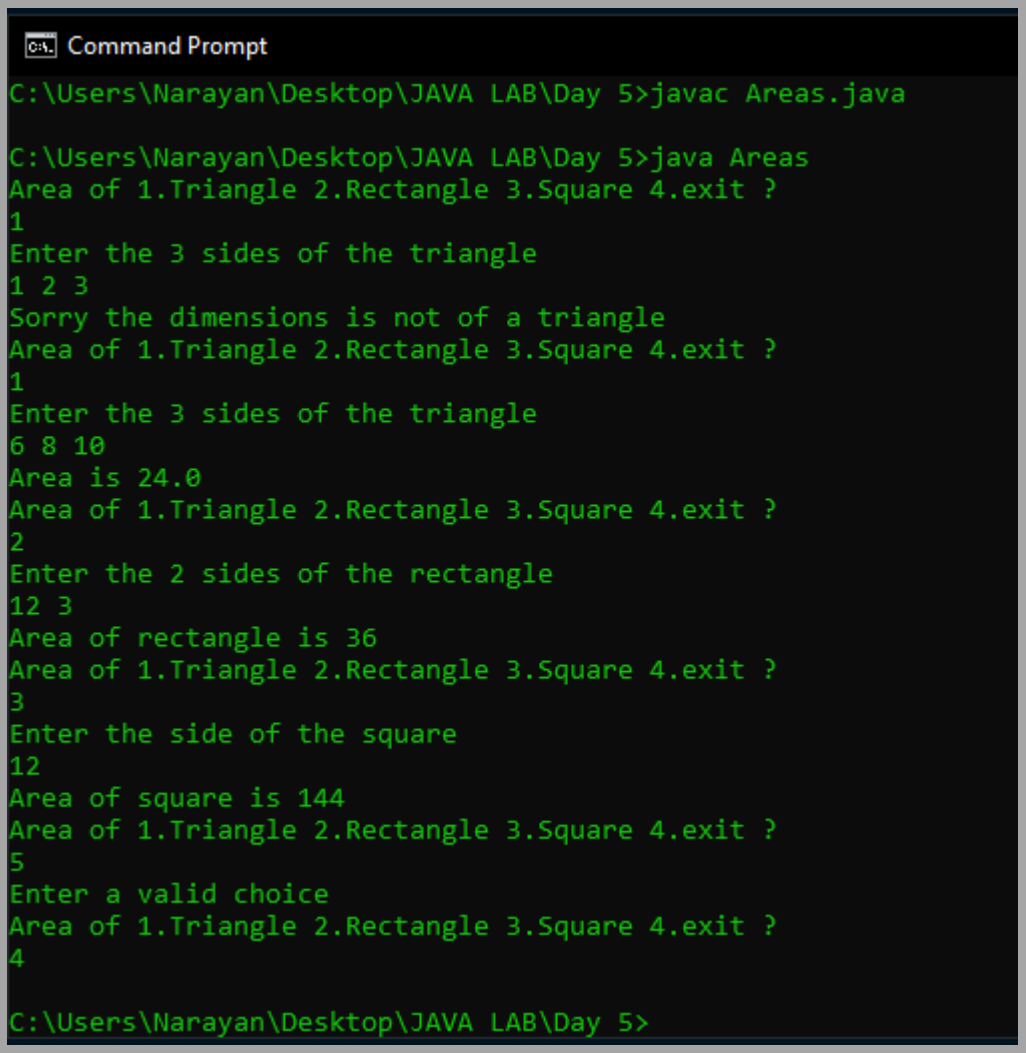



4. Write a Java program that displays area of different Figures (Rectangle, Square, Triangle) using the method overloading.
   Code:

```
//19BCE0758
//R Narayan
import java.lang.Math;
import java.util.Scanner;

class Area {
    public void retArea(int l, int b) {
        System.out.println("Area of rectangle is " + l * b);
    }

    public void retArea(int s) {
        System.out.println("Area of square is " + s * s);
    }

    public void retArea(int a, int b, int c) {
        if (a + b > c && b + c > 5 && c + a > b) {
            float s = (a + b + c) / 2;
            System.out.println("Area is " + Math.sqrt((s * (s - a) * (s - b) *
 (s - c))));
        } else {
            System.out.println("Sorry the dimensions is not of a triangle");
        }

    }
}

class Areas {
    public static void main(String args[]) {
        int n = 1;
        Scanner sc = new Scanner(System.in);
```

```
        Area ob = new Area();
        do {
            System.out.println("Area of 1.Triangle 2.Rectangle 3.Square 4.exit
?");
            n = sc.nextInt();
            if (n == 1) {
                System.out.println("Enter the 3 sides of the triangle");
                int a, b, c;
                a = sc.nextInt();
                b = sc.nextInt();
                c = sc.nextInt();
                ob.retArea(a, b, c);
            } else if (n == 2) {
                System.out.println("Enter the 2 sides of the rectangle");
                int a, b;
                a = sc.nextInt();
                b = sc.nextInt();
                ob.retArea(a, b);

            } else if (n == 3) {
                System.out.println("Enter the side of the square");
                int a;
                a = sc.nextInt();
                ob.retArea(a);

            } else if (n != 4) {
                System.out.println("Enter a valid choice");
            }
        } while (n != 4);
    }
}
```

Output:

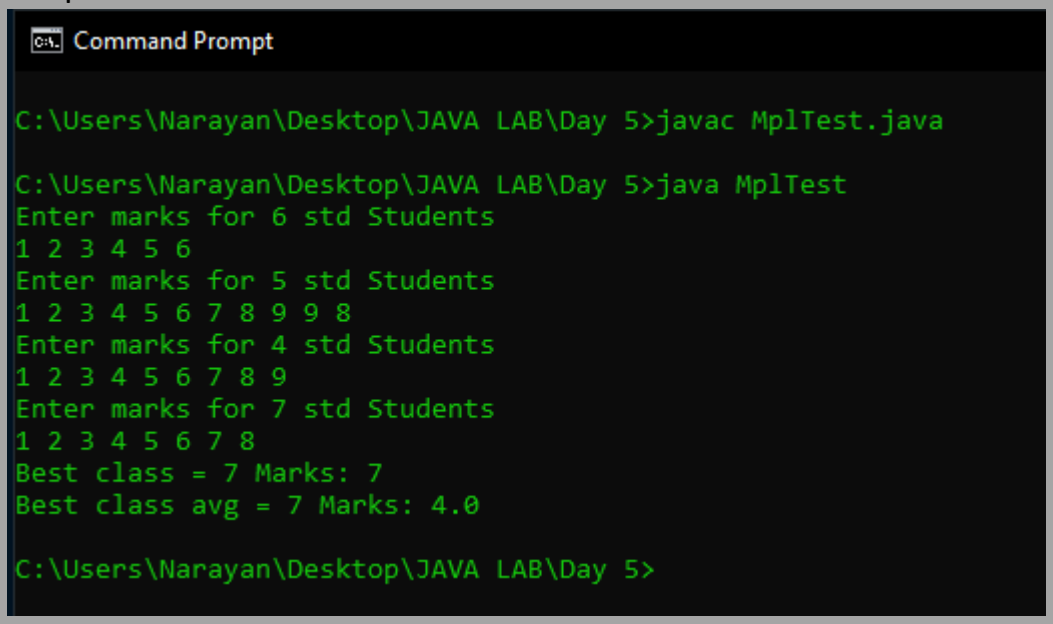

Command Prompt
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>javac Areas.java
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>java Areas
Area of 1.Triangle 2.Rectangle 3.Square 4.exit ?
1
Enter the 3 sides of the triangle
1 2 3
Sorry the dimensions is not of a triangle
Area of 1.Triangle 2.Rectangle 3.Square 4.exit ?
1
Enter the 3 sides of the triangle
6 8 10
Area is 24.0
Area of 1.Triangle 2.Rectangle 3.Square 4.exit ?
2
Enter the 2 sides of the rectangle
12 3
Area of rectangle is 36
Area of 1.Triangle 2.Rectangle 3.Square 4.exit ?
3
Enter the side of the square
12
Area of square is 144
Area of 1.Triangle 2.Rectangle 3.Square 4.exit ?
5
Enter a valid choice
Area of 1.Triangle 2.Rectangle 3.Square 4.exit ?
4
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>

5. In a school, students of all classes from std I to X appear for the MathPremierLeague examination. Define a class MPL which stores the details of the marks scored by each class. It should contain the following 4 data members: Standard, number of students, marks[] array to store the scores of all the students of the class in MPL exam. Define a parameterized constructor which receives the values for the first two data members from the main() method. Create a Form within the constructor, read the marks of all students and hence find the first mark. Define a method findBestClass() to display the standard which has secured the highest mark. Overload this method to display the standard with the highest class average. The marks array should be declared dynamically based on the strength of the class. Code:

```
// 19BCE0758
// R NARAYAN
import java.util.Scanner;

class MPL {
    int nStudents;
    int[] marks = new int[10];
    int std;
    int max = -1;

    MPL(int s, int ns) {
        this.std = s;
        this.nStudents = ns;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter marks for " + ns + " std Students");
        for (int i = 0; i < ns; i++) {
            this.marks[i] = sc.nextInt();
            if (this.marks[i] > max) {
                max = marks[i];
            }
        }
    }

    float getAvg() {
        int s = 0;
        for (int i = 0; i < this.nStudents; i++) {
            s += this.marks[i];
        }
        return s / this.nStudents;
    }
}

public class MplTest {
    public static void main(String args[]) {
        MPL[] m = new MPL[4];
        m[0] = new MPL(5, 6);
        m[1] = new MPL(10, 5);
```

```
        m[2] = new MPL(8, 4);
        m[3] = new MPL(7, 7);
        bestClass(m);
        bestClass(m, 1);

    }

    static void bestClass(MPL[] m) {
        int max = 0;
        for (int i = 0; i < 4; i++) {
            if (m[i].max > m[max].max) {
                max = i;
            }
        }
        System.out.println("Best class = " + m[max].std + " Marks: " + m[max].
max);
    }

    static void bestClass(MPL[] m, int a) {
        int max = 0;
        for (int i = 0; i < 4; i++) {
            if (m[i].getAvg() > m[max].getAvg()) {
                max = i;
            }
        }
        System.out.println("Best class avg = " + m[max].std + " Marks: " + m[m
ax].getAvg());

    }

}
```

Output:



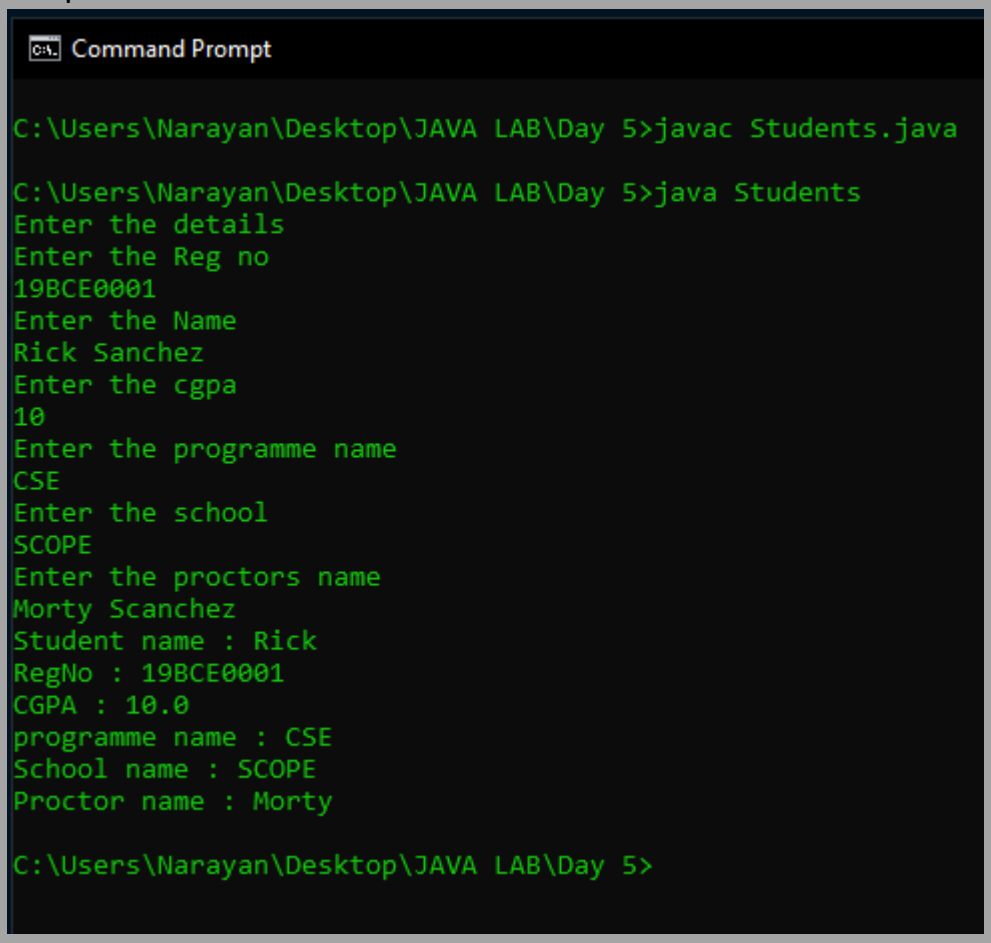
6. Read the following details of ‘n’ students using Scanner class methods and display the same.
   - Registration number ( String)
   - Name (String that may contain first name, middle name and last name)
   - CGPA (Floating point number)
   - Programme Name(String)
   - School Name (String with multiple words)
   - Proctor Name (String that may contain first, middle and last names)

   Code:

```
// 19BCE0758
// R NARAYAN
import java.util.*;

public class Students {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        String reg = new String();
        String name = new String();
        float cgpa;
        String pname = new String();
        String Sname = new String();
        String Proctorname = new String();
        System.out.println("Enter the details");
        System.out.println("Enter the Reg no");
        reg = sc.next();
        sc.nextLine();
        System.out.println("Enter the Name ");

        name = sc.next();
        sc.nextLine();
        System.out.println("Enter the cgpa ");

        cgpa = sc.nextFloat();
        System.out.println("Enter the programme name ");

        pname = sc.next();
        sc.nextLine();
        System.out.println("Enter the school");

        Sname = sc.next();
        sc.nextLine();
        System.out.println("Enter the proctors name");

        Proctorname = sc.next();
        sc.nextLine();
        System.out.println("Student name : " + name + "\nRegNo : " + reg + "\n
CGPA : " + cgpa + "\nprogramme name : "
```

```
                + pname + "\nSchool name : " + Sname + "\nProctor name : " + P
roctorname);
    }
}
```

Output:



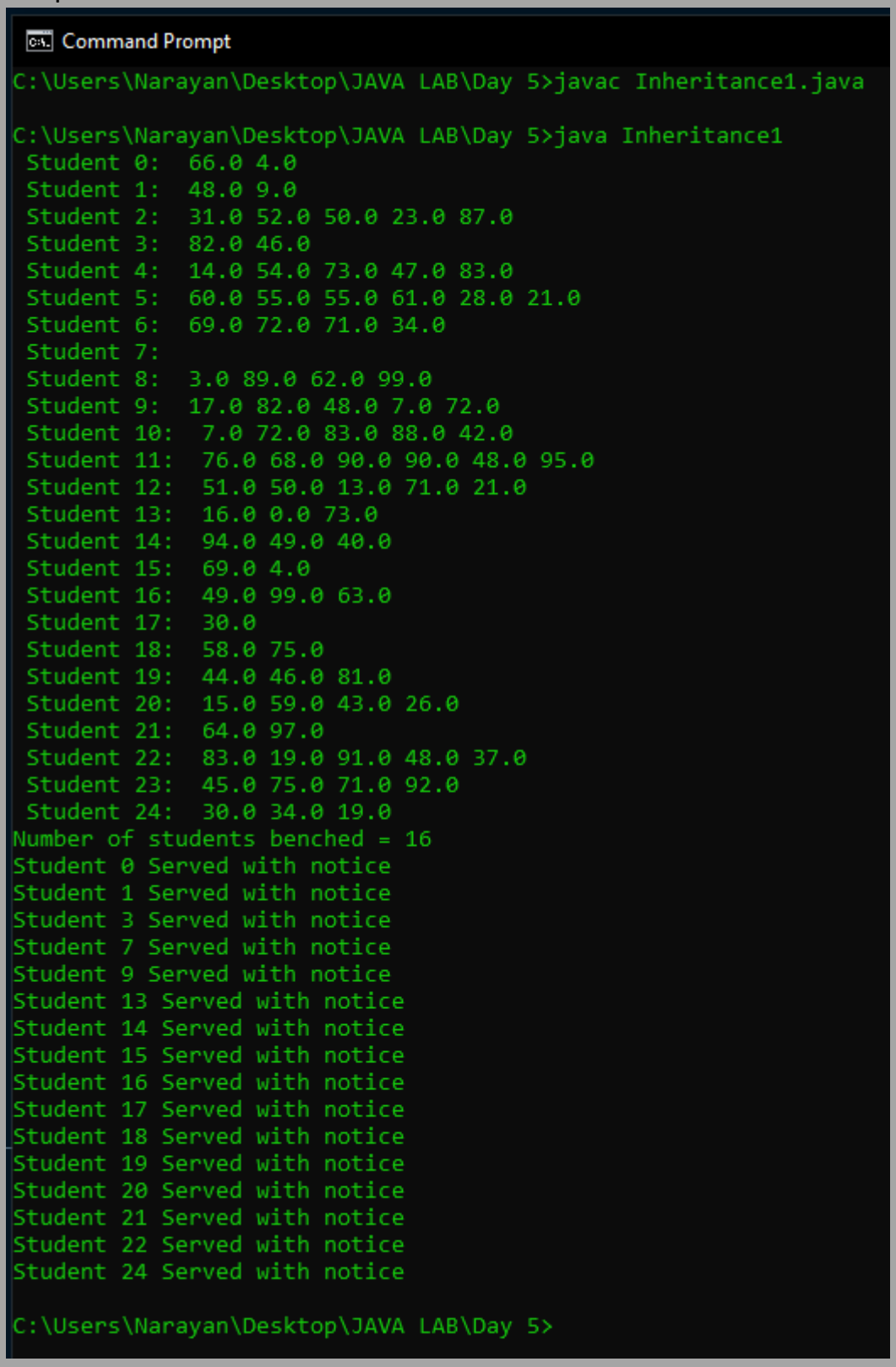
7. A training centre conducts a total of 7 tests for its students. Students are allowed to skip few tests. Let there be 25 students in the batch. So in the main class for every student, read the number of tests taken and the marks scored in each test. A class 'TestDetails' should be defined with a 2D array of float type to store the marks of all the students. Define a method 'storeMarks()' that will receive the following details for every student from the main class and create in the 2D array, those many columns equal to the number of tests, so as to store the marks. There is no need to store the number of tests. Define another method 'displayMarks()' to print the details. Also the training centre wishes to keep those students in notice period who have taken < 3 tests and those who have not scored ≥ 50 in at least 3 tests. Derive another class 'NoticePeriod' from 'TestDetails' that includes a method to count and print the number of students in bench. Also it should print the ID of those students assuming the row index of the array to be their ID. While checking do not proceed to check the marks in all tests, if the student has already scored more than 50 in 3 tests. Instantiate this class from the main class and do the required processing.
Code:

```java
// 19BCE0758
// R NARAYAN
import java.util.Random;

class TestDetails {
    final int m = 25;
    final int n = 7;
    public float[][] marks = new float[m][n];

    public void storeMarks() {
        Random rand = new Random();
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                boolean isTest = (rand.nextInt(2) == 1) ? true : false;
                if (isTest) {
                    marks[i][j] = rand.nextInt(101);
                } else
                    marks[i][j] = -1;
            }
        }
    }

    public void displayMarks() {
        for (int i = 0; i < m; i++) {
            System.out.print(" Student " + i + ":  ");
            for (int j = 0; j < n; j++) {
                if (marks[i][j] != -1) {
                    System.out.print(marks[i][j] + " ");
                }
            }
            System.out.println();
```

```
        }
    }
}

class NoticePeriod extends TestDetails {
    boolean[] notice = new boolean[m];

    void calcNotics() {
        int studentCount = 0;
        for (int i = 0; i < m; i++) {
            int testCount = 0;
            int marksCount = 0;
            for (int j = 0; j < n && marksCount < 3; j++) {
                if (marks[i][j] != -1) {
                    testCount++;
                }
                if (marks[i][j] >= 50) {
                    marksCount++;
                }

            }
            if (testCount < 3 || marksCount < 3) {
                notice[i] = true;
                studentCount++;
            } else {
                notice[i] = false;
            }
        }
        System.out.println("Number of students benched = " + studentCount);
    }

    void displayNotice() {
        for (int i = 0; i < m; i++) {
            if (notice[i]) {
                System.out.println("Student " + i + " Served with notice");
            }
        }
    }

}

public class Inheritance1 {
    public static void main(String args[]) {
        NoticePeriod n = new NoticePeriod();
        n.storeMarks();
        n.displayMarks();
        n.calcNotics();
        n.displayNotice();
```

```
    }
}
```

Output:


Command Prompt
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>javac Inheritance1.java
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>java Inheritance1
Student 0: 66.0 4.0
Student 1: 48.0 9.0
Student 2: 31.0 52.0 50.0 23.0 87.0
Student 3: 82.0 46.0
Student 4: 14.0 54.0 73.0 47.0 83.0
Student 5: 60.0 55.0 55.0 61.0 28.0 21.0
Student 6: 69.0 72.0 71.0 34.0
Student 7:
Student 8: 3.0 89.0 62.0 99.0
Student 9: 17.0 82.0 48.0 7.0 72.0
Student 10: 7.0 72.0 83.0 88.0 42.0
Student 11: 76.0 68.0 90.0 90.0 48.0 95.0
Student 12: 51.0 50.0 13.0 71.0 21.0
Student 13: 16.0 0.0 73.0
Student 14: 94.0 49.0 40.0
Student 15: 69.0 4.0
Student 16: 49.0 99.0 63.0
Student 17: 30.0
Student 18: 58.0 75.0
Student 19: 44.0 46.0 81.0
Student 20: 15.0 59.0 43.0 26.0
Student 21: 64.0 97.0
Student 22: 83.0 19.0 91.0 48.0 37.0
Student 23: 45.0 75.0 71.0 92.0
Student 24: 30.0 34.0 19.0
Number of students benched = 16
Student 0 Served with notice
Student 1 Served with notice
Student 3 Served with notice
Student 7 Served with notice
Student 9 Served with notice
Student 13 Served with notice
Student 14 Served with notice
Student 15 Served with notice
Student 16 Served with notice
Student 17 Served with notice
Student 18 Served with notice
Student 19 Served with notice
Student 20 Served with notice
Student 21 Served with notice
Student 22 Served with notice
Student 24 Served with notice
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>

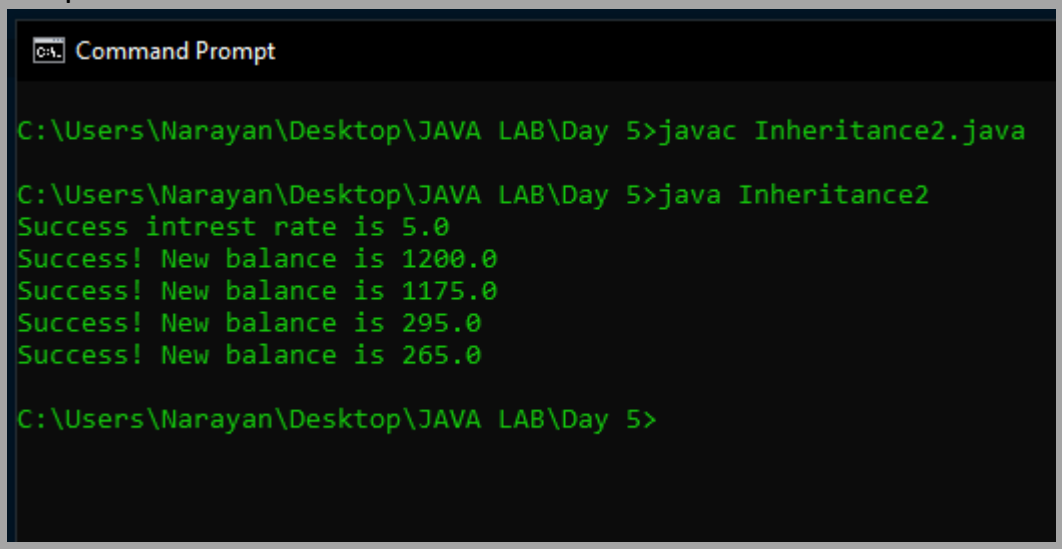
8. Create an inheritance hierarchy in java using following information given below that a bank might use to represent customers' bank accounts. Base class Account should include one data member of type double to represent account balance. The class should provide constructor that receives an initial balance and uses it to initialize the data member.The constructor should validate the initial balance to ensure that it is greater than or equal to 0.If not the balance is set to 0.0 and the constructor should display an error message,indicating that the initial balance was invalid. The class also provides three member functions credit, debit(debit amount should not exceed the account balance) and enquiry.Derived class SavingsAccount should inherit the functionality of an Account, but also include data member of type double indicating the interest rate assigned to the Account. SavingsAccount constructor should receive the initial balance, as well as an initial value for SavingsAccount's interest rate. SavingsAccount should provide public member function calculateInterest that returns double indicating the amount of interest earned by an account.The method calculateInterest should determine this amount by multiplying the interest rate by the account balance.SavingsAccount function should inherit member functions credit,debit and enquiry without redefining them. Derived class CheckingAccount should inherit the functionality of an Account, but also include data member of type double that represents the fee charged per transaction. CheckingAccount constructor should receive the initial balance, as well as parameter indicating fee amount. class CheckingAccount should redefine credit and debit function so that they subtract the fee from account balance whenever either transaction is performed. CheckingAccount's debit function should charge a fee only if the money is actually withdrawn (debit amount should not exceed the account balance).After defining the class hierarchy, write program that creates object of each class and tests their member functions. Add interest to SavingAccount object by first invoking its calculateInterest function, then passing the returned interest amount to object's credit function.
   Code:

```
// 19BCE0758
// R NARAYAN
import java.util.Scanner;

class Account {
    double balance;

    Account(double balance) {
        if (balance > 0) {
            this.balance = balance;
        } else {
            System.out.println("Invalid");
        }
    }

    void credit(double money) {
        this.balance += money;
        System.out.println("Success! New balance is " + this.balance);
    }
```

```
    void debit(double money) {
        if (money <= this.balance) {
            this.balance -= money;
            System.out.println("Success! New balance is " + this.balance);
        } else {
            System.out.println("Insufficient Funds");
        }
    }
}

class SavingsAccount extends Account {
    double intrestRate;

    SavingsAccount(double balance, double intrestRate) {
        super(balance);
        this.intrestRate = intrestRate;
    }

    public double calculateIntrest() {
        System.out.println("Success intrest rate is " + this.intrestRate);
        return this.intrestRate * this.balance;
    }
}

class checkingAccount extends Account {
    double transactionFee;

    checkingAccount(double bal, double tf) {
        super(bal);
        this.transactionFee = tf;
    }

    void credit(double money) {
        this.balance += money;
        this.balance -= this.transactionFee;
        System.out.println("Success! New balance is " + this.balance);
    }

    void debit(double money) {
        if (money <= this.balance + this.transactionFee) {
            this.balance -= money;
            this.balance -= this.transactionFee;

            System.out.println("Success! New balance is " + this.balance);
        } else {
            System.out.println("Insufficient Funds");
        }
```

```
    }

}

public class Inheritance2 {
    public static void main(String args[]) {
        SavingsAccount savings = new SavingsAccount(200, 5);
        savings.credit(savings.calculateIntrest());
        savings.debit(25);
        checkingAccount check = new checkingAccount(200, 5);
        check.credit(100);
        check.debit(25);
    }
}
```

Output:

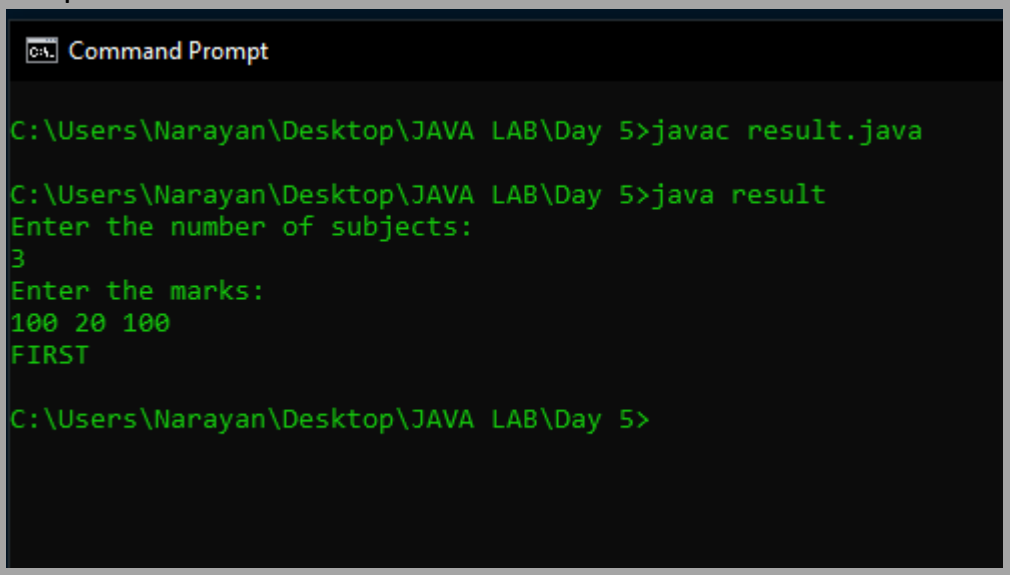


9. Write an interface called Exam with a method Pass ( ) that returns the total marks. Write another interface called Classify with a method Average (int total) which returns a string. Write a Class called Result which implements both Exam and Classify. The Pass method should get the marks from the user and finds the total marks and return. The Division method calculate the average marks and return “First” if the average is 60 or more, “SECOND” when average is 50 or more but below 60, “NO DIVISION” when average is less than 50
   Code:

```java
// 19BCE0758
// R NARAYAN
import java.util.Scanner;

interface Exam {
    int Pass();
}

interface Classify {
    String Average(int total);
}

public class result implements Exam, Classify {
    Scanner sc = new Scanner(System.in);
    public int n;
    static int tot = 0;

    public int Pass() {
        System.out.println("Enter the number of subjects:");
        n = sc.nextInt();
        System.out.println("Enter the marks:");
        for (int i = 0; i < n; i++) {
            tot += sc.nextInt();
        }
        return tot;
    }

    public String Average(int total) {
        int avg = total / n;
        if (avg > 60)
            return "FIRST";
        else if (avg > 50 & avg < 60)
            return "SECOND";
        else
            return "NO DIVISION";
    }

    public static void main(String[] args) {
        result res = new result();
```

```
        res.Pass();
        System.out.println(res.Average(tot));
    }
}
```

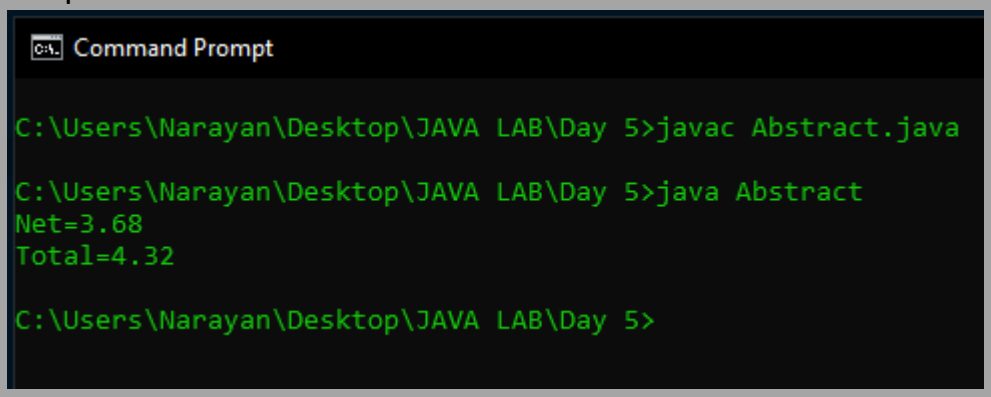
Output:



10. Write an abstract class special with an abstract method double Process (double P,double R). Create a subclass Discount and implement the Process() method with the following formula: net=P-P*R/100. Return the Process() method with the following formula: total=P+P*R/100. Return the total.
Code:

```
// 19BCE0758
// R NARAYAN
import java.util.Scanner;

abstract class Net {
    double p;
    double r;

    abstract void Process(double P, double R);
}

class Discount extends Net {
    public void Process(double x, double y) {
        p = x;
        r = y;
    }

    public double getdata() {
        double net = p - p * r / 100;
        return net;
    }
}

class Tax extends Discount {
    public double getdata() {
        double total = p + p * r / 100;
        return total;
    }
}

public class Abstract {
    public static void main(String[] args) {
        Discount o1 = new Discount();
        Tax o2 = new Tax();
        o1.Process(4, 8);
        o2.Process(4, 8);
        System.out.println("Net=" + o1.getdata());
        System.out.println("Total=" + o2.getdata());
    }
}
```

Output:


Command Prompt
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>javac Abstract.java
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>java Abstract
Net=3.68
Total=4.32
C:\Users\Narayan\Desktop\JAVA LAB\Day 5>