P .Naga Divya

Nagadivya19402@gmail.com

Differences between the else-if ladder statement and Switch statement.

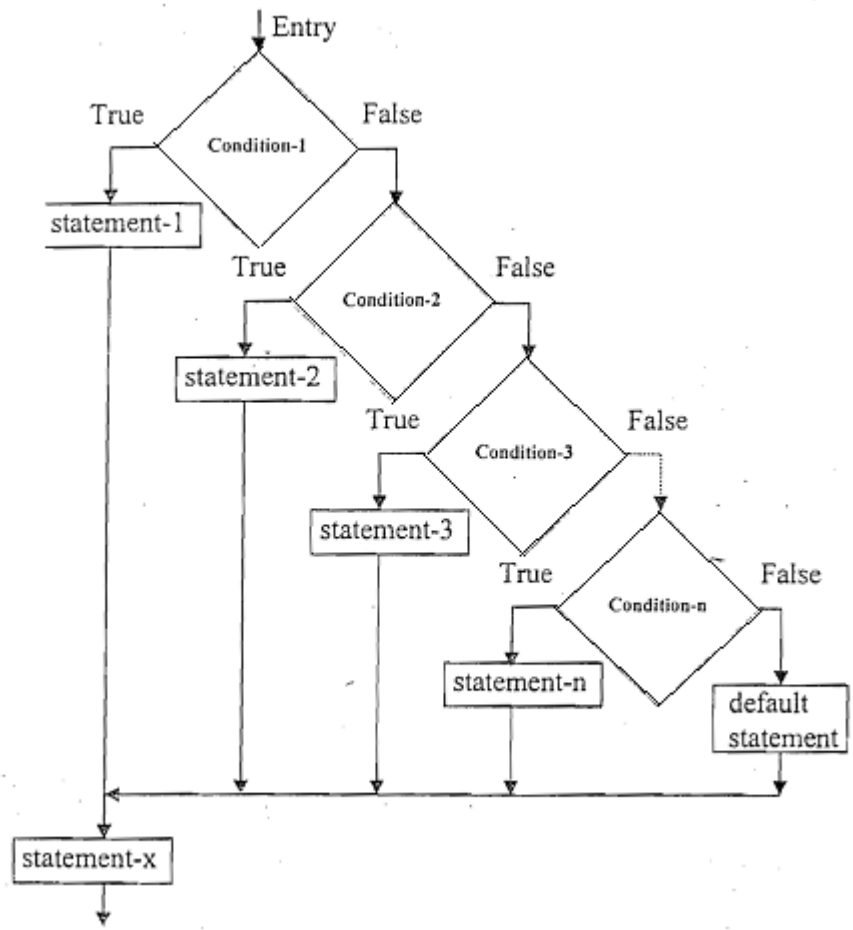
else-if ladder statement :

else if statement can be defined as a control statement which controls the statement(s) to be executed on the basis of some conditions. Whenever the else if statement is used, the compiler or interpreter initially checks the condition whether it is true or false and if the condition is found to be true then, the corresponding statements are executed. If the condition is found to be false, it continues checking the next *else if* statement until the condition comes to be true or the control comes to the end of the *else if ladder*.

else-if ladder statement can be represented as:

```
if( condition-1)
   statement-1;
else if (condition-2)
    statement-2;
else if (condition-3)
    statement-3;
else if (condition-4)
    statement-4;
else if (condition-n)
    statement-n;
else
    default statement;
```

Below is a flowchart that represents *else if ladder*.



Switch Statement:

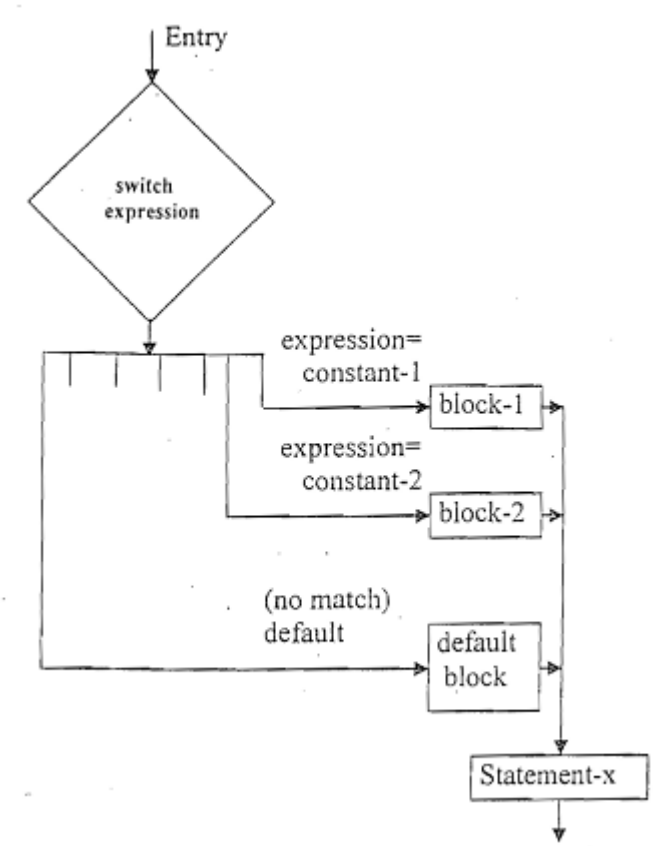
The *switch case* statement is similar to the else-if ladder as it provides multiple branching or multi-conditional processing. But, the basic difference between switch case and else if ladder is that the *switch case* statement tests the value of variable or expression against a series of different cases or values, until a match is found. Then, the block of code with in the match case is executed. If there are no matches found, the optional default case is executed.

syntax of switch case can be represented as:

```
switch(expression)

{

   case constant-1

      block-1;

      break;

   case constant-2
```

```
        block-2;
        break;
    case constant-3
        block-3;
        break;
    case constant-n
        block-n;
        break;
    default:
        default_block;
}
statement-x;
```

Below is a flowchart that represents *switch case*.



Differentiate between else-if ladder statement and switch statement.

BASIS OF COMPARISON	**ELSE IF LADDER**	**SWITCH CASE**
The control	In else if ladder, the control runs through the every else if statement until it arrives at the true value of the statement or until it comes to the end of the else if ladder.	In else if ladder, the control runs through the every else if statement until it arrives at the true value of the statement or until it comes to the end of the else if ladder.
Working	Else if ladder statement works on the basis of true false (zero/non-zero) basis.	Switch case statement work on the basis of equality operator.
Use of Break Statement	In switch, the use of break statement is mandatory and very important.	In else if ladder, the use of break statement is not very essential.
Variable Data	Integer is the only variable data type that can be in expression of switch.	Either integer or character is the variable data type used in the expression of else if ladder.

Processing Of Codes	In the case of else if ladder, the code needs to be processed in the order determined by the programmer.	In switch case, it is possible to optimize the switch statement, because of their efficiency. Each case in switch statement does not depend on the previous one.
Flexibility	Else if statement is not flexible because it does not give room for testing of a single expression against a list of discrete values.	Switch case statement is flexible because it gives room for testing of a single expression against a list of discrete values.
Usage	Else if ladder is used when there is multiple conditions are to be tested.	Switch case is used when there is only one condition and multiple values of the same are to be tested.
Values	Values are based on constraint.	Values are based on user choice.

Nested if condition

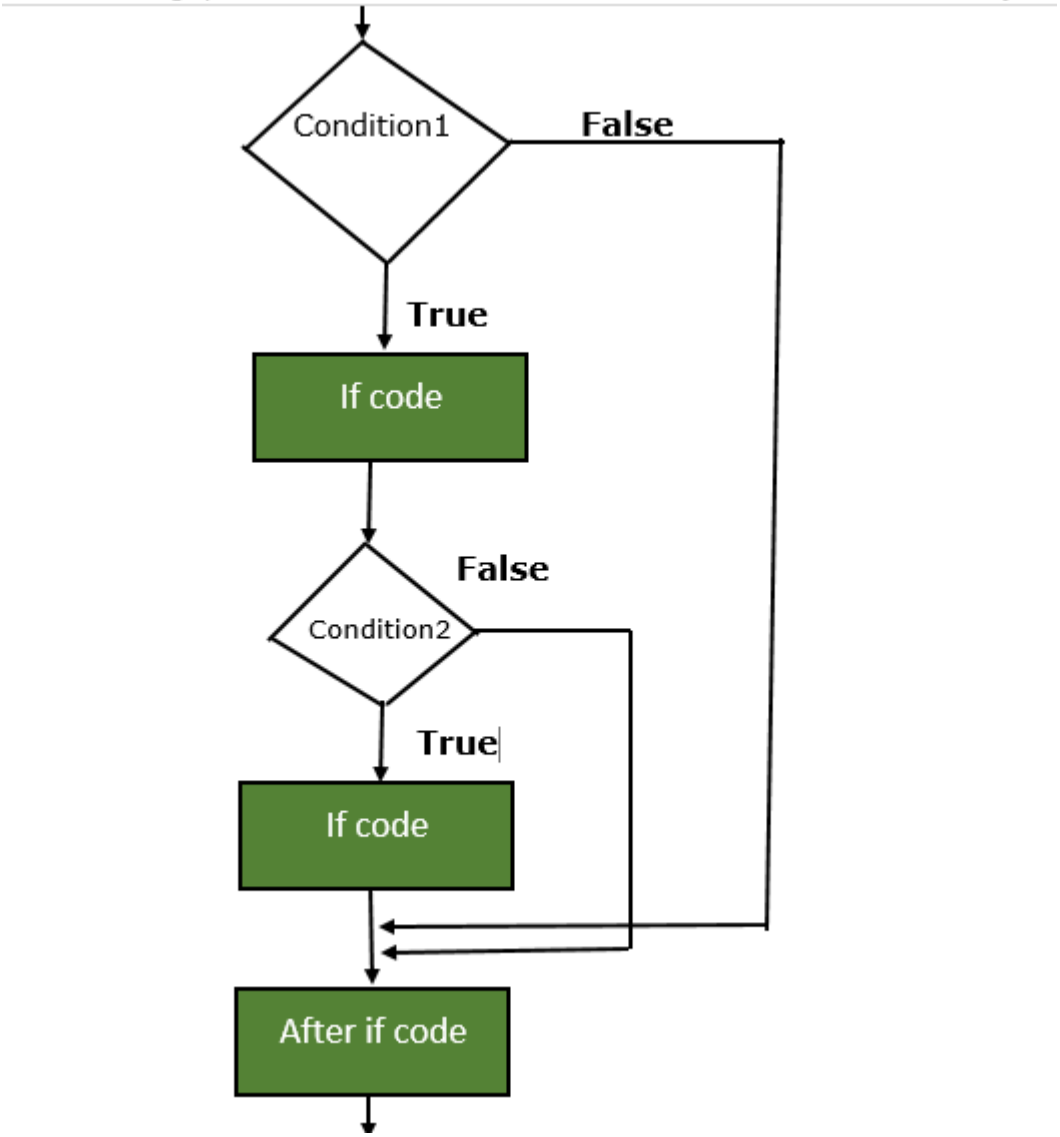
Nested means within. Nested if condition means if-within-if. Nested if condition comes under **decision-making statement in Java**. There could be infinite if conditions inside an if condition. The below syntax represents the Nested if condition.

Syntax:

```
if( condition ){

    if( condition ){

        if( condition ){

            ......
        }
```

```
    }
}
```

Nested If flow chart



Nested if-else statement

Java allows programmer to place if else block inside another if or else block. This is called as nested if else statement. Programmer can do any level of nesting in program which means you can place if else block inside another if or else block up to any number of times.

```
if(condition-1) {
```

```
  // code to be executed inside first if block
  if(condition-2) {
    // code to be executed inside second if block
    if(condition-3) {
      // code to be executed inside third if block
      . . . . .
      . . . . .
    }
    else {
      // code to be executed inside else block
    }
    // code to be executed inside second if block
  }
   // code to be executed inside first if block
}
else {
 if(condition-4) {
   // code to be executed inside this if block
 }
```

```
}
```