DECISION STATEMENTS

The if-else statement in C++ is used to perform the operations based on some specific condition. The operations specified in if block are executed if and only if the given condition is true.

There are the following variants of if statement in C++ language.

- If statement
- If-else statement
- If else-if ladder
- Nested if

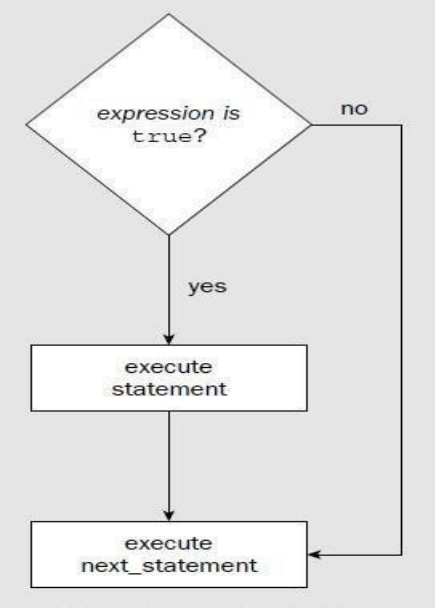
If statement: Syntax

if(expression)

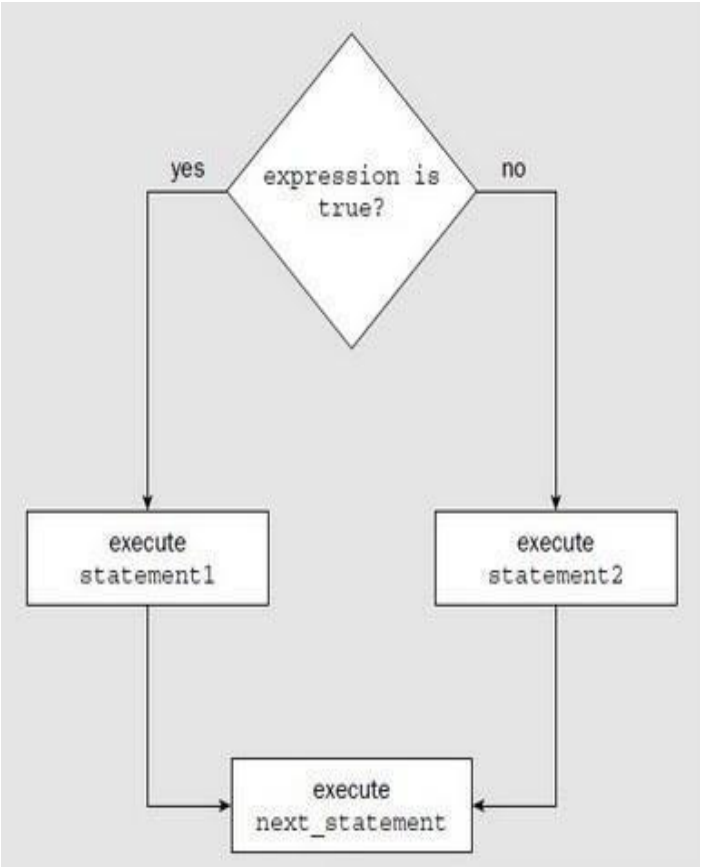
statement1;

Explanation :

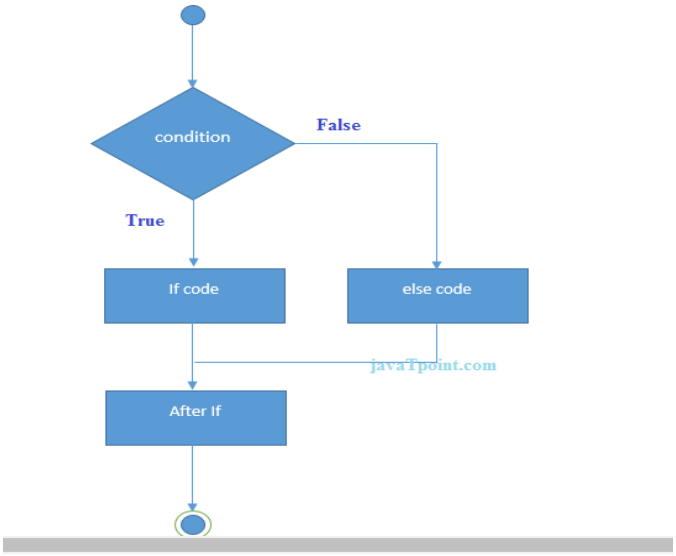
- Expression is Boolean Expression It may have true or false value



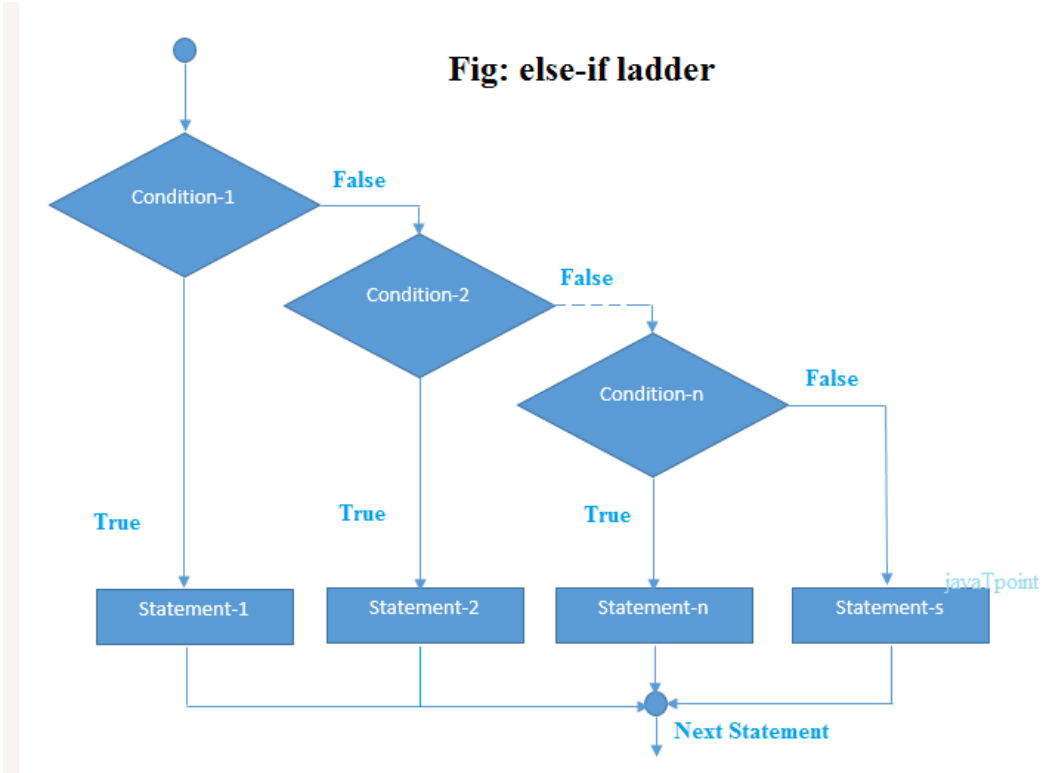
If- Else Statement :Bi Condition



NESTED CONDITION FLOWCAHRT:



LADDER CONDITION FLOWCAHRT:



SWITCH STATEMENT

Rules for switch statement :

1) The *switch expression* must be of an integer or character type.

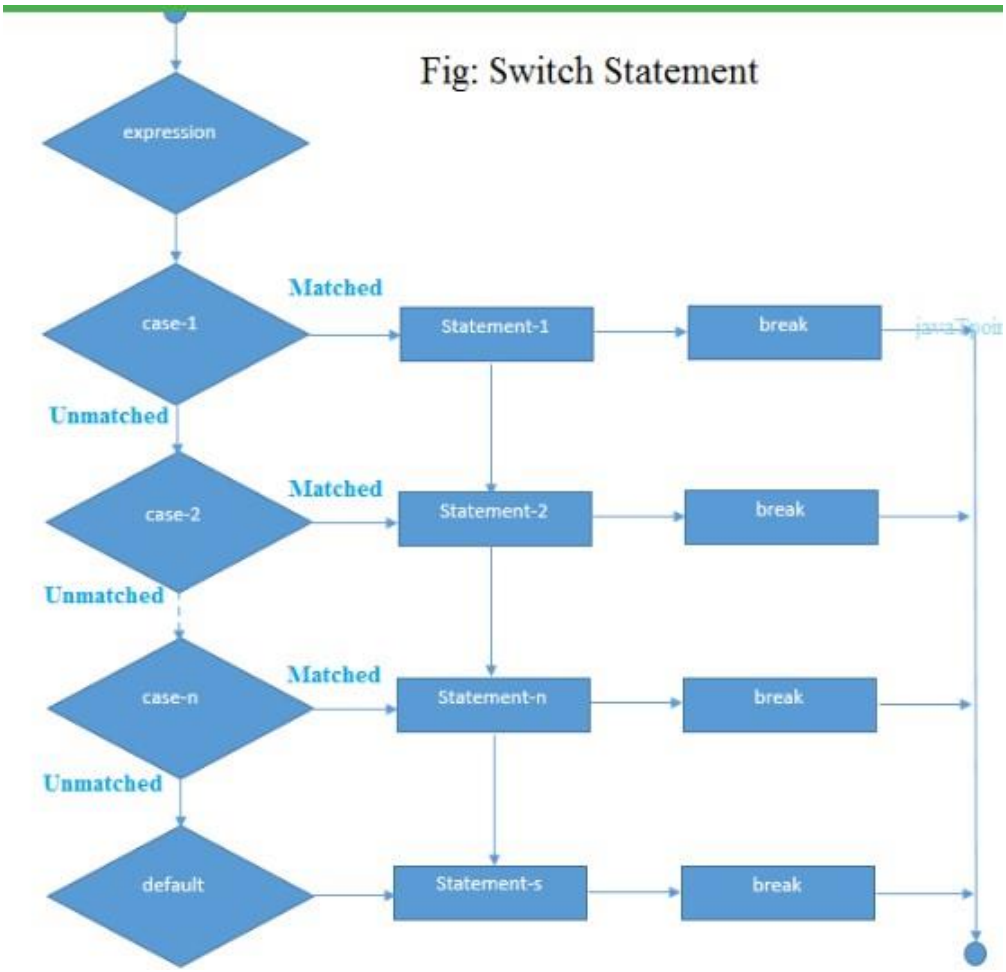
2) The *case value* must be an integer or character constant.

3) The *case value* can be used only inside the switch statement.

4) The *break statement* in switch case is not must. It is optional. If there is no break statement found in the case, all the cases will be executed present after the matched case. It is known as *fall through* the state of C++ switch statement.

Valid Switch	Invalid Switch	Valid Case	Invalid Case
switch(x)	switch(f)	case 3;	case 2.5;
switch(x>y)	switch(x+2.5)	case 'a';	case x;
switch(a+b-2)		case 1+2;	case x+2;
switch(func(x,y))		case 'x'>'y';	case 1,2,3;

Fig: Switch Statement



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.

SYNTAX:

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

do/while Flow Chart:

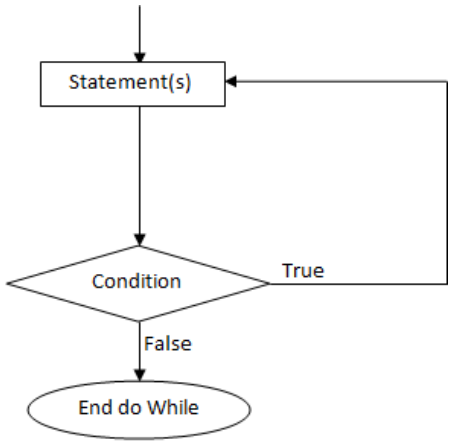


fig: Flowchart for do-while loop

EXAMPLE:

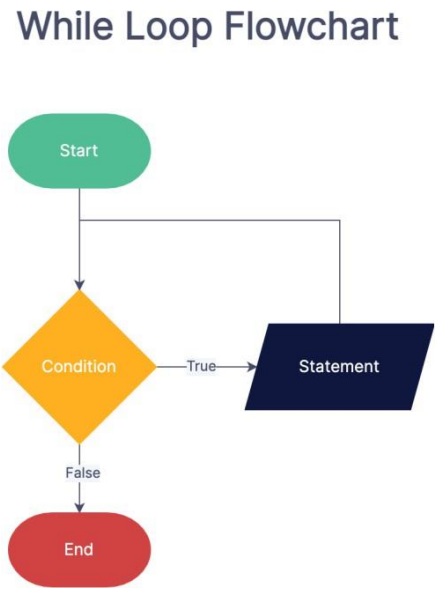
```
int i = 0;
do {
  cout << i << "\n";
  i++;
}
while (i < 5);
```

While Loop

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

SYNTAX:

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



EXAMPLE:

```
int i = 0;
while (i < 5) {
  cout << i << "\n";
  i++;
}
```

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:

SYNTAX:-

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

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

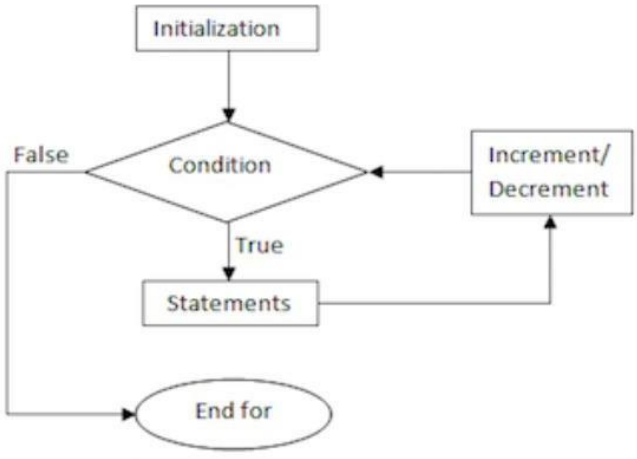


fig: Flowchart for for loop

EXAMPLE:

```
for (int i = 0; i < 5; i++)
{
  cout << i << "\n";
}
```

****