Java Conditional Statements

1. if Statement

Purpose:

Executes a block of code only if a specified condition is true.

Syntax:

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

Example:

```
int age = 20;
if (age >= 18) {
    System.out.println("You are eligible to vote.");
}
```

Output:

You are eligible to vote.

2. if-else Statement

Purpose:

Executes one block if the condition is true, and another block if the condition is false.

Syntax:

```
if (condition) {
    // Code if condition is true
} else {
    // Code if condition is false
}
```

Example:

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

Output:

Odd number

3. if-else-if Ladder

Purpose:

Tests multiple conditions sequentially. Executes the first block where the condition is true.

Syntax:

```
if (condition1) {
    // Executes if condition1 is true
} else if (condition2) {
    // Executes if condition2 is true
} else {
    // Executes if none of the conditions are true
}
```

Example:

```
int marks = 85;
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("Fail");
}
```

Output:

Grade B

4. Nested if Statements

Purpose:

Allows an if or if-else inside another if or else block. Useful for multiple level checks.

Syntax:

```
if (condition1) {
    if (condition2) {
        // Code if both condition1 and condition2 are true
    }
}
```

Example:

```
int age = 25;
boolean hasLicense = true;
if (age >= 18) {
    if (hasLicense) {
        System.out.println("You can drive.");
    } else {
        System.out.println("You need a license to drive.");
    }
} else {
    System.out.println("You are too young to drive.");
```

```
}
```

Output:

You can drive.

5. switch Statement

Purpose:

Used when you have multiple fixed options to check for equality (like int, char, String).

Syntax:

```
switch (expression) {
    case value1:
        // Code for value1
        break;
    case value2:
        // Code for value2
        break;
    ...
    default:
        // Default code if no match
}
```

Example:

```
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;
    default:
        System.out.println("Invalid day");
}
```

Output:

Wednesday

6. Summary Table

Statement Type	Use Case
if	Single condition
if-else	Binary condition (true/false)
if-else-if	Multiple conditions in sequence
nested if	Decision based on multiple dependent conditions
switch	Multiple specific values of an expression (e.g. menu)