C-DAC Mumbai
Lab Assignment-2

Answers

Problem 1: Grade Evaluation System
Problem Statement: Write a Java program that calculates the average marks of a student and determines the grade based on the following criteria:
Grade A: Average marks ≥ 90
Grade B: Average marks between 70 and 89
Grade C: Average marks between 50 and 69 **Grade D**: Average marks between 30 and 49 **Fail**: Average marks < 30
Predefined Values (Try with different values as well):
- Marks in Maths = 80
- Marks in Science = 85
- Marks in History = 90

Expected Output:
Average Marks: 85
Grade: B

```java
public class GradeEvaluationSystem {
    Run | Debug
    public static void main(String[] args) {
        int maths = 80;
        int science = 85;
        int history = 90;

        double average = (maths + science + history) / 3.0;

        String grade;
        if (average >= 90) {
            grade = "A";
        } else if (average >= 70 && average <= 89) {
            grade = "B";
        } else if (average >= 50 && average <= 69) {
            grade = "C";
        } else if (average >= 30 && average <= 49) {
            grade = "D";
        } else {
            grade = "Fail";
        }

        System.out.println(x:"Marks:");
        System.out.println("Maths = " + maths);
        System.out.println("Science = " + science);
        System.out.println("History = " + history);
        System.out.println("Average Marks = " + average);
        System.out.println("Grade = " + grade);
    }
}
```

```
PS C:\Users\baenu\test> javac GradeEvaluationSystem.java
PS C:\Users\baenu\test> java GradeEvaluationSystem
Marks:
Maths = 80
Science = 85
History = 90
Average Marks = 85.0
Grade = B
```

Problem 2: Leap Year

Problem Statement: Write a Java program that checks whether the year is a leap year or not. A year is a leap year if:

It is divisible by 4, but not divisible by 100, **or** It is divisible by 400.

Predefined Value (Try with different values as well):

- Year = 2024
- Year = 1900

Expected Output:

2024 is a leap year.

1900 is not a leap year.

```java
public class LeapYearCheck {
    Run | Debug
    public static void main(String[] args) {
        int year1 = 2024;
        int year2 = 1900;

        checkLeapYear(year1);
        checkLeapYear(year2);
    }

    static void checkLeapYear(int year) {
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
    }
}
```

```
PS C:\Users\baenu\test> javac LeapYearCheck.java
PS C:\Users\baenu\test> java LeapYearCheck
2024 is a leap year.
1900 is not a leap year.
```

Problem 3: Days of the Week

Problem Statement: Write a Java program that takes an integer between 1 and 7 and prints the corresponding day of the week using a switch-case statement. If the input is outside the range of 1 to 7, the program should display **"Invalid day number".**

Predefined Value:

Day number = 3

Expected Output:

The day is Wednesday.

```java
public class DaysOfWeek {
    Run | Debug
    public static void main(String[] args) {
        int dayNumber = 3;

        switch (dayNumber) {
            case 1:
                System.out.println(x:"The day is Sunday.");
                break;
            case 2:
                System.out.println(x:"The day is Monday.");
                break;
            case 3:
                System.out.println(x:"The day is Wednesday.");
                break;
            case 4:
                System.out.println(x:"The day is Thursday.");
                break;
            case 5:
                System.out.println(x:"The day is Friday.");
                break;
            case 6:
                System.out.println(x:"The day is Saturday.");
                break;
            case 7:
                System.out.println(x:"The day is Tuesday.");
                break;
            default:
                System.out.println(x:"Invalid day number.");
        }
    }
}
```

```
PS C:\Users\baenu\test> javac DaysOfWeek.java
PS C:\Users\baenu\test> java DaysOfWeek
The day is Wednesday.
```

Problem 4: Identify the Values of Uninitialized Variables

Scenario: You are working on a program that handles different data types. Your manager has asked you to quickly check the values of various variables, but you're in a rush and forget to initialize them. As you go through the code, you expect some values to show up, but Java has something else in mind. Your task is to fix the issue and ensure the variables hold proper values.

Instructions:

1. Declare the following variables: byte a; short b; int c; long d; float e; double f; char g; boolean h;

2. Print out their values.

Issue: Local variables inside methods must be initialized before use, otherwise we get a compilation error. Instance variables declared inside class but outside method get default values.

```java
public class UninitializedVariables {
    Run | Debug
    public static void main(String[] args) {
        byte a = 0;
        short b = 0;
        int c = 0;
        long d = 0L;
        float e = 0.0f;
        double f = 0.0;
        char g = '\u0000';
        boolean h = false;

        System.out.println("byte a = " + a);
        System.out.println("short b = " + b);
        System.out.println("int c = " + c);
        System.out.println("long d = " + d);
        System.out.println("float e = " + e);
        System.out.println("double f = " + f);
        System.out.println("char g = '" + g + "'");
        System.out.println("boolean h = " + h);
    }
}
```

```
PS C:\Users\baenu\test> javac UninitializedVariables.java
PS C:\Users\baenu\test> java UninitializedVariables
byte a = 0
short b = 0
int c = 0
long d = 0
float e = 0.0
double f = 0.0
char g = ''
boolean h = false
```