**C# Tutorial:**

1. **What is C#:**

C# is pronounced as "C-Sharp". It is an object-oriented programming language provided by Microsoft that runs on .Net Framework.

By the help of C# programming language, we can develop different types of secured and robust applications:

Window applications, Web applications, Distributed applications , Web service applications, Database applications etc.

The most recent stable version of the language is C# 12.0, which was released in 2023 in .

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**02. C# Example: Hello World**

In C# programming language, a simple "hello world" program can be written by multiple ways. Let's see the top 4 ways to create a simple C# example:

## C# Simple Example

**class** Program

{

**static** **void** Main(string[] args)

{

System.Console.WriteLine("Hello World!");

}

}

### Description

**class:** is a keyword which is used to define class.

**Program:** is the class name. A class is a blueprint or template from which objects are created. It can have data members and methods. Here, it has only Main method.

**static:** is a keyword which means object is not required to access static members. So it saves memory.

**void:** is the return type of the method. It does't return any value. In such case, return statement is not required.

**Main:** is the method name. It is the entry point for any C# program. Whenever we run the C# program, Main() method is invoked first before any other method. It represents start up of the program.

**string[] args:** is used for command line arguments in C#. While running the C# program, we can pass values. These values are known as arguments which we can use in the program.

**System.Console.WriteLine("Hello World!"):** Here, System is the namespace. Console is the class defined in System namespace. The WriteLine() is the static method of Console class which is used to write the text on the console.

## C# Example: Using System

If we write using System before the class, it means we don't need to specify System namespace for accessing any class of this namespace. Here, we are using Console class without specifying System.Console.

using System;

**class** Program

{

**static** **void** Main(string[] args)

{

Console.WriteLine("Hello World!");

}

}

## C# Example: Using public modifier

We can also specify public modifier before class and Main() method. Now, it can be accessed from outside the class also.

using System;

**public** **class** Program

{

**public** **static** **void** Main(string[] args)

{

Console.WriteLine("Hello World!");

}

}

## C# Example: Using namespace

We can create classes inside the namespace. It is used to group related classes. It is used to categorize classes so that it can be easy to maintain.

using System;

namespace ConsoleApplication1

{

**public** **class** Program

{

**public** **static** **void** Main(string[] args)

{

Console.WriteLine("Hello World!");

}  } }

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**03. C# Variable**

A variable is a name of memory location. It is used to store data. Its value can be changed and it can be reused many times.

It is a way to represent memory location through symbol so that it can be easily identified.

The basic variable type available in C# can be categorized as:

**Variable Type Example**

Decimal types -> decimal

Boolean types -> True or false value, as assigned

Integral types -> int, char, byte, short, long

Floating point types -> float and double

Nullable types -> Nullable data types

Let's see the syntax to declare a variable:

type variable\_list;

The example of declaring variable is given below:

**int** i, j;

**double** d;

**float** f;

**char** ch;

Here, i, j, d, f, ch are variables and int, double, float, char are data types.

We can also provide values while declaring the variables as given below:

**int** i=2,j=4;  //declaring 2 variable of integer type

**float** f=40.2;

**char** ch='B';

### Rules for defining variables

A variable can have alphabets, digits and underscore.

A variable name can start with alphabet and underscore only. It can't start with digit.

No white space is allowed within variable name.

A variable name must not be any reserved word or keyword e.g. char, float etc.

Valid variable names:

**int** x;

**int** \_x;

**int** k20;

Invalid variable names:

**int** 4;

**int** x y;

**int** **double**;