Conditional Statements in Java

Introduction

In Java programming, conditional statements are used to perform different actions based on different conditions. A condition is an expression that evaluates to either true or false. Conditional statements help the program to make decisions during execution, which makes programs flexible and intelligent. Without conditional statements, programs would execute line by line without any decision-making ability.

Why Conditional Statements are Required

Conditional statements are required to control the flow of execution of a program. They allow the program to check conditions such as marks, age, eligibility, comparison of values, and logical decisions. For example, deciding whether a student has passed or failed, or whether a user is eligible to vote, is possible only using conditional statements.

Types of Conditional Statements in Java

Java provides the following conditional statements: 1) if statement, 2) if-else statement, 3) if-else-if ladder, 4) nested if statement, and 5) switch statement. Each of these is used in different situations depending on the complexity of conditions.

1. if Statement

The if statement is the simplest conditional statement in Java. It executes a block of code only when the given condition is true. If the condition is false, the code inside the if block is skipped and execution continues with the next statement.

Syntax:

```
if(condition) {
  // code to be executed
}
```

Example: Checking whether a person is eligible to vote.

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

2. if-else Statement

The if-else statement is used when there are two possible outcomes. If the condition is true, the if block is executed. If the condition is false, the else block is executed. This ensures that one block of code will always execute.

Example: Checking whether a student has passed or failed.

```
int marks = 40;
if(marks >= 35) {
  System.out.println("Pass");
} else {
  System.out.println("Fail");
}
```

3. if-else-if Ladder

The if-else-if ladder is used when multiple conditions need to be checked one after another. As soon as one condition becomes true, the corresponding block is executed and the rest of the conditions are skipped. It is commonly used for grading systems and range-based decisions.

Example: Assigning grades based on marks.

```
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");
}
```

4. Nested if Statement

A nested if statement is an if statement inside another if statement. It is used when one condition depends on another condition. Nested if statements should be used carefully to avoid complexity.

Example: Checking eligibility for blood donation.

```
int age = 25;
int weight = 55;
if(age >= 18) {
  if(weight >= 50) {
    System.out.println("Eligible for blood donation");
  }
}
```

5. switch Statement

The switch statement is used when a variable needs to be compared with multiple constant values. It is easier to read than multiple if-else statements when checking fixed options. Each case must end with a break statement to prevent fall-through.

Example: Displaying day names.

```
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");
}
```

Conclusion

Conditional statements are one of the most important concepts in Java programming. They allow the program to make decisions and execute different blocks of code based on conditions. Understanding conditional statements is essential before learning loops, arrays, and advanced Java concepts.