C# Basic- Operators

By Dhananjay Masal– CODEMIND Technology

Contact us +91 9890217463

Control Flow Statements in C#

statements that Alter the Flow of Program Execution and provide better control to the programmer on the flow of execution. The Control Flow Statements are useful to write better and more complex programs. A program executes from top to bottom except when we use control statements. We can control the order of execution of the program, based on logic and values.

Types of Control Flow Statements in C#:

1. Selection Statements or Branching Statements: (Examples: if-else, switch case, nested if-else, if-else ladder)
2. Iteration Statements or Looping Statements: (Examples: while loop, do-while loop, for-loop, and foreach loop)
3. Jumping Statements: (Examples: break, continue, return, goto)

If-Else Statements in C#

It executes a block of statements (one or more instructions) when the condition in the if block is true and when the condition is false, it will skip the execution of the if block. Using else block is optional in C#. Following is the syntax to use the if block in the C# language.

```
if(condition) {
  Statements;   // true
} else {
  Statements;  // false
}
```

Is it mandatory to use else block?

- No, it is not mandatory to use else block. It is an optional block. You can use only if block also. If you want to provide some information when the condition failed, then you need to use this optional else block.

Nested If-Else Statements in C# Language:

1. When an if-else statement is present inside the body of another if or else then this is called nested if-else. Nested IF-ELSE statements are used when we want to check for a condition only when the previous dependent condition is true or false.

```
if(condition1){          if(condition2){

            } } else {

}
```

Switch Statements in C#

- using this switch keyword we can create selection statements with multiple blocks. And the Multiple blocks can be constructed by using the case keyword
- Switch case statements in C# are a substitute for long if else statements that compare a variable or expression to several values.

Switch

```
switch(variable) {
        case 1:
                // execute code;
        break;
        case 2:
                // execute code;
        break;
        default:
                // execute code;
        break;
}
```

Thank You

Success is not a milestone, it's a journey. And we have vowed to help you in yours.

www.codemindtechnology.com