# Angel Maria S

### IT Engineer

To work in a challenging and dynamic environment and to keep adding value to the organization that I represent and serve, while also concurrently upgrading my skills and knowledge

angel3008mary@gmail.com 9961944064 Pala, India

### **Assignment 7**

## Differences between the switch statement and if-else-if ladder statement Switch:

In Java, the switch statement only supports integer types (byte, short, int, char) and certain enumerated types. Starting from Java 7, you can also use strings in switch statements.

In Java's switch Statement, after a case block is executed, there is no automatic fall-through behaviour. You need to explicitly use break statements to exit the switch block.

The switch statement is more suitable for simple value comparisons (equality checks) rather than complex conditions involving multiple variables.

The switch statement can have a default case that is executed when none of the cases match the expression.

```
Syntax,
switch (expression) {
   case value1:
     // code to execute if expression == value1
     break;
   case value2:
     // code to execute if expression == value2
     break;
   // more cases
   default:
     // code to execute if no cases match
}
```

### If-Else-If Ladder:

The if-else-if ladder in java can handle Boolean expressions and comparisons of any type.

Each if or else if block is independent, and there is no automatic fall-through. An if-else-if ladder allows for more complex conditions and can involve multiple variables and logical operators.

An if-else-if ladder can also have a final else block that is executed when none of the conditions are met.

```
Syntax,
if (condition1) {
    // code to execute if condition1 is true
} else if (condition2) {
    // code to execute if condition2 is true
} else if (condition3) {
    // code to execute if condition3 is true
}
// more else if blocks
else {
    // code to execute if no conditions are true
}
```

### Nested Simple if and Nested if-else in Java Nested if Statements:

A nested if statements is an if statement that is placed within another if statement. This allows for more complex conditional logic where you have multiple conditions that need to be checked. The inner if statement is only evaluated if the outer if statement's condition is true.

```
Syntax;
if (condition1) {
    // Outer if block
    if (condition2) {
        // Inner if block
        // Code to execute if both condition1 and condition2 are true
    }
}
```

#### **Nested if-else Statements:**

A nested if-else statement involves placing an if-else statement within another if or else block. This allows you to handle multiple cases and different outcomes based on various conditions.

```
Syntax;
if (condition1) {
    // Outer if block
    if (condition2) {
        // Inner if block
        // Code to execute if condition1 and condition2 are true
    } else {
        // Inner else block
        // Code to execute if condition1 is true but condition2 is false
    }
} else {
    // Outer else block
    // Code to execute if condition1 is false
```

```
E.g.
int x = 10;
int y = 20;

if (x > 5) {
    if (y > 15) {
        System.out.println("Both conditions are true.");
    } else {
        System.out.println("Condition 1 is true, but condition 2 is false.");
    }
} else {
    System.out.println("Condition 1 is false.");
}
```