Course Name - Object Oriented Programming using Java

Lecture 7– Access specifiers, Operators, Control statements & loops

Presented By
Dr. Sudipta Sahana
Asso. Prof.
Dept. of CSE
UEM - Kolkata

Topic of Interest

- Access specifiers
- Access Control Modifiers
- Non-Access Modifiers Object
- Operators
- Control statements
- If statement
- If...else statement
- Nested if statement
- Switch statement
- Java Loop
- For loop
- While loop
- Do...while loop

Access specifiers

Modifiers/specifiers are keywords that you add to those definitions to change their meanings. Java language has a wide variety of modifiers, including the following −

- Java Access Modifiers
- Non Access Modifiers

Access Control Modifiers:

· Java provides a number of access modifiers to set access levels for classes, variables, methods. The four access levels are −

- Visible to the package, the default. No modifiers are needed.
- Visible to the class only (private).
- Visible to the world (public).
- Visible to the package and all subclasses (protected).

Non-Access Modifiers:

Java provides a number of non-access modifiers to achieve many other functionality.

- The static modifier for creating class methods and variables.
- The final modifier for finalizing the implementations of classes, methods, and variables.
- The abstract modifier for creating abstract classes and methods.
- The synchronized and volatile modifiers, which are used for threads.

Operators:

Java provides a rich set of operators to manipulate variables. We can divide all the Java operators into the following groups −

- Arithmetic Operators
- Relational Operators
- Bitwise Operators
- Logical Operators
- Assignment Operators
- Misc Operators

Control statements

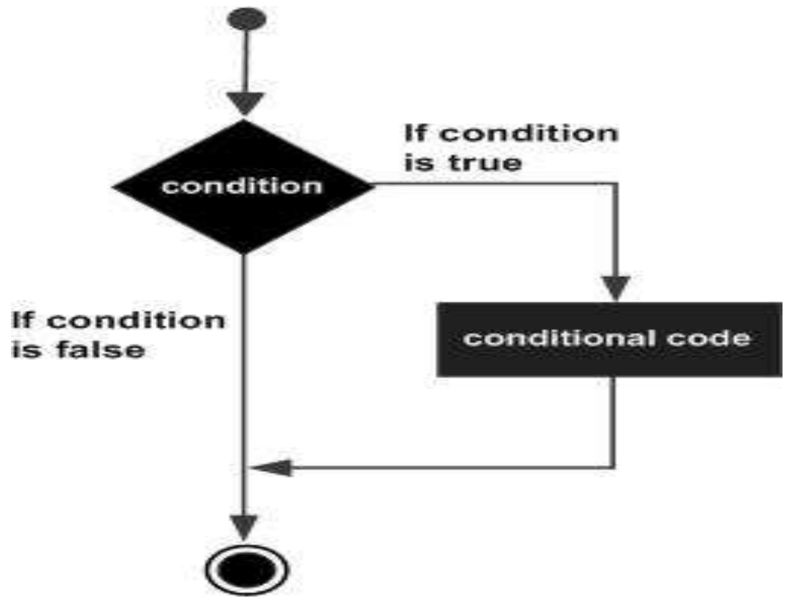
A control statement works as a determiner for deciding the next task of the other statements whether to execute or not. An ‘If’ statement decides whether to execute a statement or which statement has to execute first between the two. In Java, the control statements are divided into three categories which are selection statements, iteration statements, and jump statements. A program can execute from top to bottom but if we use a control statement. We can set order for executing a program based on values and logic.

- **if statement**
- **if...else statement**
- **nested if statement**
- **switch statement**

If statement

An if statement consists of a Boolean expression followed by one or more statements.

```
if (condition) {
Statement 1; //if condition becomes true then this will be executed
}
Statement 2; //this will be executed irrespective of condition becomes true or false
```

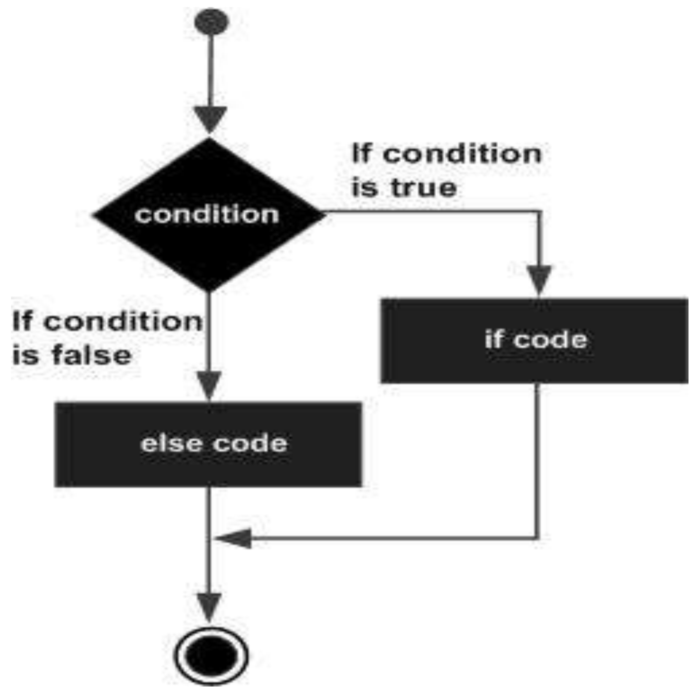


Flow diagram

If...else statement

In if…else statement, if condition is true then statements in if block will be executed but if it comes out as false then else block will be executed..

```
if (condition) {
Statement 1; //if condition becomes true then this will be executed
 }
```



Flow diagram

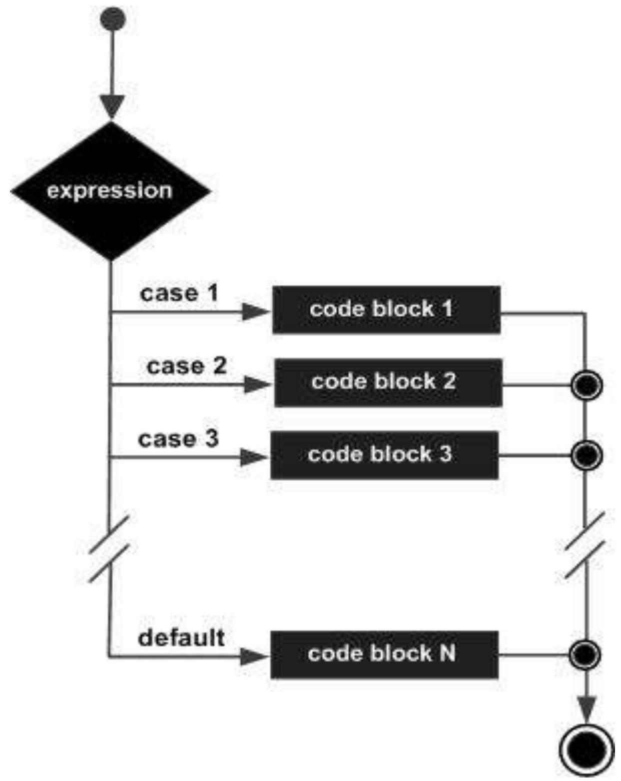
Nested if statement

Nested if statement is if inside an if block. It is same as normal if…else statement but they are written inside another if…else statement.

```
if (condition1) {
Statement 1; //executed when condition1 is true
 if (condition2) {
Statement 2; //executed when condition2 is true
}
 else {
Statement 3; //executed when condition2 is false
}
 }
```

Switch statement

Java switch statement compares the value and executes one of the case blocks based on the condition. It is same as if…else if ladder. Below are some points to consider while working with switch statements:



Flow diagram

Java Loop

- **While loop**

Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

- **For loop**

Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

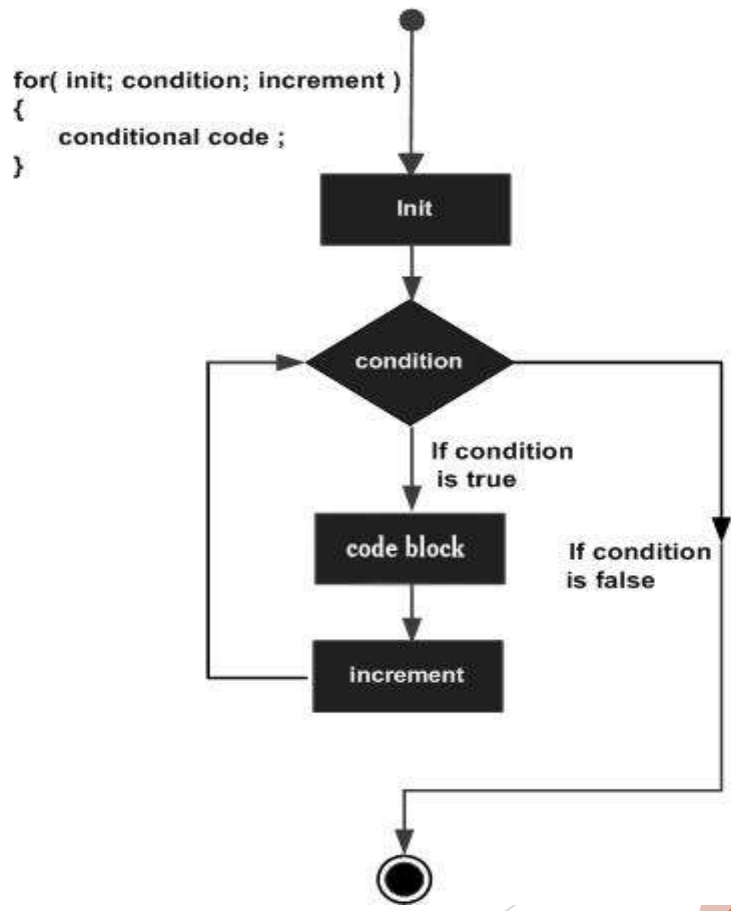
- **Do...while loop**

Like a while statement, except that it tests the condition at the end of the loop body.

For loop

When you know exactly how many times you want to loop through a block of code, use the for loop instead of a while loop:

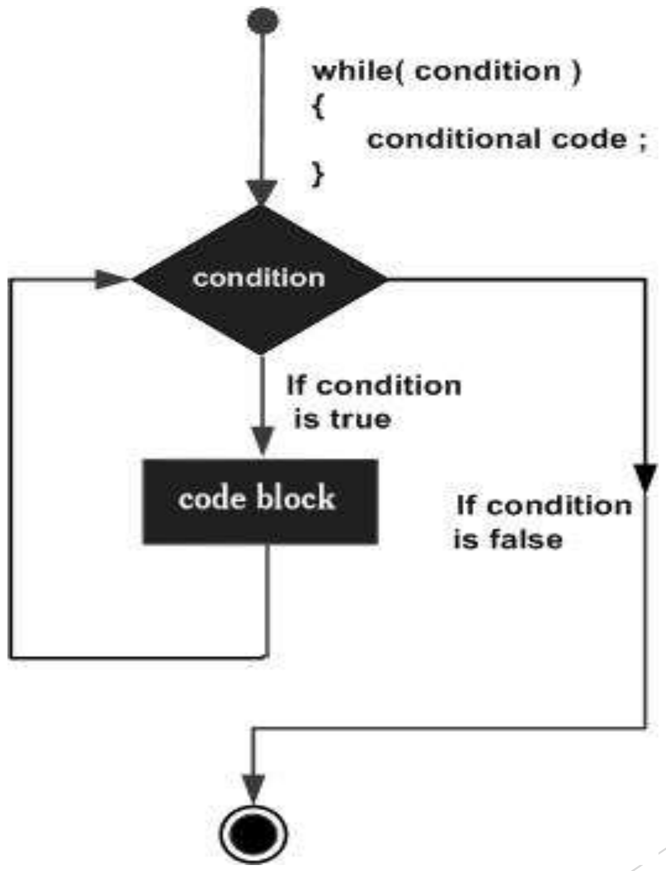
```
for (statement 1; statement 2; statement 3)
 {
 // code block to be executed
}
```



While loop

The while loop loops through a block of code as long as a specified condition is true:

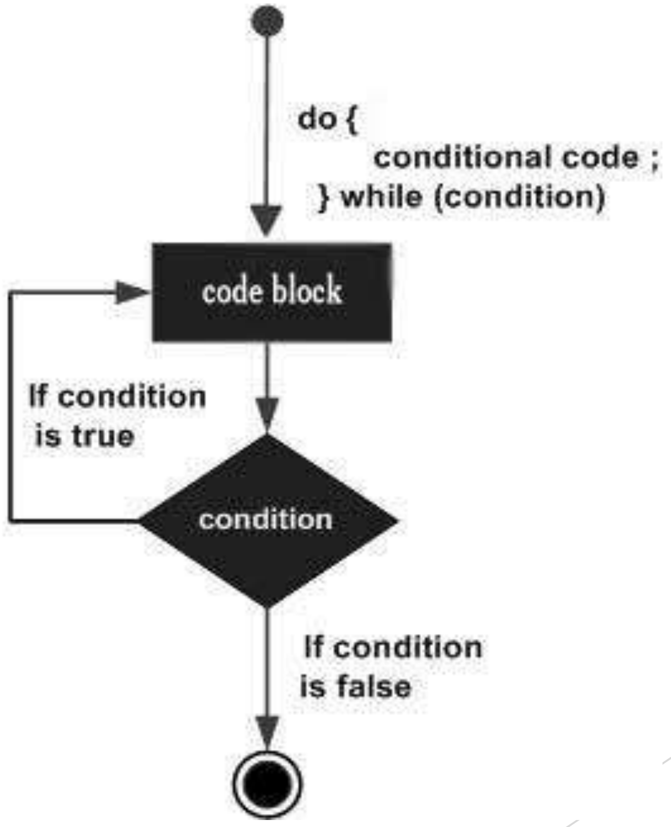
```
while (condition) {
// code block to be executed
}
```



Do...while loop

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true:

```
do {
// code block to be executed
}
while (condition);
```



UEM
UNIVERSITY OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Thank You