Conditional Operators in Java:
Conditional operators in Java are:

Ternary Operator (`? :`): It acts as a shorthand for an `if-else` statement.
java
Copy code
```
int a = 10, b = 20;
int min = (a < b) ? a : b;  // Assigns min = 10
```

-

Types of Operators Based on Number of Operands:

- **Unary Operators:** Operate on a single operand (e.g., `++`, `--`, `!`).
- **Binary Operators:** Operate on two operands (e.g., `+`, `-`, `*`, `/`, `==`).
- **Ternary Operator:** Uses three operands (e.g., `condition ? value1 : value2`).

Use of Switch Case in Java:

- The `switch` statement is used to replace multiple `if-else` conditions.
- It is useful when a variable needs to be compared against multiple values.

Example:
java
Copy code
```
int day = 3;
switch (day) {
    case 1: System.out.println("Monday"); break;
    case 2: System.out.println("Tuesday"); break;
    default: System.out.println("Other Day");
}
```

-

Priority Levels of Arithmetic Operations in Java:
The precedence of arithmetic operators in Java follows:

- **Highest:** `()`
- `*`, `/`, `%`
- `+`, `-`

- **Lowest:** `=`

Conditional Statements and Their Use in Java:

- Conditional statements allow the execution of different code blocks based on conditions.
- **Types:** `if`, `if-else`, `else-if`, `switch-case`.
- **Use:** Decision-making in programs.

Syntax of if-else Statement in Java:

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

Three Types of Iterative Statements in Java:

- **For loop**
- **While loop**
- **Do-while loop**

Difference Between For Loop and Do-While Loop:

Feature	For Loop	Do-While Loop
Execution	Executes if condition is true	Executes at least once
Condition Check	Before loop execution	After loop execution
Syntax	`for(initialization; condition; update) {}`	`do { } while(condition);`

Program to Print Numbers from 1 to 10:

java
Copy code
```
public class PrintNumbers {
    public static void main(String[] args) {
```

```
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

Conditional Operators in Java:

Conditional operators in Java are:

Ternary Operator (`? :`): It acts as a shorthand for an `if-else` statement.

java

Copy code

```
int a = 10, b = 20;
int min = (a < b) ? a : b;  // Assigns min = 10
```

-

Types of Operators Based on Number of Operands:

- **Unary Operators:** Operate on a single operand (e.g., `++`, `--`, `!`).
- **Binary Operators:** Operate on two operands (e.g., `+`, `-`, `*`, `/`, `==`).
- **Ternary Operator:** Uses three operands (e.g., `condition ? value1 : value2`).

Use of Switch Case in Java:

- The `switch` statement is used to replace multiple `if-else` conditions.
- It is useful when a variable needs to be compared against multiple values.

Example:

java

Copy code

```
int day = 3;
switch (day) {
    case 1: System.out.println("Monday"); break;
    case 2: System.out.println("Tuesday"); break;
    default: System.out.println("Other Day");
}
```

-

Priority Levels of Arithmetic Operations in Java:
The precedence of arithmetic operators in Java follows:

- **Highest:** `()`
- `*`, `/`, `%`
- `+`, `-`
- **Lowest:** `=`

Conditional Statements and Their Use in Java:

- Conditional statements allow the execution of different code blocks based on conditions.
- **Types:** `if`, `if-else`, `else-if`, `switch-case`.
- **Use:** Decision-making in programs.

Syntax of if-else Statement in Java:

java
Copy code

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

Three Types of Iterative Statements in Java:

- **For loop**
- **While loop**
- **Do-while loop**

Difference Between For Loop and Do-While Loop:

Feature	For Loop	Do-While Loop
Execution	Executes if condition is true	Executes at least once
Condition Check	Before loop execution	After loop execution
Syntax	`for(initialization; condition; update) {}`	`do { } while(condition);`

Program to Print Numbers from 1 to 10:

java
Copy code

```java
public class PrintNumbers {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```