Aima khan
2023F-BSE-064

LAB # 2

Good practices of programming

OBJECTIVE

Implementing good code practices and code optimization techniques.

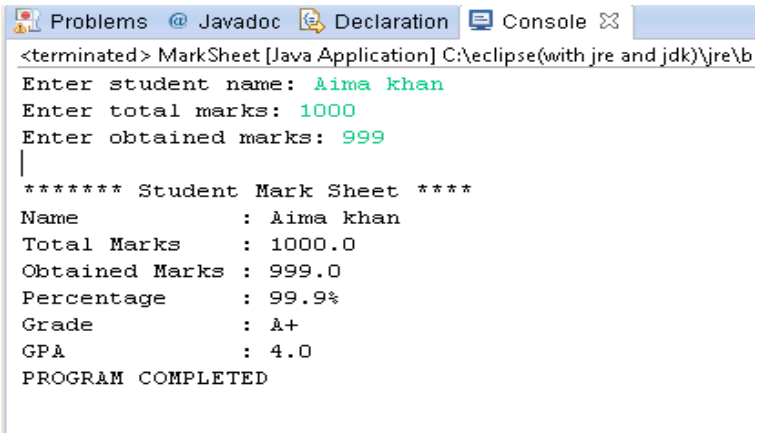
Lab Task:

1. Create a design for the mark sheet by taking runtime value of student name, total marks, obtained marks and calculate its percentage, grade and GPA. Use good practices of programming that we have studied and ensure that the outcomes should be presented in a proper Viewable approach.

CODE:

```
import java.util.Scanner;
public class MarkSheet {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter student name: ");
        String name = input.nextLine();
        System.out.print("Enter total marks: ");
        double totalMarks = input.nextDouble();
        System.out.print("Enter obtained marks: ");
        double obtainedMarks = input.nextDouble();
        double percentage = (obtainedMarks / totalMarks) * 100;
        String grade;
        double gpa;
        if (percentage >= 90) {
            grade = "A+";
            gpa = 4.0;
        } else if (percentage >= 80) {
            grade = "A";
            gpa = 3.7;
        } else if (percentage >= 70) {
            grade = "B";
            gpa = 3.0;
        } else if (percentage >= 60) {
            grade = "C";
            gpa = 2.0;
        } else if (percentage >= 50) {
            grade = "D";
            gpa = 1.0;
        } else {
            grade = "F";
            gpa = 0.0;
        }
         System.out.println("\n******* Student Mark Sheet ****");
         System.out.println("Name            : " + name);
         System.out.println("Total Marks     : " + totalMarks);
         System.out.println("Obtained Marks  : " + obtainedMarks);
         System.out.println("Percentage      : " + percentage + "%");
         System.out.println("Grade           : " + grade);
         System.out.println("GPA             : " + gpa);
         System.out.println("PROGRAM COMPLETED");
         input.close();
    }
}
```

OUTPUT:



2. Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide methods that calculate the rectangle's perimeter and area. It has set and get methods for both length and width. The set methods should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0. Write a program to test class Rectangle.

CODE:

```
class Rectangle {
    private double length;
    private double width;
    public Rectangle() {
        this.length = 1.0;
        this.width = 1.0;
    }
    public Rectangle(double length, double width) {
        setLength(length);
        setWidth(width);
    }
    public void setLength(double length) {
        if (length > 0.0 && length < 20.0) {
            this.length = length;
        }
    }
    public void setWidth(double width) {
        if (width > 0.0 && width < 20.0) {
            this.width = width;
        }
    }
    public double getLength() {
        return length;
    }
    public double getWidth() {
        return width;
    }
    public double getArea() {
        return length * width;
    }
    public double getPerimeter() {
        return 2 * (length + width);
    }
}
```

Aima khan
2023F-BSE-064

```
public class Main {
    public static void main(String[] args) {
        Rectangle rect1 = new Rectangle();
        Rectangle rect2 = new Rectangle(7.5, 4.3);
        System.out.println("Rect1 :Length: " + rect1.getLength() + ", Width: " + rect1.getWidth());
        System.out.println("Rect1 : Area: " + rect1.getArea() + ", Perimeter: " + rect1.getPerimeter(
        System.out.println("Rect2 : Length: " + rect2.getLength() + ", Width: " + rect2.getWidth());
        System.out.println("Rect2 : Area: " + rect2.getArea() + ", Perimeter: " + rect2.getPerimeter(
    }
}
```

OUTPUT:

```
<terminated> Main (3) [Java Application] C:\eclipse(with jre and jdk)
Rect1 :Length: 1.0, Width: 1.0
Rect1 : Area: 1.0, Perimeter: 4.0
Rect2 : Length: 7.5, Width: 4.3
Rect2 : Area: 32.25, Perimeter: 23.6
```