ShreeSoft INFORMATICS

Address: Tirumala Plaza, Upendra Nagar, Cidco, Nashik.

Join us on Telegram:
https://t.me/ShreeSoftinformatics

Join us on WhatsApp:
https://chat.whatsapp.com/BrLM0ZoWMbpA8nXi9Mrocl

Contact: +91 8308341531

Program Control Statement in Java

What is Program Control Statement?

A program written in Java programming language is generally executed by JVM sequentially (one by one) in the order in which they appear.

These statements are called sequential statements. The flow of execution takes place from top to bottom. But if we want to break the sequential execution then we can use Program control Statement.

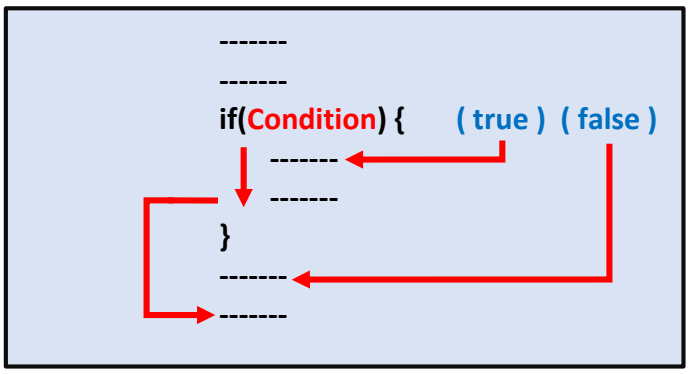
Types of Program Control Statement

1) **Unconditional Program control statement (Jumping Statement)**
   - break
   - continue
   - return
2) **Conditional Program Control Statement (Selection Statement)**
   - If
   - If else
   - switch
3) **Repetitive Program Control Statement (Loop Statement)**
   - while
   - do while
   - for
   - for-each **(introduced in java 1.5 version)**

Simple if:

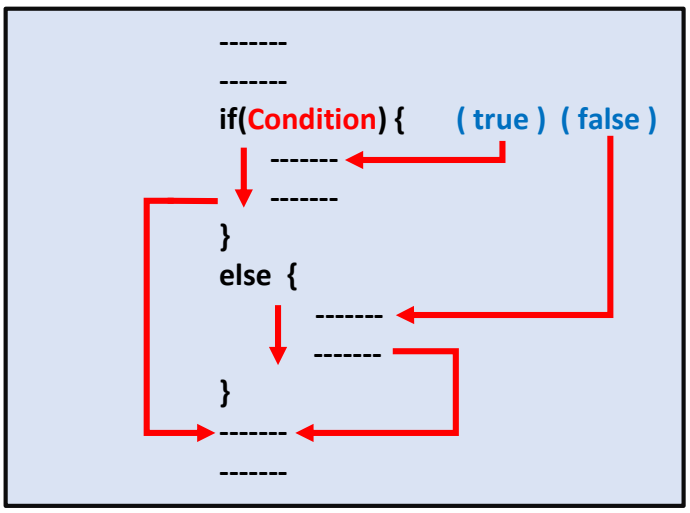
❒ We can use simple if control statement when we want to handle single decision.

❒ If the condition is true then it can be execute if block otherwise execute



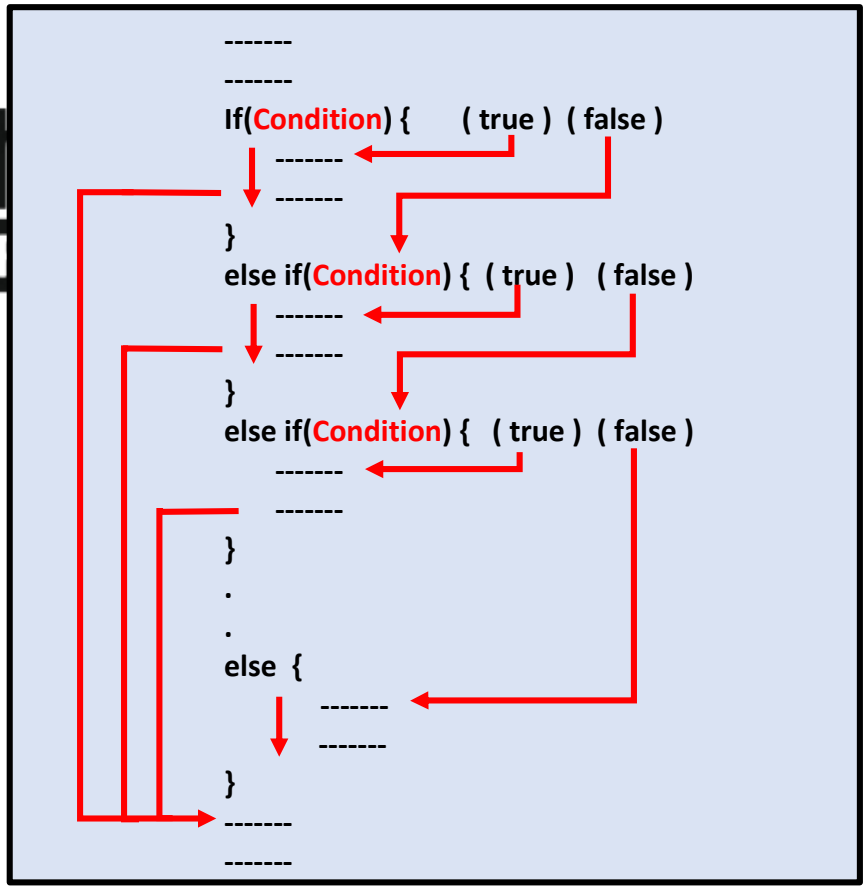
if else:

- We can use if else control statement when we want to handle two decisions.
- If the condition is true then execute if block otherwise it execute else block.
- It means that, in if else control statement either execute if block or else block. After that it execute other statements of



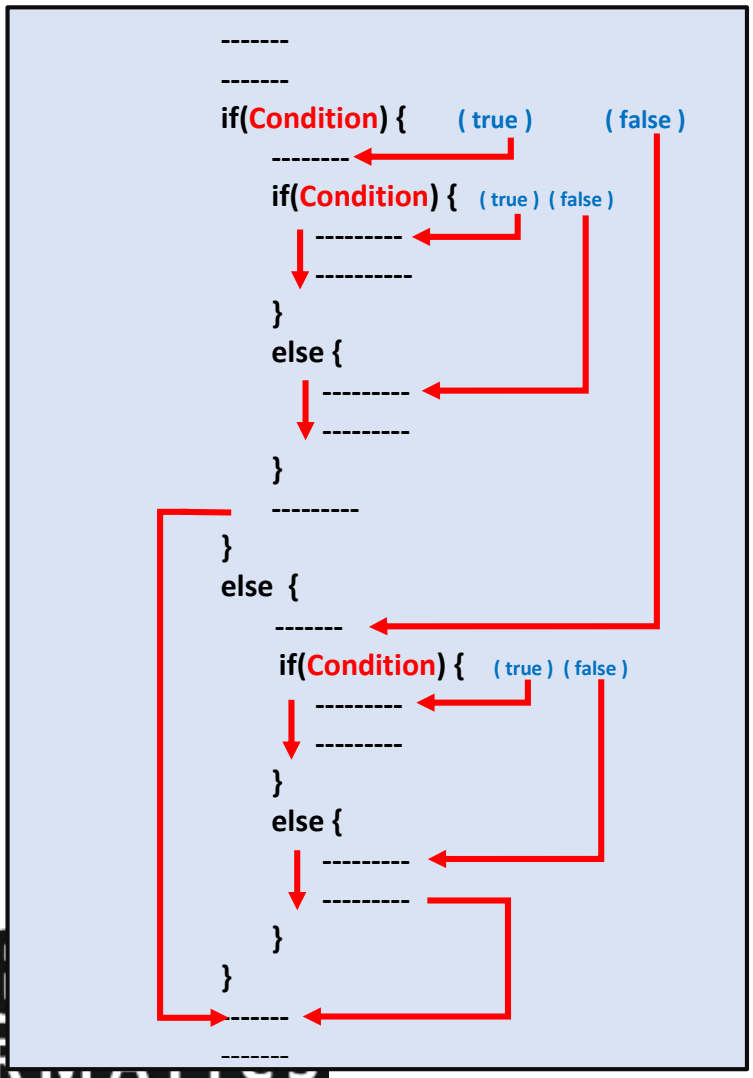
Ladder if else if:

- We can use Ladder if else if control statement when we want to handle multiple decisions.
- In the Ladder if else if there is multiple conditions. Whichever condition is become true that block will be executed.
- If any condition is not true then else block will be executed. After that other statements will be executed of program.



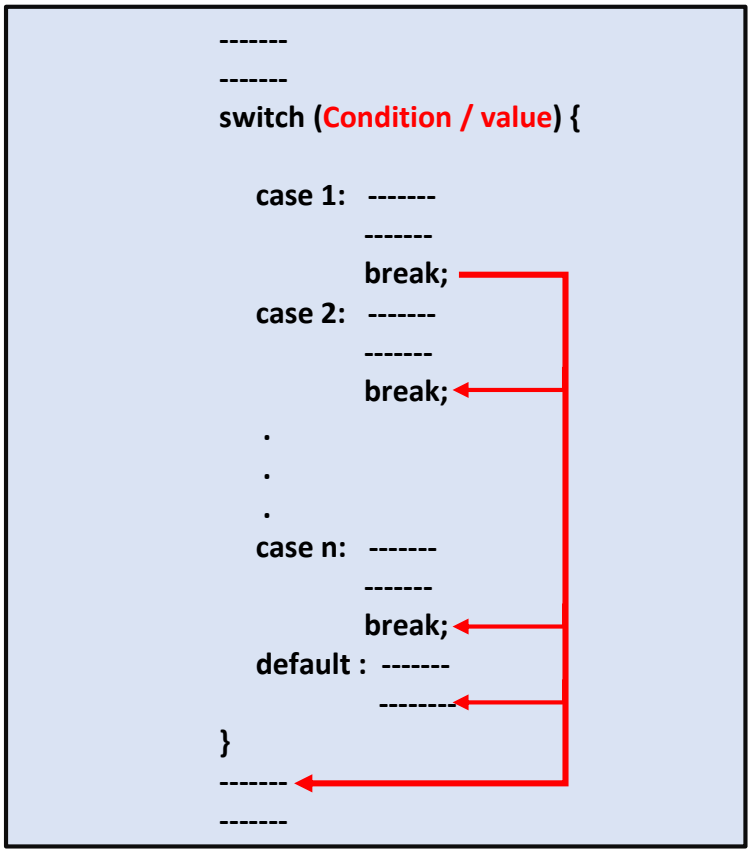
Nested if else:

One if else statement will be within another if else statement is called as Nested if else control statement.



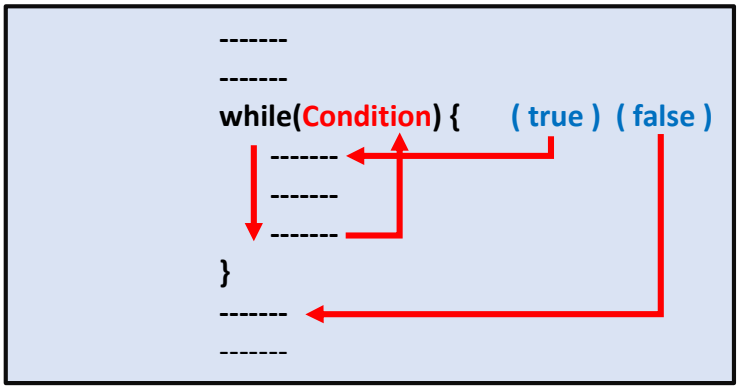
Switch case statement:

- Switch case is also used to handle multiple decisions. It is alternative of Ladder if else if.
- In switch case the value will be compared with case value and those case value will be equal or match then that block will be executed.
- break statement will be used to transfer program control out of switch case control statement.
- default case will be executed when the value will not be match with any case.



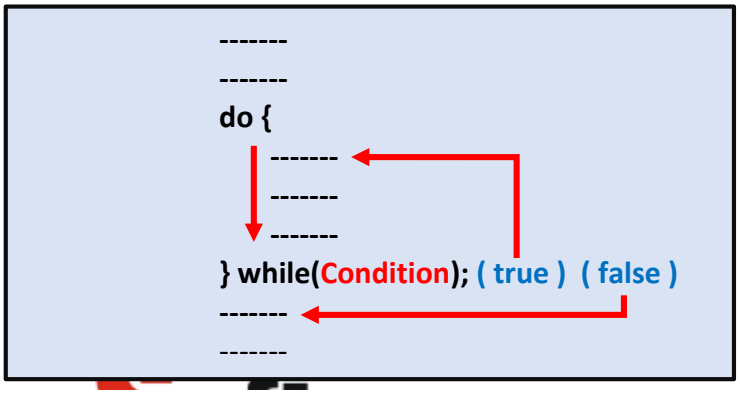
while Loop Statement:

We can use while loop when we want execute some group of statement repetitively. It means that to perform certain operation multiple times to get the result.



do while Loop Statement:

We can use do while loop also to execute some group of statement repetitively. It means that to perform certain operation multiple times to get the result.

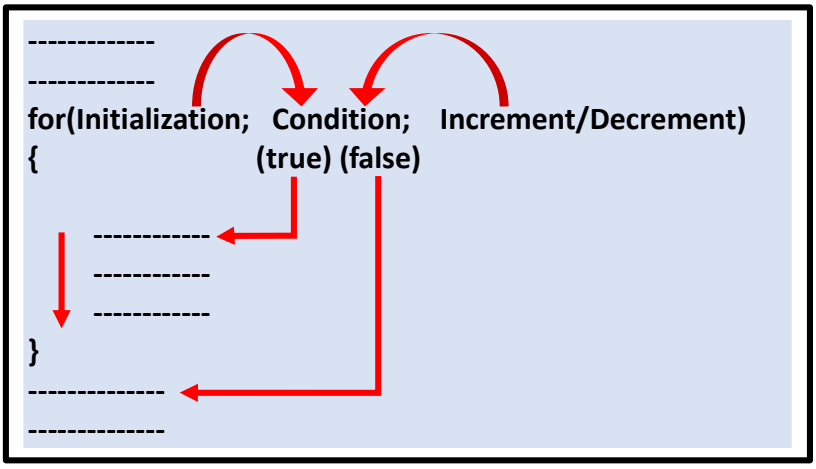


Difference between while and do while loop

while loop	do while loop
1. while is also called as Entry Controlled Loop.	1. do while loop is also called as Exit Controlled Loop.
2. In while loop, It first check condition and then execute the group statement.	2. In do while loop, It first execute the group statement and then checked condition.
3. It might occur statement(s) is executed zero times, If condition is false.	3. At least once the statement(s) is executed.
4. No semicolon at the end of while. while(condition)	4. Semicolon at the end of while. while(condition);
5. If there is a single statement, brackets are not required.	5. Brackets are always required.

For loop:

We can use for loop when we want execute some group of statement repetitively. Generally for loop will be used to iterate element from array.



break statement:

A break statement in Java is used to break a loop or switch statement. When a break statement encounters inside a loop statement, the loop immediately ends at a specified condition.

As a result, the program control continues the execution of the next statement immediately following the loop body.

Similarly, when break statement executes inside the switch block, it terminates 'case' inside the switch block and stops the execution of more cases inside the switch.

When the loops are nested, break statement breaks only the inner loop. We can use Java break statement in all types of loops such as for loop, while loop, and do-while loop.

Syntax: **break;**

We can also use the break statement with a label reference to break out of any code block.

Syntax: **break labelName;**

Example 1:

```
public class BreakExample1  {
        public static void main(String[] args) {
                // Using for loop.
                for(int i = 1; i <= 10; i++) {
                        if(i == 5) {
                            break; //  Breaking a loop.
                        }
                        System.out.println("I: " +i);
                 }
         }
}
```

Example 2: Program based on break statement with labelled name.

```
public class BreakExample2
{
        public static void main(String[] args)
        {
                for(int i = 1; i <= 3; i++)    // Outer for loop.
                {
                        bb:
                        for(int j = 1; j <= 3; j++)    // Inner for loop.
                        {
                                if(i == 2 && j == 2)
                                        break bb; // Using break statement with label.
                                System.out.println(i+" "+j);
                        }
                }
        }
}
```

continue statement:

Continue statement in Java is another similar statement, like break statement that is used inside a loop to repeat the next iteration of the loop.

In other words, continue statement stops the current iteration of loop and begins a new iteration of loop.

When the continue statement encounters, subsequent statements in the loop skips (i.e. not executed) at a specified condition, and the control of execution continues with the next repetition of the loop.

Continue statement in Java do not break out of the loop entirely. It just jumps back to the beginning of the loop by skipping the rest of code in the loop body for the next iteration.

Syntax: **continue;**

We can also use the break statement with a label reference to break out of any code block.

Syntax: **continue labelName;**

Example 1:

```
public class ContinueExample1 {
	public static void main(String args[]) {
		int i;
		for(i=1;i<=10;i++)
		{
			if(i%2==0)
				continue;
			System.out.println("i = "+ i);
		}
	}
}
```

Example 2: Program based on continue statement with labelled name.

```
public class ConsoleClassExample
{
	public static void main(String[] args)
	{
		int i, j;
		outerloop:
		for(i = 0; i < 3; i++)
		{
			for(j = 0; j < 3; j++)
			{
				if((i == 2) && (j == 2))
				{
					continue outerloop; // skips when both expressions are true.
				}
				else
				{
					System.out.println(i+ ", " +j+ " ");
				}
			}
		}
	}
}
```