Java Programming - Day 3

Control structures in Java are constructs that allow developers to control the flow of execution of a program based on certain conditions or repetitions.

These structures are divided into three main types:

1. **Decision-Making/Conditional Statements**
2. **Looping/Iterative Statements**
3. **Jump Statements**

1. **Decision-Making/Conditional Statements**
    1. What are conditional statements in Java? Why are they used?
    2. List the different types of conditional statements available in Java.

- if statement
- if-else statement
- else-if ladder
- Nested if
- switch statement
- Ternary operator (?:)

3. Syntax and Usage:

Problem Statements:

1. **Check if a Character is a Vowel or Consonant**:
   Given a character, determine if it is a vowel (a, e, i, o, u) or a consonant using conditional statements.
2. Find the biggest among two numbers.?
3. Find the number belonging to which group above 50, between 40 to 50, less than 40.?
4. Write a Program to check the given number is divisible by both 3&4 if it is so print "Good Morning". If a number is divisible by only 4 but not 3 then print "Good Afternoon",
   If it is divisible by only 3 but not 4 then print "Good Evening" otherwise print "Good Night".?
5. Java program to check if a given year is a leap year or not.?
6. Write a program that takes a score as input and prints out the corresponding grade according to the following grading scheme:
    - 90 or above: A
    - 80-89: B
    - 70-79: C
    - 60-69: D
    - Below 60: F.?

7. Develop a simple program to calculation to perform an operation we have four choices if you choose
    1.Addition
    2.Subtraction
    3.Multiplication
    4.Division

8. **Problem Statement 1**

Shraddha Kapoor's professor suggested that she study hard and prepare well for the lesson on seasons. If her professor says month then, she has to tell the name of the season corresponding to that month. So write the program to get the solution to the above task?

- March to May – Spring Season
- June to August – Summer Season
- September to November – Autumn Season
- December to February – Winter Season

Note: The entered month should be in the range of 1 to 12. If the user enters a month less than 1 or greater than 12 then the message "Invalid Month Entered" should get displayed.

Sample Input 1:
Enter month: 6

Sample Output 1:
Season: Summer

Sample Input 2:
Enter month: 15

Sample Output 2:
Invalid Month Entered

---

1. Decision-Making Statements

These statements execute a block of code based on conditions.

a. if Statement

Executes a block of code if the condition is true.

Syntax:

```
if (condition) {
   // code to execute if condition is true
}
```

Example 1: Checking if a Number is Positive

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

```
        int number = 10;
        if (number > 0) {
            System.out.println("The number is positive.");
        }
    }
}
```

b. if-else Statement

Provides an alternative block of code if the condition is false.

Syntax:

```
if (condition) {
    // code to execute if condition is true
} else {
    // code to execute if condition is false
}
```

Example 1:

```
class Numcheck{
    public static void main(String[] args) {
        int number = -5;
        if (number > 0) {
            System.out.println("The number is positive.");
        } else {
            System.out.println("The number is not positive.");
        }
    }
}
```

Example 2:

```
class EvenOrOdd {
    public static void main(String[] args) {
        int number = 7;
        if (number % 2 == 0) {
            System.out.println("The number is even.");
        } else {
            System.out.println("The number is odd.");
        }
    }
}
```

Example 3: Using of Scanner class

```
import java.util.Scanner;
```

```
class EvenOrOdd {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a number: ");
      int number = scanner.nextInt();
      if (number % 2 == 0) {
         System.out.println("The number is even.");
      } else {
         System.out.println("The number is odd.");
      }
   }
}
```

c. else if Ladder

Allows multiple conditions to be checked in sequence.

Syntax:

```
if (condition1) {
   // code for condition1
} else if (condition2) {
   // code for condition2
} else {
   // code if none of the above conditions are true
}
```

Example 1:

```
class Numcheck {
   public static void main(String[] args) {
      int number = 0;
      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.");
      }
   }
}
```

Example 2: We check if a number is divisible by both 3 and 5:

```
class NumDiv {
   public static void main(String[] args) {
      int number = 15;
      if (number % 3 == 0 && number % 5 == 0) {
         System.out.println("The number is divisible by both 3 and 5.");
      } else if (number % 3 == 0) {
         System.out.println("The number is divisible by 3.");
```

```
        } else if (number % 5 == 0) {
            System.out.println("The number is divisible by 5.");
        } else {
            System.out.println("The number is not divisible by 3 or 5.");
        }
    }
}
```

d. nested if-else Statement

A **nested if-else statement** is when an if-else statement is placed inside another if-else block. It allows multiple conditions to be evaluated in a hierarchical or dependent manner.

Syntax:

```
if (condition1) {
    if (condition2) {
        // Code to execute if condition1 and condition2 are true
    } else {
        // Code to execute if condition1 is true and condition2 is false
    }
} else {
    // Code to execute if condition1 is false
}
```

Example 1:

```
class NestedIfExample {
    public static void main(String[] args) {
        int number = 0;
        if (number >= 0) {
            if (number == 0) {
                System.out.println("The number is zero.");
            } else  {
                System.out.println("The number is positive.");
            }
        } else {
            System.out.println("The number is negative.");
        }
    }
}
```

---

e. switch Statement

Executes a block of code based on the value of an expression.

Syntax:

```
switch (expression) {
   case value1:
      // code for value1
      break;
   case value2:
      // code for value2
      break;
   default:
      // code if no case matches
}
```

Example:

```
class SwitchCaseExample {

   public static void main(String[] args) {

      int day = 3;

      switch (day) {

         case 1:

            System.out.println("Monday");

            break;

         case 2:

            System.out.println("Tuesday");

            break;

         case 3:

            System.out.println("Wednesday");

            break;

         case 4:

            System.out.println("Thursday");

            break;

         case 5:

            System.out.println("Friday");

            break;
```

```
            case 6:
                System.out.println("Saturday");
                break;
            case 7:
                System.out.println("Sunday");
                break;
            default:
                System.out.println("Invalid day");
        }
    }
}
```

Ternary operator:

The **ternary operator** in Java is a concise way to write simple if-else conditions. It is represented by the symbols ?: and is also known as the conditional operator.

Syntax:

condition ? expression1 : expression2;

- **condition**: A boolean expression that evaluates to either true or false.
- **expression1**: This is executed and returned if the condition is true.
- **expression2**: This is executed and returned if the condition is false.

Example 1: Basic Usage

```
class Main {
    public static void main(String[] args) {
        int a = 10, b = 20;
        int max = (a > b) ? a : b; // Finds the maximum of a and b
        System.out.println("The maximum is: " + max);
    }
}
```

Output:

The maximum is: 20

Example 2: Nested Ternary Operator

```
class Main {
   public static void main(String[] args) {
      int num = -5;
      String result = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";
      System.out.println("The number is: " + result);
   }
}
```

Output:

The number is: Negative

Example 3: Assign Values Conditionally

```
class Main {
   public static void main(String[] args) {
      boolean isEven = true;
      String message = isEven ? "The number is even." : "The number is odd.";
      System.out.println(message);
   }
}
```

Output:

The number is even.

Advantages:

1. **Conciseness**: Reduces the lines of code for simple conditions.
2. **Readability**: Makes straightforward conditions easier to read.

Problem Statements:

1. **Check if a Character is a Vowel or Consonant**:
   Given a character, determine if it is a vowel (a, e, i, o, u) or a consonant using conditional statements.

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

```
        System.out.print("Enter a character: ");
        char ch = sc.next().charAt(0);
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
            ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
            System.out.println(ch + " is a vowel.");
        } else {
            System.out.println(ch + " is a consonant.");
        }
    }
}
```

4. Find the biggest among two numbers.?

CODE:-

```
import java. util.Scanner;

    class BiggestNumber{
            public static void main(String[] args){
            Scanner scan=new Scanner(System.in);
            int num1=scan.nextInt();
            int num2=scan.nextInt();
            if(num1>num2){
            System.out.println(num1+"is the bigger number");
            }
            else if(num1<num2){
            System.out.println(num2+"is the bigger number");
            }
            else{
            System.out.println("both number are equals");
            }
        }
    }
```

5. Find the number belonging to which group above 50, between 40 to 50, less than 40.?

CODE:-

```
    import java. util.Scanner;
    class BelongingToWitchGroup{
            public static void main(String [] args){
            Scanner scan=new Scanner(System.in);
            int num=45;
            if(num>50){
          System. out.println("Given number is belongs to group 50");
            }
```

```
        else if(num>=40&&num<=50){
        System.out.println("Given number is belongs to group b/w 40 to 50");
        }
        else{
        System.out.println("Given number is less then 40");
        }
    }
}
```

6. Write a Program to check the given number is divisible by both 3&4 if it is so print "Good Morning".

If a number is divisible by only 4 but not 3 then print "Good Afternoon",

If it is divisible by only 3 but not 4 then print "Good Evening" otherwise print "Good Night".?

CODE:-

```
class Number{
        public static void main(String[] args){
        int num=19;
        if(num%3==0&&num%4==0){
        System. out.println("Good morning");
        }
        else if(num%3==0){
        System.out.println("Good afternoon");
        }
        else if(num%4==0){
        System.out.println("Good evening");
        }
        else{
        System.out.println("Good night");
        }
    }
}
```

7. Java program to check if a given year is a leap year or not.?

CODE:-

```
import java.util.Scanner;
class LeapYearChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the year: ");
        int year = scanner.nextInt();
        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.");
        }
    }

}
```

8. Write a program that takes a score as input and prints out the corresponding grade according to the following grading scheme:

- **90 or above: A**
- **80-89: B**
- **70-79: C**
- **60-69: D**
- **Below 60: F.?**

CODE:-

```
import java.util.Scanner;
class GradeCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the score: ");
        int score = scanner.nextInt();
        char grade;
        if (score >= 90) {
            grade = 'A';
        } else if (score >= 80) {
            grade = 'B';
        } else if (score >= 70) {
            grade = 'C';
        } else if (score >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        System.out.println("Grade: " + grade);
    }
}
```

9. Develop a simple program to calculation to perform an operation we have four choices if you choose

1.Addition
2.Subtraction
3.Multiplication
4.Division

CODE:-

```
import java.util.Scanner;
class SimpleCalculator {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Select an operation:");
      System.out.println("1. Addition");
      System.out.println("2. Subtraction");
      System.out.println("3. Multiplication");
      System.out.println("4. Division");
      int choice = scanner.nextInt();
      System.out.print("Enter the first number: ");
      double num1 = scanner.nextDouble();
      System.out.print("Enter the second number: ");
      double num2 = scanner.nextDouble();
      switch (choice) {
         case 1:
            System.out.println("Result: " + (num1 + num2));
            break;
         case 2:
            System.out.println("Result: " + (num1 - num2));
            break;
         case 3:
            System.out.println("Result: " + (num1 * num2));
            break;
         case 4:

            if (num2 != 0) {
               System.out.println("Result: " + (num1 / num2));
            } else {
               System.out.println("Error: Division by zero is not allowed.");
            }
            break;
         default:
            System.out.println("Invalid choice. Please select a valid operation.");
            break;
      }
         }
}
```

10. Problem Statement 1

Shraddha Kapoor’s professor suggested that she study hard and prepare well for the lesson on seasons. If her professor says month then, she has to tell the name of the season corresponding to that month. So write the program to get the solution to the above task?

- March to May – Spring Season
- June to August – Summer Season
- September to November – Autumn Season
- December to February – Winter Season

Note: The entered month should be in the range of 1 to 12. If the user enters a month less than 1 or greater than 12 then the message “Invalid Month Entered” should get displayed.

Sample Input 1:
Enter month: 6

Sample Output 1:
Season: Summer

Sample Input 2:
Enter month: 15

Sample Output 2:
Invalid Month Entered

CODE:

```
import java.util.Scanner;

class SeasonFinder {

   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter month: ");

      int month = scanner.nextInt();

      if (month < 1 || month > 12) {

         System.out.println("Invalid Month Entered");

      } else if (month >= 3 && month <= 5) {

         System.out.println("Season: Spring");

      } else if (month >= 6 && month <= 8) {
```

```
            System.out.println("Season: Summer");
        } else if (month >= 9 && month <= 11) {
            System.out.println("Season: Autumn");
        } else {
            System.out.println("Season: Winter");
        }
    }
}
```