**C# BASICS**

**Evolution and Overview of .Net:**

* Microsoft started the development of the. NET framework in the late 1990s, originally under the Next Generation Windows Services, and by late 2000, the first beta version of the . NET Framework was released.
* Used to develop various software applications.
* The class libraries present in this framework is known as FCL(Framework Class Library).
* Execution takes place in environment called CLR –Common Language runtime.
* .Net supports 60+ prgrmng langs.

**Overview of Visual Studio:**

* Visual studio is an IDE (Integrated Development Environment) used to develop application,games and others.
* It is product of Microsoft developed in 2000.
* It has capability to support 36+ programming langs.

Features of Visual Code:

**3 Types of Applications**

1.**Console Application** : A console program on the other hand is a text application. There are not fancy controls like buttons or textboxes in a console application and they are run from the command prompt. They are commonly used for test applications

2.**Windows Application** :A Windows form application is an application that has a graphical user interface(GUI) like the Visual C# IDE.  Microsoft word is an example of a Windows application.

3.**Web Application** : Web applications include online forms, shopping carts, word processors, spreadsheets, video and photo editing, file conversion, file scanning, and email programs such as Gmail, Yahoo and AOL. Popular applications include Google Apps and Microsoft 365**.(ASP.NET Web Application->**template(EMPTY,SinglePage,WebAPI,WebForms,MVC) when we select a webform we get a Default.aspx **)**

**NOTE**

* **Active Server Pages** (ASPX) is a file format used by web servers and generated using the Microsoft ASP.NET framework - an open-source development framework used by web developers to create dynamic web pages, often with the . NET and C# programming languages.
* ASP stands for Active Server Pages and also have capability to dynamically produce web pages based on a specific request from the client. **HTML uses tag to write its code which is interpreted by the web browsers to display the content which includes images and objects to be embedded in the webpage**.
* Solution Explorer is **a special window that enables you to manage solutions, projects, and files**. It provides a complete view of the files in a project, and it enables you to add or remove files and to organize files into subfolders.
* Using Keyword:2 ways to use
* Using directive: 1.using namespace;

2.using static namespace;

3.using N = namespace;

* Using statement: using (var b= new SqlConnection()){//some operation on b}
* In C#, namespaces are used to logically arrange classes, structs, interfaces, enums and delegates. The namespaces in C# can be nested. That means one namespace can contain other namespaces also. namespaces are called pre defined libraries
* Main Method():The Main() method is **the entry point a C# program from where the execution starts**. Main() method must be static because it is a class level method. To invoke without any instance of the class it must be static. Non-static Main() method will give a compile-time error.
* Hierarchy: namespace->class->main() or method->statements.
* Class as different methods(cube symbol),properties(span symbol),events(thunder symbol).

1.C# is case sensitive.

2.Every statement should end with terminator;

3.We cannot used uninitialized value types;

Comments:

* Single line comment: //
* Multi line comment /\* \*/

**Console Class**

Represents the standard input, output, and error streams for console applications. This class cannot be inherited.

* WriteLine() method, which is used to print the entire string on a simple line.
* Write():Same line
* ReadLine()method is used to read group of characters, reads string value of character, returns string
* Read():reads only one character, reads ascii value of character, returns integer
* \n-newline \b backspace
* Clear()-Clears the console

**Variables:**

 Variable is a placeholder of the information which can be changed at runtime. And variables allows to **Retrieve and Manipulate** the stored information.

Syntax:type variable\_name = value;

**Rules for Naming Variables**

* Variable names can contain the letters ‘a-z’ or ’A-Z’ or digits 0-9 as well as the character ‘\_’.
* The name of the variables cannot be started with a digit.
* The name of the variable cannot be any C# keyword say int, float, null, String, etc
* If you want to use reserved keyword as an identifier include @ before variable name.

**DataType**

* Data types specify the type of data that a valid [C#](https://www.geeksforgeeks.org/introduction-to-c-sharp/) variable can hold.
* **(Basic Datatype)Value Data Types:** In [C#](https://www.geeksforgeeks.org/introduction-to-c-sharp/), the Value Data Types will directly store the variable value in memory and it will also accept both signed and unsigned literals. The derived class for these data types are **System.ValueType**.

1.Numbers :

1st: Integer

* Byte -8 bit
* Short -16 bit
* Int -32 bit
* Long – 64 bit

2nd Decimal

* Float – 32 bit
* Double -64 bit
* Decimal -128 bit

Char

* Char- 8 bit(single quote)

Boolean

* Bool -8 bit
* **Reference Data Types** The Reference Data Types will contain a memory address of variable value because the reference types won’t store the variable value directly in memory. The built-in reference types are **string(double quote), object.**
* **Pointer Data Type:** The Pointer Data Types will contain a memory address of the variable value.  
  To get the pointer details we have a two symbols **ampersand (&) and asterisk (\*)**.  
  **ampersand (&):** It is Known as Address Operator. It is used to determine the address of a variable.  
  **asterisk (\*):** It also known as Indirection Operator. It is used to access the value of an address.

**Operators:**

 Operators allow us to perform different kinds of operations on **operands**

In C#, Operators can also categorized **based upon Number of Operands :**

* **Unary Operator:** Operator that takes **one** operand to perform the operation.increment,pre-increment,decrement,pre-decrement.
* **Binary Operator:** Operator that takes **two** operands to perform the operation.
* **Ternary Operator:** Operator that takes **three** operands to perform the operation.
* [**Arithmetic Operators**](https://www.geeksforgeeks.org/c-sharp-operators/#Arithmetic%20Operators)**:** The **Binary Operators** falling in this category  + - \* / (quotient)%(remainder)
* [**Relational Operato**rs](https://www.geeksforgeeks.org/c-sharp-operators/#Relational%20Operators): **‘=='(Equal To), ‘!='(Not Equal To), ‘>'(Greater Than), ‘<‘(Less Than), ‘>='(Greater Than Equal To), ‘<='(Less Than Equal To)**
* [**Logical Operators**](https://www.geeksforgeeks.org/c-sharp-operators/#Logical%20Operators):&&, ||,!
* [**Bitwise Operators**](https://www.geeksforgeeks.org/c-sharp-operators/#Bitwise%20Operators):&(Bitwise and),|(Bitwise or),^(bitwise XOR),<<(left shift),>>(right shift)
* [**Assignment Operators**](https://www.geeksforgeeks.org/c-sharp-operators/#Assignment%20Operators)**:** =,+=,-=,\*=,/=,%=
* [**Conditional Operator**](https://www.geeksforgeeks.org/c-sharp-operators/#Conditional%20Operator)**:** (? :)

**Conditions:**

1**. If condition:** if statement to specify a block of C# code to be executed if a condition is True.

Use the else statement to specify a block of code to be executed if the condition is False.

if (condition)

{

// block of code to be executed if the condition is True

}

else

{

// block of code to be executed if the condition is False

}

**Note:**A breakpoint, is **an intentional stop marked in the code of an application where execution pauses for debugging**. This allows the programmer to inspect the internal state of the application at that point.

switch statement to select one of many code blocks to be executed.

switch(expression)

{

case x:

// code block

break;

case y:

// code block

break;

default:

// code block

break;

}

* The switch expression is evaluated once
* The value of the expression is compared with the values of each case
* If there is a match, the associated block of code is executed
* The break it breaks out of the switch block.
* This will stop the execution of more code and case testing inside the block.
* When a match is found, and the job is done, it's time for a break.There is no need for more testing.
* The default keyword is optional and specifies some code to run if there is no case match.

**Forloop**

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.

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.

* Loops can execute a block of code as long as a specified condition is reached.

**While Loop**

The while loop loops through a block of code as long as a specified condition is True.

while (condition)

{

*// code block to be executed*

}

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.

do

{

*// code block to be executed*

}

while (condition);