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
```

Code:

```
import java.util.Scanner;

public class PersonInfo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Age: ");
        int age = sc.nextInt();

        System.out.print("Height: ");
        double height = sc.nextDouble();

        System.out.print("Weight: ");
        double weight = sc.nextDouble();

        System.out.println("Age: " + age);
        System.out.println("Height: " + height);
        System.out.println("Weight: " + weight);

        sc.close();
    }
}
```

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
```

Code:

```
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("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
```

Code:

```
import java.util.Scanner;
public class Operators{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Number1: ");
        int number1 = sc.nextInt();
        System.out.print("Number2: ");
        int number2 = sc.nextInt();

        int sum = number1 + number2;
        int greater = (number1 > number2) ? number1 : number2;
        boolean bothPositive = (number1 > 0) && (number2 > 0);

        System.out.println("Addition: " + sum);
        System.out.println();
        System.out.println("Greater number: " + greater);
        System.out.println("Are both positive? " + bothPositive);

        sc.close();
    }
}
```

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.
```

Code:

```
import java.util.Scanner;

public class Greeting {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("First Name: ");
        String firstName = sc.nextLine();

        System.out.print("Last Name: ");
        String lastName = sc.nextLine();
        String greeting = "Hello, " + firstName + " " + lastName + "! Welcome to the system.";

        System.out.println(greeting);

        sc.close();
    }
}
```

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
```

Code:

```
import java.util.Scanner;

public class ReverseSentence {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Input: ");
        String sentence = sc.nextLine();

        // Use StringBuilder to reverse the sentence
        StringBuilder sb = new StringBuilder(sentence);
        String reversed = sb.reverse().toString();

        System.out.println("Original: " + sentence);
        System.out.println("Reversed: " + reversed);

        sc.close();
    }
}
```

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.
```

Code:

```
import java.util.Scanner;

public class CharacterCount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("String: ");
        String inputString = sc.nextLine();

        System.out.print("Character: ");
        char ch = sc.nextLine().charAt(0);

        int count = 0;
        for (int i = 0; i < inputString.length(); i++) {
            if (inputString.charAt(i) == ch) {
                count++;
            }
        }

        System.out.println("Character '" + ch + "' appears " + count + " times.");

        sc.close();
    }
}
```

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
```

Code :

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

public class DateCurrencyFormat {
    public static void main(String[] args) {

        LocalDate currentDate = LocalDate.now();
        DateTimeFormatter dateFormatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
        String formattedDate = currentDate.format(dateFormatter);


        double amount = 12345.678;


        Locale indiaLocale = new Locale("en", "IN");
        NumberFormat currencyFormatter = NumberFormat.getCurrencyInstance(indiaLocale);
        String formattedAmount = currencyFormatter.format(amount);


        System.out.println("Current Date: " + formattedDate);
        System.out.println("Formatted Amount: " + formattedAmount);
    }
}
```

8. Flow Control

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

Sample Input:
Number: -5

Sample Output:

```
The number is negative.
```

Code:

```
import java.util.Scanner;

public class NumberCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Number: ");
        int number = sc.nextInt();

        if (number > 0) {
            System.out.println("The number is positive.");
        } else if (number < 0) {
            System.out.println("The number is negative.");
        } else {
            System.out.println("The number is zero.");
        }

        sc.close();
    }
}
```

9. Conditions

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

Sample Input:
Marks: 76

Sample Output:

```
Grade: B
```

Code:

```
import java.util.Scanner;

public class GradeCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Marks: ");
        int marks = sc.nextInt();

        char grade;
        if (marks >= 80 && marks <= 100) {
            grade = 'A';
        } else if (marks >= 70) {
            grade = 'B';
        } else if (marks >= 60) {
            grade = 'C';
        } else if (marks >= 50) {
            grade = 'D';
        } else {
            grade = 'F';
        }

        System.out.println("Grade: " + grade);

        sc.close();
    }
}
```

10. Switch

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

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

Sample Output:

```
Result: 50
```

Code:

```
import java.util.Scanner;

public class SimpleCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Number1: ");
        double number1 = sc.nextDouble();

        System.out.print("Number2: ");
        double number2 = sc.nextDouble();

        System.out.print("Operation (+, -, *, /): ");
        char operation = sc.next().charAt(0);

        double result;

        switch (operation) {
            case '+':
                result = number1 + number2;
                System.out.println("Result: " + result);
                break;
            case '-':
                result = number1 - number2;
                System.out.println("Result: " + result);
                break;
            case '*':
                result = number1 * number2;
                System.out.println("Result: " + result);
                break;
            case '/':
                if (number2 != 0) {
                    result = number1 / number2;
                    System.out.println("Result: " + result);
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operation.");
        }

        sc.close();
    }
}
```

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
```

Code:

```
import java.util.Scanner;

public class EvenNumbers {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("N = ");
        int N = sc.nextInt();

        for (int i = 0; i < N; i++) {
            System.out.print((2 * i) + " ");
        }

        System.out.println();

        sc.close();
    }
}
```

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
```

Code:

```
import java.util.Scanner;

public class Average {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int[] numbers = new int[5];
        int sum = 0;

        System.out.println("Enter 5 numbers:");

        for (int i = 0; i < 5; i++) {
            numbers[i] = sc.nextInt();
            sum += numbers[i];
        }

        double average = (double) sum / numbers.length;

        System.out.println("Average: " + average);

        sc.close();
    }
}
```

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!
```

Code:

```
import java.util.Scanner;

public class WeekDays {
    enum Day {
        MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Day: ");
        String input = sc.next().toUpperCase();

        Day day;
        try {
            day = Day.valueOf(input);
        } catch (IllegalArgumentException e) {
            System.out.println("Invalid day entered.");
            sc.close();
            return;
        }

        switch (day) {
            case MONDAY:
                System.out.println("Start of the work week!");
                break;
            case FRIDAY:
                System.out.println("Last work day, almost weekend!");
                break;
            case SATURDAY:
            case SUNDAY:
                System.out.println("It's the weekend. Relax!");
                break;
            default:
                System.out.println("Another working day.");
        }
        sc.close();
    }
}
```

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
```

Code:

```
class Student {
    String name;
    int marks;


    public Student(String name, int marks) {
        this.name = name;
        this.marks = marks;
    }


    public void display() {
        System.out.println("Student Name: " + name);
        System.out.println("Marks: " + marks);
    }
}

public class StudentObject {
    public static void main(String[] args) {

        Student student = new Student("Riya", 87);

        student.display();
    }
}
```

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
```

Code:

```
class Employee {
   String name;
   double salary;


   public Employee(String name, double salary) {
      this.name = name;
      this.salary = salary;
   }


   public void display() {
      System.out.println("Name: " + name);
      System.out.println("Salary: " + (int)salary);
   }
}

class Manager extends Employee {
   String department;

   public Manager(String name, double salary, String department) {
      super(name, salary);
      this.department = department;
   }

   @Override
   public void display() {
      super.display();
      System.out.println("Department: " + department);
   }
}

public class EmployeeDemo {
   public static void main(String[] args) {

      Manager manager = new Manager("Raj", 50000, "Sales");
      manager.display();
   }
}
```