Flow of Control

Statements

Statements are the instructions given to the computer to perform any kind of action. Action may be in the form of data movement, decision making etc. Statements form the smallest executable unit within a C++ program. Statements are always terminated by semicolon.

Compound Statement

A compound statement is a grouping of statements in which each individual statement ends with a semi-colon. The group of statements is called block. Compound statements are enclosed between the pair of braces ({}.). The opening brace ({) signifies the beginning and closing brace (}) signifies the end of the block.

Null Statement

Writing only a semicolon indicates a null statement. Thus ';' is a null or empty statement. This is quite useful when the syntax of the language needs to specify a statement but the logic of the program does not need any statement. This statement is generally used in for and while looping statements.

Conditional Statements

Sometimes the program needs to be executed depending upon a particular condition. C++ provides the following statements for implementing the selection control structure.

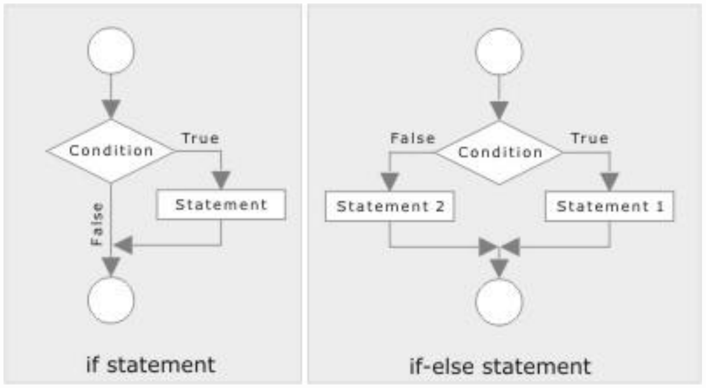
- if statement
- if else statement
- nested if statement
- switch statement

if statement

syntax of the if statement

```
if (condition)
{
  statement(s);
}
```

From the flowchart it is clear that if the if condition is true, statement is executed; otherwise it is skipped. The statement may either be a single or compound statement.



if else statement

syntax of the if - else statement

```
if (condition)
  statement1;
else
  statement2;
```

From the above flowchart it is clear that the given condition is evaluated first. If the condition is true, statement1 is executed. If the condition is false, statement2 is executed. It should be kept in mind that statement and statement2 can be single or compound statement.

if example	if else example

`if (x == 100)`
`    cout << "x is 100";`
	`if (x == 100)`
	`    cout << "x is 100";`
	`else`
	`    cout << "x is not 100";`

Nested if statement

The if block may be nested in another if or else block. This is called nesting of if or else block.

syntax of the nested if statement

```
if(condition 1)
{
  if(condition 2)
  {
    statement(s);
  }
}
```

```
if(condition 1)
  statement 1;
else if (condition 2)
  statement2;
else
  statement3;
```

if-else-if example

```
if(percentage>=60)
     cout<<"Ist division";
else if(percentage>=50)
     cout<<"IInd division";
else if(percentage>=40)
     cout<<"IIIrd division";
else
     cout<<"Fail" ;
```

switch statement

The if and if-else statements permit two way branching whereas switch statement permits multiple branching. The syntax of switch statement is:

```
switch (var / expression)
{
   case constant1 : statement 1;
   break;
   case constant2 : statement2;
   break;
    .
    .
   default: statement3;
   break;
}
```

The execution of switch statement begins with the evaluation of expression. If the value of expression matches with the constant then the statements following this statement execute sequentially till it executes break. The break statement transfers control to the end of the switch statement. If the value of expression does not match with any constant, the statement with default is executed.

Some important points about switch statement

- The expression of switch statement must be of type integer or character type.
- The default case need not to be used at last case. It can be placed at any place.
- The case values need not to be in specific order.