CDAC Mumbai

Lab Assignment

Lab Assignment No – 02

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

Code:

```
public class GradeEvaluationSystem {

   public static void main(String[] args) {


      int maths_marks = 95;

      int science_marks = 97;

      int history_marks = 90;


      int average_of_marks = (maths_marks + science_marks + history_marks) / 3;


      System.out.println("Average " + average_of_marks);



      if (average_of_marks >= 95) {

         System.out.println("Grade A");
```

```
    }
    else if (average_of_marks >=70) {
       System.out.println("Grade B");
    }
    else if (average_of_marks >= 50) {
       System.out.println("Grade C");
    }
    else if (average_of_marks >= 30) {
       System.out.println("Grade D");
    }
    else {
       System.out.println("Failed");
    }
  }
}
```

Output:-

```
Average 94
 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.

Code:

```
public class LeapYear{
  public static void main(String args[]){
   int year = 2024;
```

```
if(year%400==0){
   System.out.println(year + " is leap year");
}

 else if(year%4==0 && year%100 !=0){
       System.out.println(year + " is leap year");

}
else{
       System.out.println(year +" is not leap year");
}




}
}
```

Output:-
2024 is 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.

Code:-

```
public class DaysoftheWeek{
 public static void main(String args[]){
  int day = 5;
 switch(day){
  case 1 :
     System.out.println(" Day one Sunday");
   break;
  case 2 :
     System.out.println(" Day two Monday");
   break;
  case 3:
     System.out.println(" Day three Tuesday");
   break;
   case 4 :
     System.out.println(" Day four Wednesday");
   break;
   case 5 :
     System.out.println(" Day five Thursday");
   break;
```

```
 case 6 :
    System.out.println(" Day six friday");
  break;
 case 7 :
    System.out.println(" Day five Saturday");
  break;
 default:
    System.out.println("Invalid day number");
}


}
}
```

Output:
Day five Thursday.

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 value
3.

Code:

Identify the Values of Uninitialized Variables(Local variables)

```
public class IdentifyValues{
public static void main(String args[]){
byte a;
short b;
int c;
long d;
float e;
double f;
char g;
boolean h;

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

}
}

```
F:\LogicalBuilding_AC>javac IdentifyValues.java
IdentifyValues.java:12: error: variable a might not have been initialized
System.out.println("byte: " + a);
                             ^
IdentifyValues.java:13: error: variable b might not have been initialized
        System.out.println("short: " + b);
                                       ^
IdentifyValues.java:14: error: variable c might not have been initialized
        System.out.println("int: " + c);
                                     ^
IdentifyValues.java:15: error: variable d might not have been initialized
        System.out.println("long: " + d);
                                       ^
IdentifyValues.java:16: error: variable e might not have been initialized
        System.out.println("float: " + e);
                                       ^
IdentifyValues.java:17: error: variable f might not have been initialized
        System.out.println("double: " + f);
                                        ^
IdentifyValues.java:18: error: variable g might not have been initialized
        System.out.println("char: " + g);
                                       ^
IdentifyValues.java:19: error: variable h might not have been initialized
        System.out.println("boolean: " + h);
                                         ^
8 errors
```

Identify the Values of Uninitialized Variables(instance variables)
Code:

```
public class InstanceVariable{

byte a;
short b;
int c;
long d;
float e;
double f;
char g;
boolean h;

public static void main(String args[]){
InstanceVariable obj = new InstanceVariable();
      System.out.println("byte: " + obj.a);
      System.out.println("short: " + obj.b);
      System.out.println("int: " + obj.c);
```

```
        System.out.println("long: " + obj.d);
        System.out.println("float: " + obj.e);
        System.out.println("double: " + obj.f);
        System.out.println("char: " + obj.g);
        System.out.println("boolean: " + obj.h);
}
}
```

Output:

```
F:\LogicalBuilding_AC>javac InstanceVariable.java

F:\LogicalBuilding_AC>java InstanceVariable
byte: 0
short: 0
int: 0
long: 0
float: 0.0
double: 0.0
char:
boolean: false
```

Identify the Values of Uninitialized Variables(static variables)

Code:

```
public class StaticVariable{
static byte a;
static short b;
static int c;
static long d;
static float e;
static double f;
static char g;
static boolean h;

public static void main(String args[]){
        System.out.println("byte: " + a);
        System.out.println("short: " + b);
        System.out.println("int: " + c);
        System.out.println("long: " + d);
        System.out.println("float: " + e);
        System.out.println("double: " + f);
        System.out.println("char: " + g);
        System.out.println("boolean: " + h);
}
}
```

```
F:\LogicalBuilding_AC>javac StaticVariable.java

F:\LogicalBuilding_AC>java StaticVariable
byte: 0
short: 0
int: 0
long: 0
float: 0.0
double: 0.0
char:
boolean: false

F:\LogicalBuilding_AC>
```