## Day2_Java_Assignment1

## 1. Primitive Data Types

**Task:** Create a program that accepts age, height, and weight of a person and prints them with appropriate data types.

**Sample Input:**
Age: 25
Height: 5.9
Weight: 68.5

**Sample Output:**
```
Age: 25

Height: 5.9
Weight: 68.5
```

**Program:**

```
import java.util.Scanner;
public class PrimitiveDataTypes {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int age = sc.nextInt();
      double height = sc.nextDouble();
      double weight = sc.nextDouble();
      System.out.println("Age: " + age);
      System.out.println("Height: " + height);
      System.out.println("Weight: " + weight);
   }
}
```

## 2. Variables

**Task:** Declare and initialize different types of variables to store a student's information: ID, name, marks, and grade. Print them.

**Sample Input:**
ID: 101
Name: Arun
Marks: 89.5
Grade: A

**Sample Output:**
```
Student ID: 101

Name: Arun
Marks: 89.5
Grade: A
```

```
public class StudentInfo {
    public static void main(String[] args) {
        int id = 101;
        String name = "Arun";
        double marks = 89.5;
        char grade = 'A';
        System.out.println("Student ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Marks: " + marks);
        System.out.println("Grade: " + grade);
    }
}
```

## 3. Operators

**Task:** Accept two numbers and perform arithmetic, relational, and logical operations on them.

**Sample Input:**
Number1: 10
Number2: 20

**Sample Output:**

```
Addition: 30
Greater number: 20
Are both positive? true
```

```
import java.util.Scanner;
public class Operators {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println("Addition: " + (a + b));
        System.out.println("Greater number: " + (a > b ? a : b));
        System.out.println("Are both positive? " + (a > 0 && b > 0));
    }
}
```

## 4. String Concatenation

**Task:** Create a greeting message using first name and last name entered by the user.

**Sample Input:**
First Name: Ravi
Last Name: Kumar

**Sample Output:**

```
Hello, Ravi Kumar! Welcome to the system.
```

```
import java.util.Scanner;
public class Greeting {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String firstName = sc.next();
        String lastName = sc.next();
        System.out.println("Hello, " + firstName + " " + lastName + "! Welcome to the
system.");
    }
}
```

## 5. StringBuilder

**Task:** Accept a sentence and reverse it using `StringBuilder`.

**Sample Input:**
Input: Hello Java Learners

**Sample Output:**

```
Original: Hello Java Learners

Reversed: srenraeL avaJ olleH
```

```
import java.util.Scanner;
public class ReverseString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        sc.nextLine(); // to consume newline
        String input = sc.nextLine();
        StringBuilder sb = new StringBuilder(input);
        System.out.println("Original: " + input);
        System.out.println("Reversed: " + sb.reverse());
    }
}
```

## 6. String API

**Task:** Count how many times a specific character appears in a string.

**Sample Input:**
String: banana
Character: a

**Sample Output:**

```
Character 'a' appears 3 times.
```

```
import java.util.Scanner;
public class CharCount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        char ch = sc.next().charAt(0);
        int count = 0;
```

```
        for (char c : str.toCharArray()) {
            if (c == ch) count++;
        }
        System.out.println("Character '" + ch + "' appears " + count + " times.");
    }
}
```

## 7. Date, Time, and Numeric Objects

**Task:** Display the current date and format it as DD-MM-YYYY. Also, show a formatted currency value.

**Sample Input:**
Date: [current system date]
Amount: 12345.678

**Sample Output:**

```
Current Date: 20-07-2025

Formatted Amount: ₹12,345.68
```

```
import java.text.NumberFormat;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Locale;

public class DateFormatCurrency {
    public static void main(String[] args) {
        LocalDate date = LocalDate.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
        System.out.println("Current Date: " + date.format(formatter));

        double amount = 12345.678;
        NumberFormat currency = NumberFormat.getCurrencyInstance(new Locale("en", "IN"));
        System.out.println("Formatted Amount: " + currency.format(amount));
    }
}
```

## 8. Flow Control

```
import java.util.Scanner;

public class NumberType {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        if (num > 0) System.out.println("The number is positive.");
        else if (num < 0) System.out.println("The number is negative.");
        else System.out.println("The number is zero.");
    }
}
```

**Task:** Based on a number entered, print whether it's positive, negative, or zero.

**Sample Input:**
Number: -5

**Sample Output:**

```
The number is negative.
```

## 9. Conditions

**Task:** Accept marks and display the grade using `if-else`.

**Sample Input:**
Marks: 76

**Sample Output:**
```
Grade: B
```

```
import java.util.Scanner;

public class GradeCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int marks = sc.nextInt();
        if (marks >= 90) System.out.println("Grade: A");
        else if (marks >= 75) System.out.println("Grade: B");
        else if (marks >= 60) System.out.println("Grade: C");
        else System.out.println("Grade: D");
    }
}
```

## 10. Switch

**Task:** Build a simple calculator using `switch` to perform operations (+, -, *, /).

**Sample Input:**
Number1: 10
Number2: 5
Operation: *

**Sample Output:**
```
Result: 50
```

```
import java.util.Scanner;
public class Calculator {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int a = sc.nextInt();
      int b = sc.nextInt();
      char op = sc.next().charAt(0);
      switch (op) {
         case '+': System.out.println("Result: " + (a + b)); break;
         case '-': System.out.println("Result: " + (a - b)); break;
         case '*': System.out.println("Result: " + (a * b)); break;
         case '/': System.out.println("Result: " + (a / b)); break;
         default: System.out.println("Invalid operation");
      }
   }
}
```

## 11. Loops and Branching

**Task:** Print the first N even numbers using a loop.

**Sample Input:**
N = 5

**Sample Output:**

```
0 2 4 6 8
```

```
import java.util.Scanner;
public class EvenNumbers {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      for (int i = 0; i < n * 2; i += 2) {
         System.out.print(i + " ");
      }
   }
}
```

## 12. Arrays

**Task:** Accept 5 numbers, store them in an array, and display their average.

**Sample Input:**
Numbers: 10, 20, 30, 40, 50

**Sample Output:**
```
Average: 30.0
```

```
import java.util.Scanner;
public class ArrayAverage {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] nums = new int[5];
        int sum = 0;
        for (int i = 0; i < 5; i++) {
            nums[i] = sc.nextInt();
            sum += nums[i];
        }
        System.out.println("Average: " + (sum / 5.0));
    }
}
```

## 13. Enum

**Task:** Create an enum for days of the week. Print a message depending on the day.

**Sample Input:**
Day: MONDAY

**Sample Output:**
```
Start of the work week!
```

```
import java.util.Scanner;
enum Day {
   MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY,
SUNDAY
}
public class DayMessage {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      Day day = Day.valueOf(sc.next().toUpperCase());
      switch (day) {
         case MONDAY -> System.out.println("Start of the work week!");
         case FRIDAY -> System.out.println("Almost weekend!");
         case SUNDAY -> System.out.println("Relax, it's Sunday!");
         default -> System.out.println("Midweek day!");
      }
   }
}
```

## 14. OOPs Concepts

**Task:** Create a `Student` class with fields for name and marks. Create an object and display its data.

**Sample Input:**
Name: Riya
Marks: 87

**Sample Output:**
```
Student Name: Riya

Marks: 87
```

## 15. Inheritance

**Task:** Create a class `Employee` and a subclass `Manager` that extends `Employee` and adds department information.

**Sample Input:**
Name: Raj

Salary: 50000
Department: Sales

**Sample Output:**

```
Name: Raj

Salary: 50000
Department: Sales
```

**14. program**

```
class Student {
    String name;
    int marks;
    Student(String name, int marks) {
        this.name = name;
        this.marks = marks;
    }
    void display() {
        System.out.println("Student Name: " + name);
        System.out.println("Marks: " + marks);
    }
}
public class StudentTest {
    public static void main(String[] args) {
        Student s = new Student("Riya", 87);
        s.display();
    }
}
```

**15 .PROGRAM**

```
class Employee {
    String name;
    double salary;
    Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
}
class Manager extends Employee {
    String department;
    Manager(String name, double salary, String department) {
        super(name, salary);
        this.department = department;
    }
    void display() {
        System.out.println("Name: " + name);
        System.out.println("Salary: " + salary);
        System.out.println("Department: " + department);
    }
}
public class InheritanceTest {
    public static void main(String[] args) {
        Manager m = new Manager("Raj", 50000, "Sales");
        m.display();
    }
}
```