Java Conditional Statements

Conditional statements are a key part of Java and almost every programming language. They help our program make decisions. Think of it as a statement like "if something happens, do this, otherwise do something else" instructions for our computer.

Conditional statements let our code choose different paths depending on whether a condition is true or false. It's almost like asking, "Is it raining? If yes, take an umbrella. If not, leave the umbrella at home."

- The **if** Statement : This is the most basic conditional in Java.

```
if (condition) {
        // code to run if the condition is TRUE
}
```

Example:

```
int age = 18;
if (age >= 18) {
        System.out.println("You can vote!");
}
```

If `age` is 18 or more, then it prints the message.

- The **if-else** Statement : This lets you do one thing if the condition is true and something else if it's false.

```
if (condition) {
        // code to run  if TRUE
} else {
        // code to run if FALSE
}
```

Example:

```
int number = 7;
if (number % 2 == 0) {
        System.out.println("Even number");
} else {
        System.out.println("Odd number");
}
```

If `number` is even, it prints "Even number". Otherwise, it prints "Odd number".

- The **else if** Ladder : When we have more than two choices, we use `else if`:

```
if (condition1) {
        // code to run  if condition1 is TRUE
} else if (condition2) {
        // code to run if condition2 is TRUE
} else {
        // code if none are TRUE
}
```

Example:

```
int marks = 85;
if (marks >= 90) {
        System.out.println("Grade: A");
} else if (marks >= 75) {
        System.out.println("Grade: B");
} else {
        System.out.println("Grade: C");
}
```

With `marks` of 85, it prints "Grade: B".

- The **switch** Statement : The `switch` statement is useful when we want to compare one variable against many possible values.

```
switch (variable) {
        case value1:
                // code for value1
                break;
        case value2:
                // code for value2
                 break;
        default:
                 // code if variable doesn't match any case
}
```

Example:

```
int day = 2;
switch (day) {
        case 1:
                System.out.println("Monday");
                break;
        case 2:
                System.out.println("Tuesday");
                break;
        default:
                System.out.println("Another day");
}
```

If `day` is 2, the program prints "Tuesday".

Why Are Conditionals Important?

- They allow your program to make decisions.
- They help in solving real-life problems using code (like checking if a student passes or fails). Most games and apps use conditionals to respond to user actions.

Tips to remember

- Always check your conditions carefully.
- Use curly braces `{}` even if they're optional—it makes your code clearer. (When only one statement is to be executed, {} are optional).
- Try making small example programs and see how the output changes as you change the values.
- When using switch, don't forget to write 'break;' for each case. If any case matches with test variable then all the executable statements written after it gets executed. This is called the fall-through feature.
- ***The 'if' statement can check complex conditions that can be written using relational (==, !=, <, <=, >, >=) and logical operators (!, ||, &&) while the 'switch' statement can check for equality only.***

Summary Table

Java Statement	When to Use	Example
If	One condition (true/false)	if (x > 5) {...}
if-else	Two possible outcomes	if (...){...} else {...}
else if ladder	Multiple choices	else if (x < 0) {...}
Switch	Many specific values for one variable	switch (day) {...}