TutorialKart

# C# HelloWorld Example – Basic Program to Understand C# Program Structure

## C# Basic Example – HelloWorld

Basic Program to Understand C# Program Structure.

**Program.cs**

```
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

## What do we have in this C# HelloWorld program?

**1. Imports**

We can import other namespaces and use them in our program.

```
using System
```

In here, we are importing `System` namespace with `using` keyword.

**2. Namespace**

Namespace is a collection of collection of classes and sub namespaces. These sub namespaces are also namespaces in their own scope, but may have to be referenced using the parent namespace.

```
namespace HelloWorld
 {
```

```
}
```

namespace is the keyword used to define a namespacec. `HelloWorld` is the name of the namespace by which we can refer to. And this namespace has a class inside it.

**3. class Definition**

A class is a piece of code that defines how a class of objects should behave, and what properties they can have.

```
class Program
 {

 }
```

`class` is the keyword that defines a class, followed by the class name `Program`.

**4. Method Definition**

Method is a logical representation of a behavior. Methods can be used to create functionalities for a class. A method can have a scope, the type of data it returns, a name to be referenced from other code, list of arguments that it can take.

A method takes in argument(s), not mandatory, and transforms them to return an output or change the state of program or file system or anything that it can get access to.

```
static void Main(string[] args)
 {

 }
```

In this example, the scope of this function is declared as `static`. Return type `void`, meaning it returns nothing. The name that one has to use to access this method is `Main`. And it receives a String array as an argument.

**5. Function Body**

Function body is a set of statements. These could contain print statements, conditional statements, looping statements, or statements that could call other functions, etc.

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

In this example, we are saying ‘Hello World!’ by printing it in the Console.

## C# Tutorial



## C# Console Operations



## C# Object Oriented Programming Concepts



## C# String Operations

- C# String Length
- C# Substring

## C# File Operations

- C# Read Text File
- C# Write to File
- C# Delete File
- C# Copy File

## C# Exception Handling

- C# try-catch
- C# finally
- C# throw
- C# Custom Exception
- C# SystemException
- C# DivideByZeroException
- C# NullReferenceException
- C# InvalidCastException
- C# IOException
- C# FieldAccessException

## C# Collections

- C# List
- C# SortedList
- C# HashSet
- C# SortedSet
- C# Stack
- C# Queue
- C# LinkedList
- C# Dicionary
- C# SortedDictionary

## C# Errors [Solved]

✦ C# Error: Class does not contain a constructor that takes arguments