



**IT LEGEND**

Create legends Programming

# C# Statements and Control Flow



| [www.itlegend.net](http://www.itlegend.net)



| 01013855500



| [Ceo@itlegend.net](mailto:Ceo@itlegend.net)

# Table of Content

- What is C# Statements ?
- What is C# Blocks?
- Types of Statements
- What is C# Expressions?
- Operators
- Simple Statements
- Compound Statements
- Control Statements
  - Selection Statements
  - Jump Statements
  - Loops Statements
  - Exception Handling Statements



# What is C# Statements?

- A statement is a basic unit of execution of a program, and the program consists of multiple statements.
- There is many types of statements like
  - Declaration Statement
  - Expression Statement
  - (there are other types we will cover them later)



# types of statements

Declaration

Expresion

Control Flow

Method Calling



# C# Statements Examples

Ex.

```
int age = 21;  
int marks = 90;
```

These both lines are statements



# C# Statements Examples

## Declaration Statement

Declaration statements are used to declare and initialize variables.

```
char ch;  
int maxValue = 55;
```

Both char ch;  
and int maxValue = 55;  
are declaration statements.

## Expression Statement

An expression followed by a semicolon is called an expression statement.

```
area = 3.14 * radius * radius;  
/* Method call is an expression*/  
System.Console.WriteLine("Hello");
```

Here, `3.14 * radius * radius` is an expression,  
`area = 3.14 * radius * radius;` is an expression statement.

`System.Console.WriteLine("Hello");` is both an expression and a statement.



# What is C# Blocks?

- A block is a combination of zero or more statements that is enclosed inside curly brackets { }



# C# Block Examples

```
using System;

namespace Blocks
{
    class BlockExample
    {
        public static void Main(string[] args)
        {
            double temperature = 42.05;
            if (temperature > 32)
            {
                // Start of block
                Console.WriteLine("Current temperature = {0}", temperature);
                Console.WriteLine("It's hot");
            }
            // End of block
        }
    }
}
```

# Types of Statements

## Simple Statements

Are any expression that terminates with a semicolon

EX.

```
var1 = var2 + var3;
```

Such as,

Declaration  
Statements

## Compound Statements

Related statements can be grouped together in braces to form a compound statement or block

EX. {

```
    int i = 4;  
    Console.WriteLine (i);
```

Such as,

Blocks

## Control Statements

Are the statements that can change the flow of execution

Loops  
Statements

Jump  
Statements

Conditions  
Statements

Exception  
Handling  
Statements



# What is C# Expressions?

- An expression in C# is a combination of operands(variables, literals, **Method calls**) and operators that can be evaluated to a single value. To be precise, an expression must have at least one operand but may not have any operator.



# C# Expression Examples

Ex.1

```
double temperature;  
temperature = 42.05;  
  
int a, b, c, sum;  
sum = a + b + c;
```

Here, 42.05 is an expression.  
Also, temperature = 42.05 is an expression too.  
There, a + b + c is an expression

Ex.2

```
if (age>=18 && age<58)  
    Console.WriteLine("Eligible to work");
```

Here, (age>=18 && age<58) is an expression  
that returns a Boolean value.  
"Eligible to work" is also an expression.



# Operators

Arithmetic

Relational

Logical

Assignment



# Arithmetic Operators

Assume variable A holds 10 and variable B holds 20

Operator	Description	Example
+	Adds two operands	$A + B = 30$
-	Subtracts second operand from the first	$A - B = -10$
*	Multiplies both operands	$A * B = 200$
/	Divides numerator by de-numerator	$B / A = 2$
%	Modulus Operator and remainder of after an integer division	$B \% A = 0$
++	Increment operator increases integer value by one	$A++ = 11$
--	Decrement operator decreases integer value by one	$A-- = 9$



# Arithmetc Operators Example

```
using System;
public class ArithmaticOperators
{
    static void Main(string[] args) {
        int a = 21;
        int b = 10;
        int c;

        c = a + b;
        Console.WriteLine("Line 1 - Value of c is {0}", c);

        c = a - b;
        Console.WriteLine("Line 2 - Value of c is {0}", c);

        c = a * b;
        Console.WriteLine("Line 3 - Value of c is {0}", c);
    }
}
```

Note:

Rest of code in the next slide



# Arithmetic Operators Example

```
c = a / b;  
Console.WriteLine("Line 4 - Value of c is {0}", c);  
  
c = a % b;  
Console.WriteLine("Line 5 - Value of c is {0}", c);  
  
c = a++;  
Console.WriteLine("Line 6 - Value of c is {0}", c);  
  
c = a--;  
Console.WriteLine("Line 7 - Value of c is {0}", c);  
Console.ReadLine();  
}  
}
```

## Output:

```
Line 1 - Value of c is 31  
Line 2 - Value of c is 11  
Line 3 - Value of c is 210  
Line 4 - Value of c is 2  
Line 5 - Value of c is 1  
Line 6 - Value of c is 21  
Line 7 - Value of c is 22
```

