

# Book



# Introduction

- C# vs .Net
- CLR – Common Language Runtime
- Architecture of .Net Applications
- First C# Application – Hello World!

# C# vs .Net

C# is a programming language

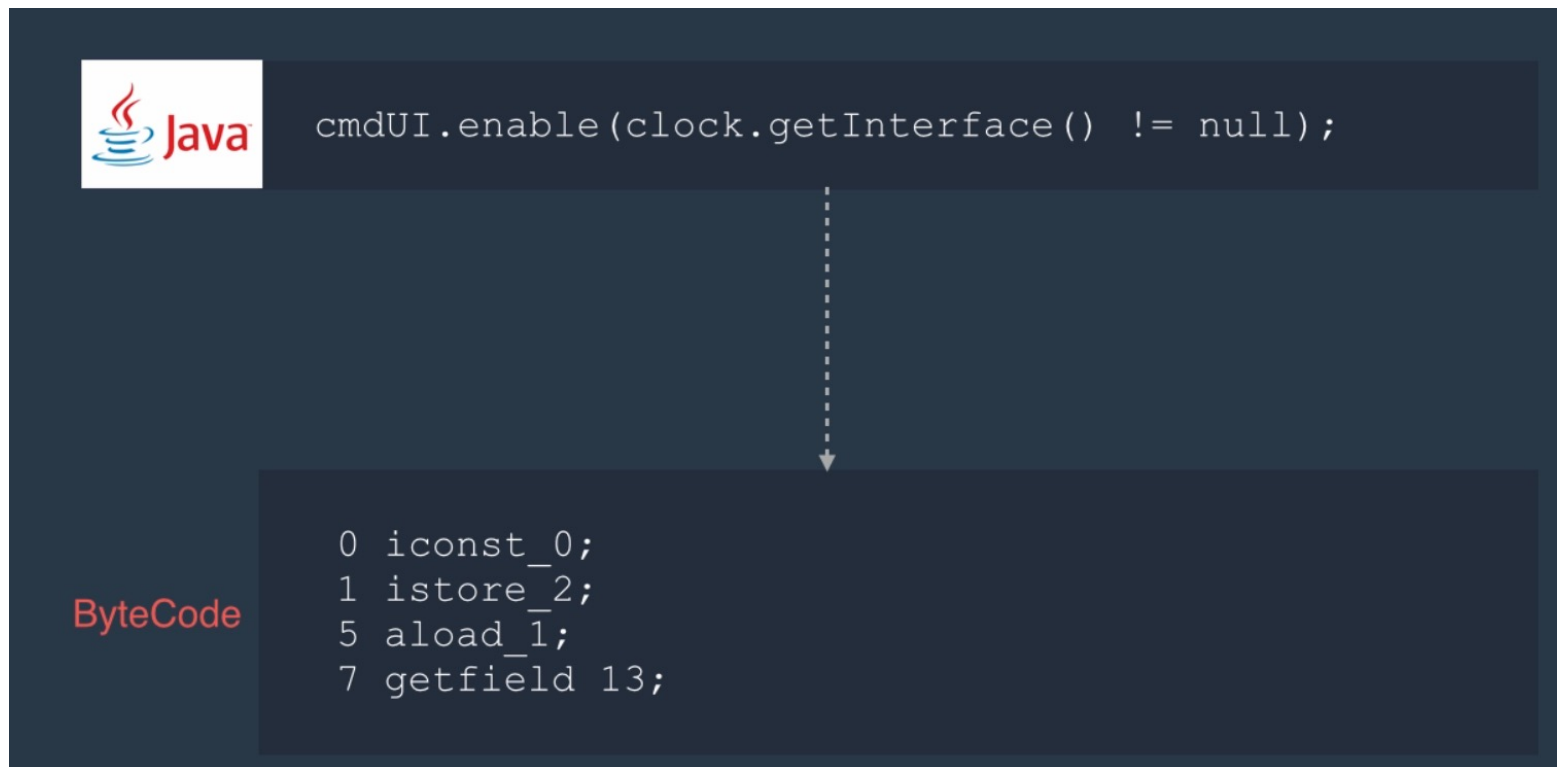
.Net is a framework or an ecosystem

# C# is a programming language

- History of C#
- Java and C#
- Platform independent
- Bytecode and Intermediate Language Code
- Just Intime Compiler (JIT)

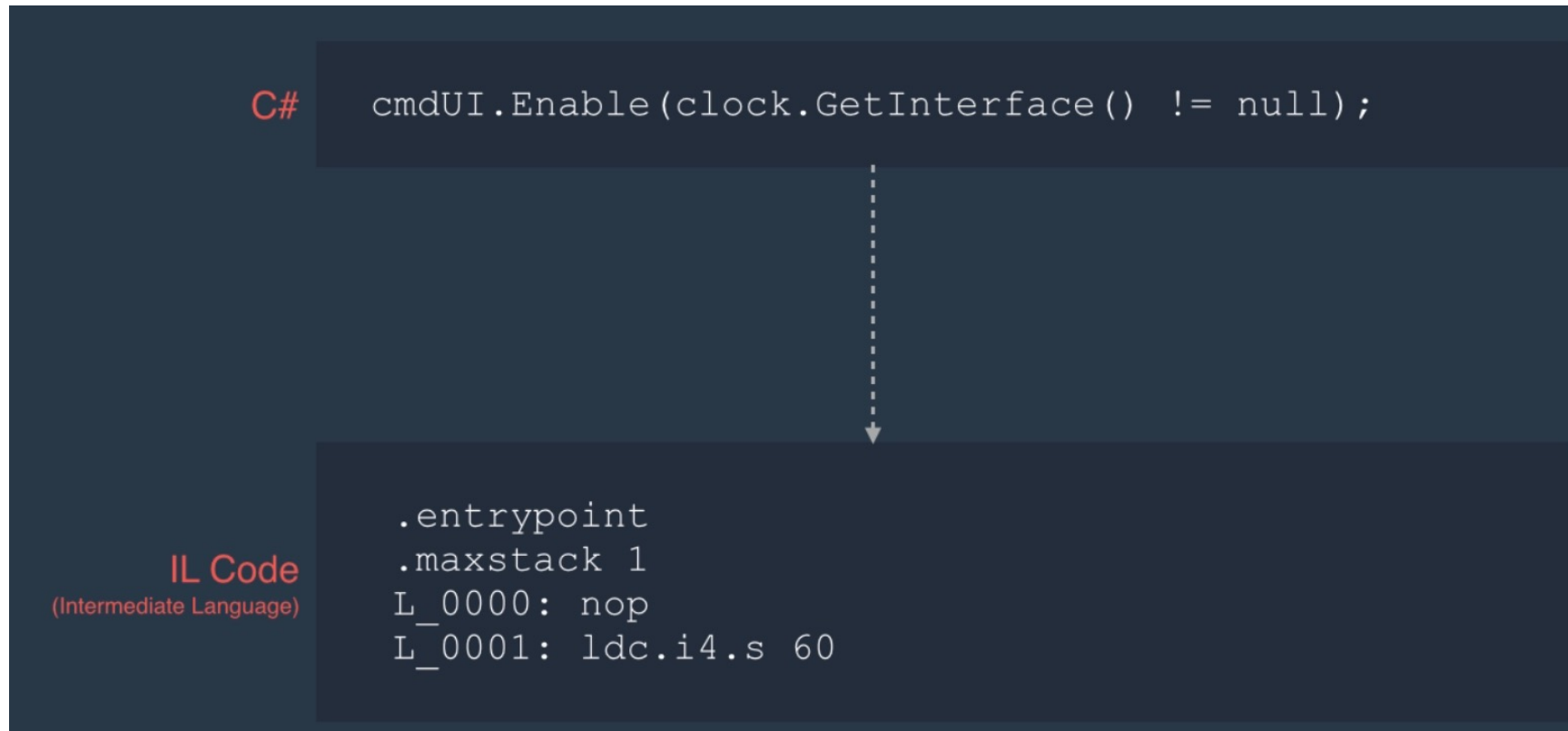
# What is CLR – Common Language Runtime?

We have different OSs, and computer architectures; Java have the following solution to become platform independent. In java we have bytecode and bytecode run on native machine using JVM



# What is CLR – Common Language Runtime?

In C# we have IL code and CLR with the same objective



# What is CLR – Common Language Runtime?

CLI - An application to translate the IL code to machine code to make C# as an platform independent. This process is known as Just in time Compilation (JIT)

# What is .Net?

## **.NET = The Ecosystem**

Languages - C#, F#, Visual Basic, etc.

Runtimes – Common Language Runtime CLR or Core CLR

Libraries – Basic Class Libraries

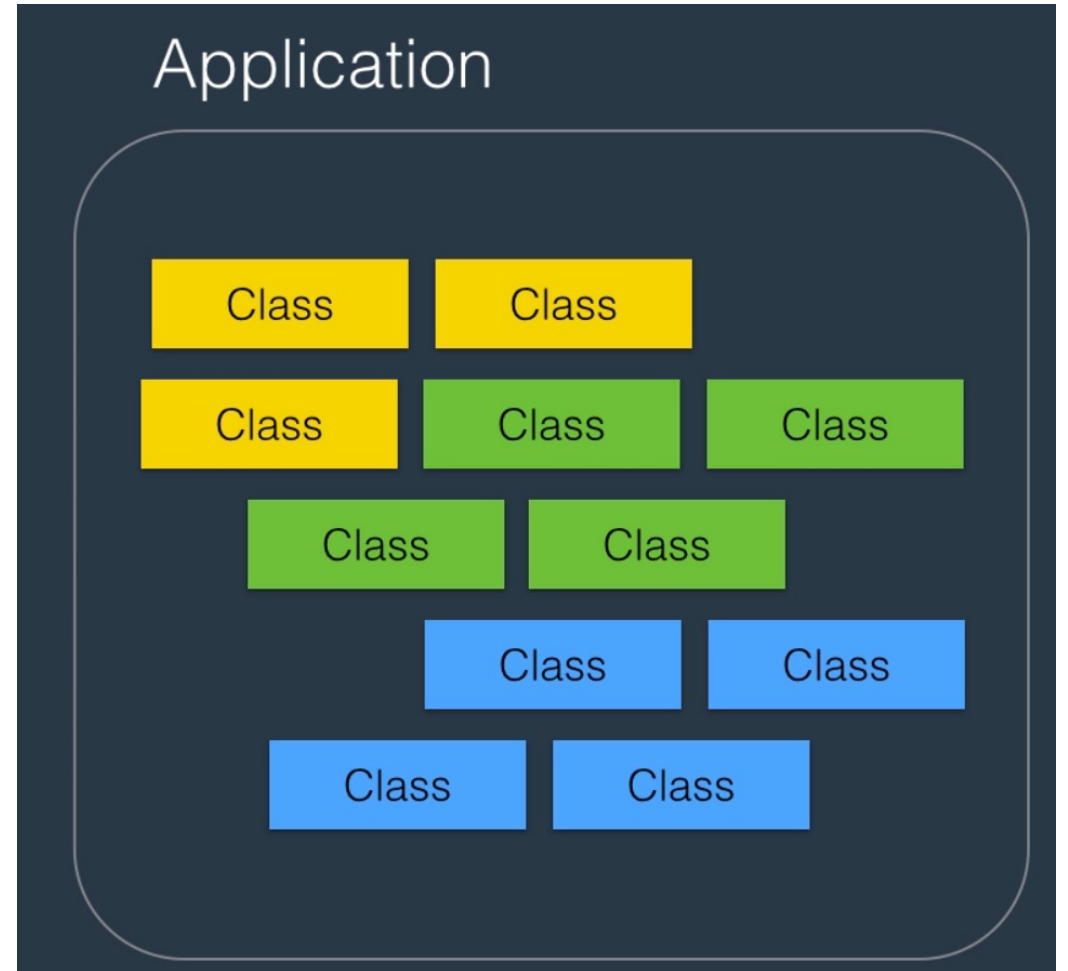


# Others

Nuget – Package manager  
dotnet cli -> command line  
dotnet new  
dotnet run  
dotnet publish

# Architecture of .Net Applications

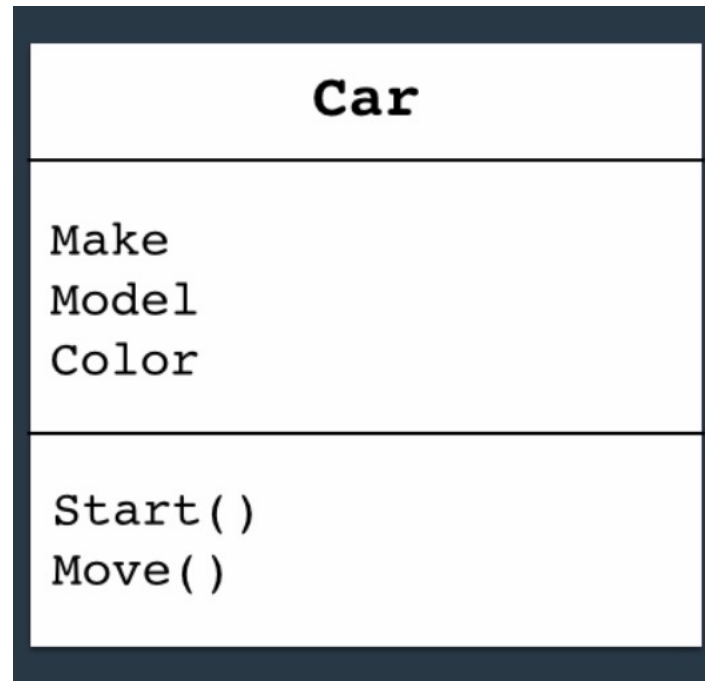
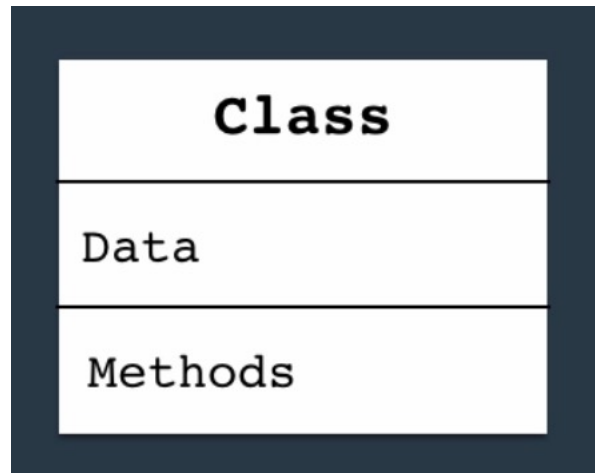
Building Blocks of program are classes which collaborate with each other at run to provide the functionality.



# Class

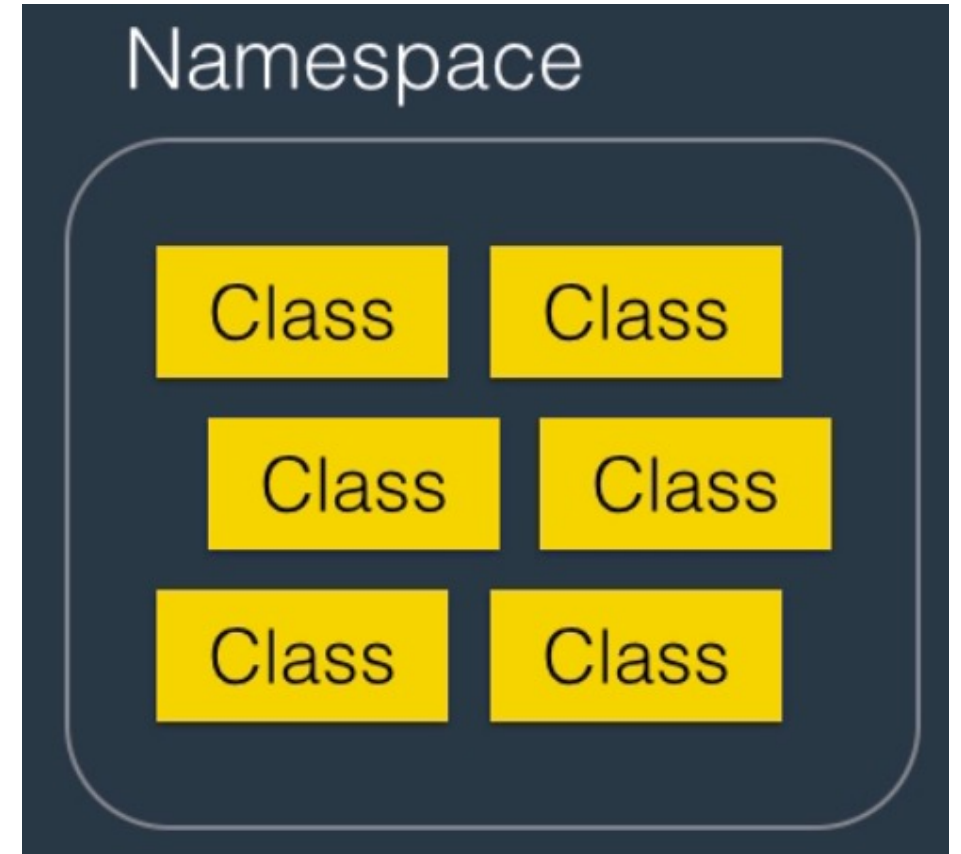
Class is a container which have

1. Data or Attribute (Represents the state of the application)
2. Functions or Methods (its a behavior or do some work on data)



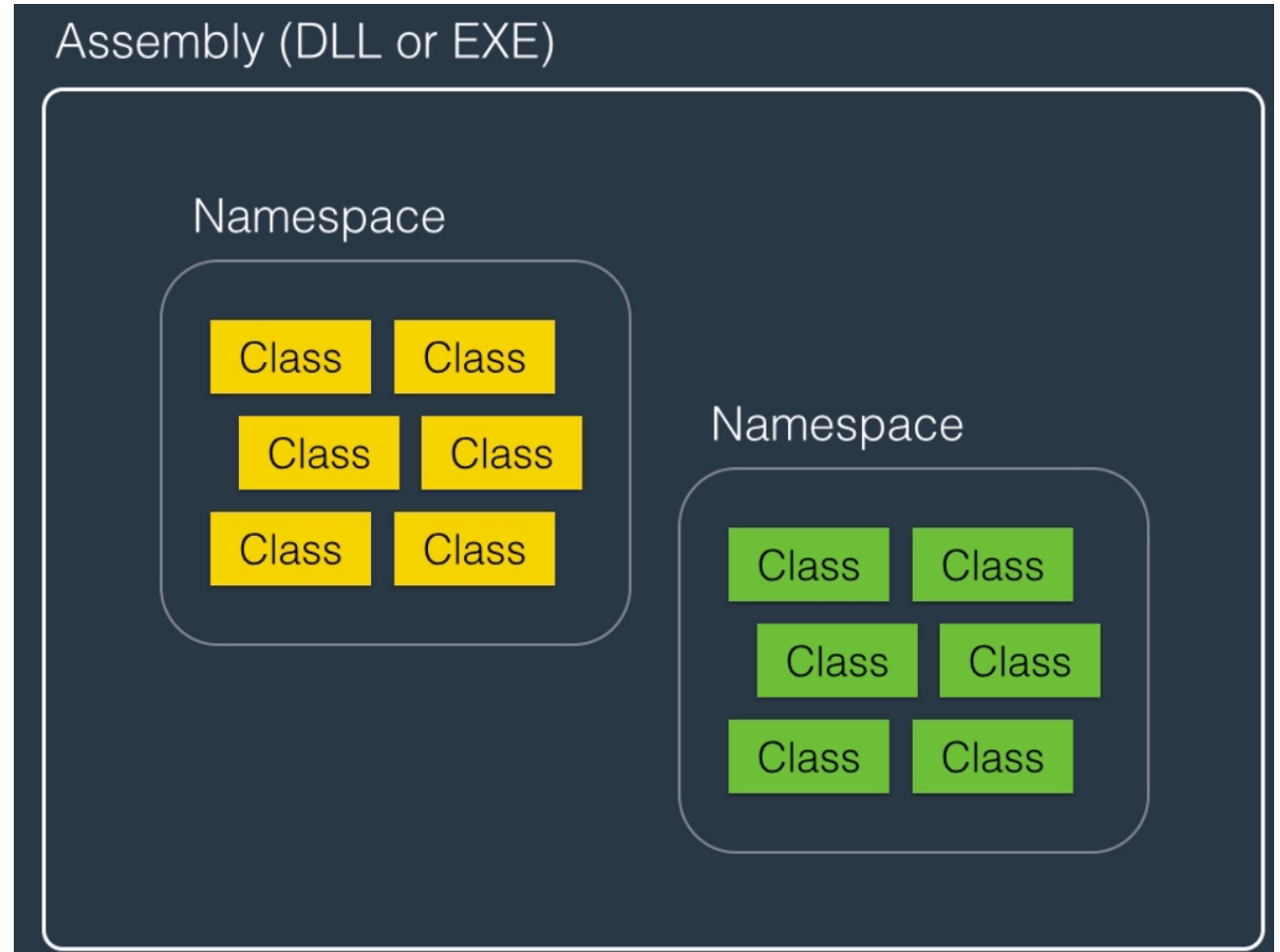
# Namespaces

To organize similar classes - we group them into namespaces. Thus, namespace is a container for similar classes. For example, in dot net we have namespaces with tens of related classes to work with databases, graphics & images, security, etc.



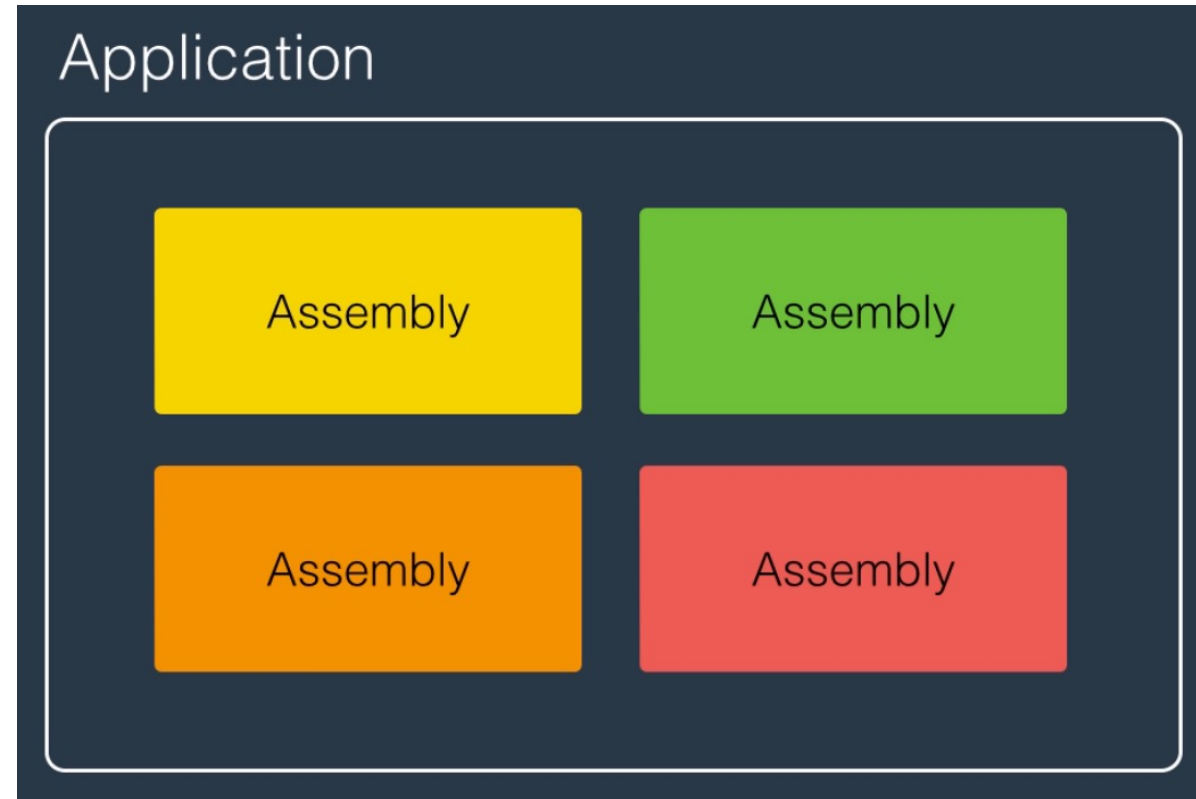
# Assembly

In real world application, as these namespaces grows - we have a different way to partitioning the application known as assembly. Simply an assembly is a container of related namespaces.



# Assembly

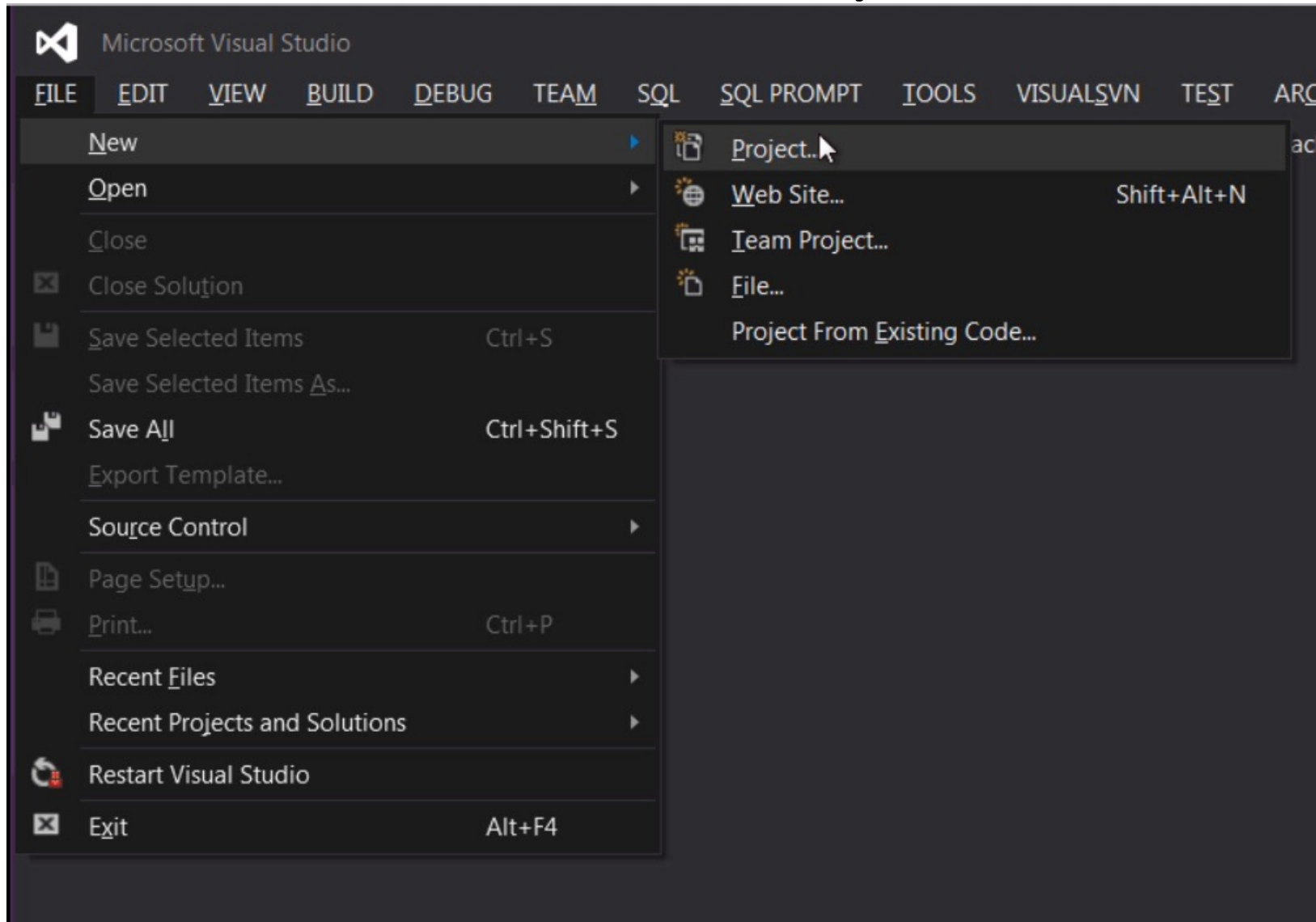
Physically, it is a file on the disk which can either be an executable or DLL file - Dynamic Linked Library. When you compile an application, the compiler builds one or more assemblies depending on how you partition the code. An EXE file represents a program that can be executed. A DLL is a file that includes code that can be re-used across different programs.



# Visual Studio Code

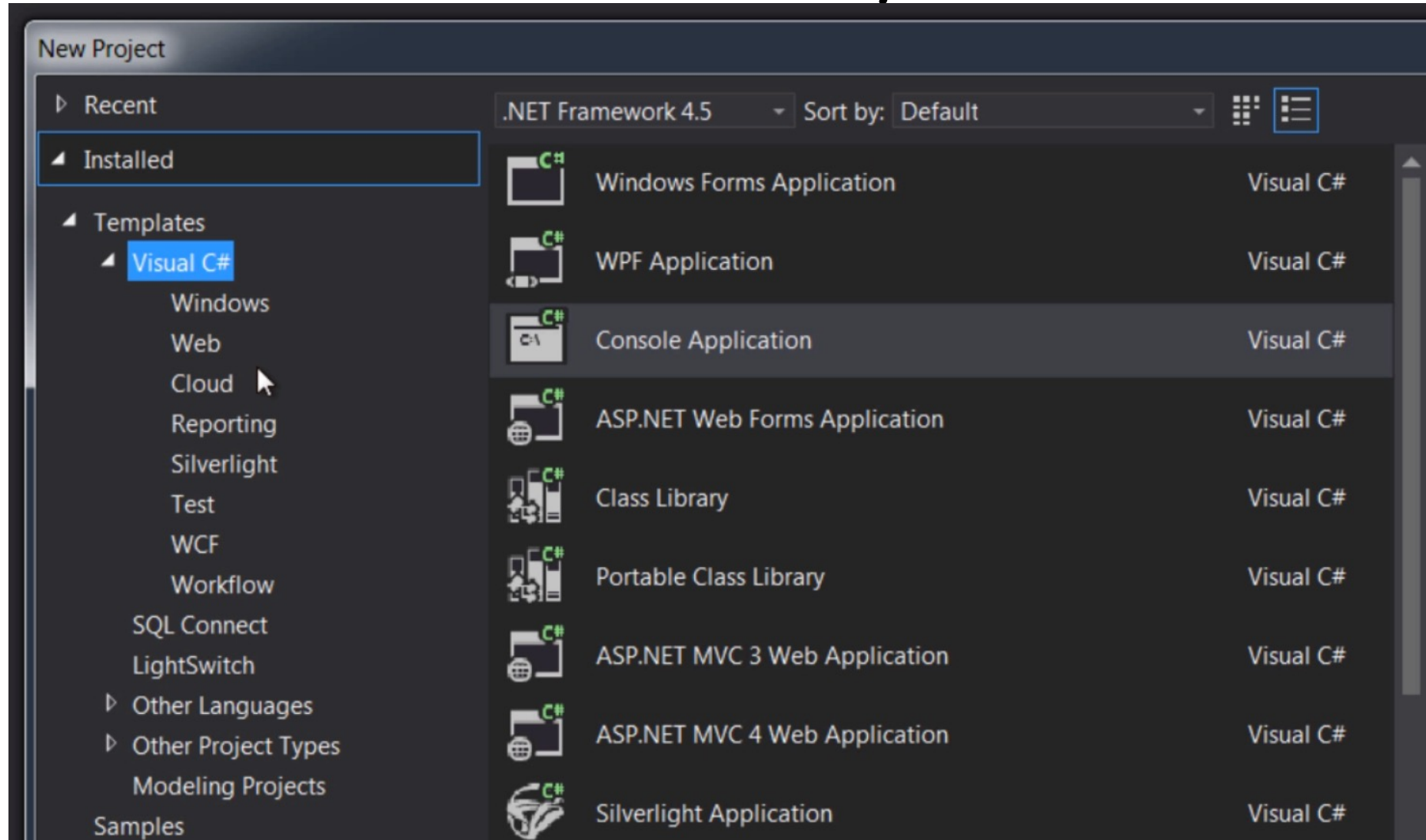
Follow Book

# Visual Studio – Community Edition





# Visual Studio – Community Edition



# HelloWord – Types of files

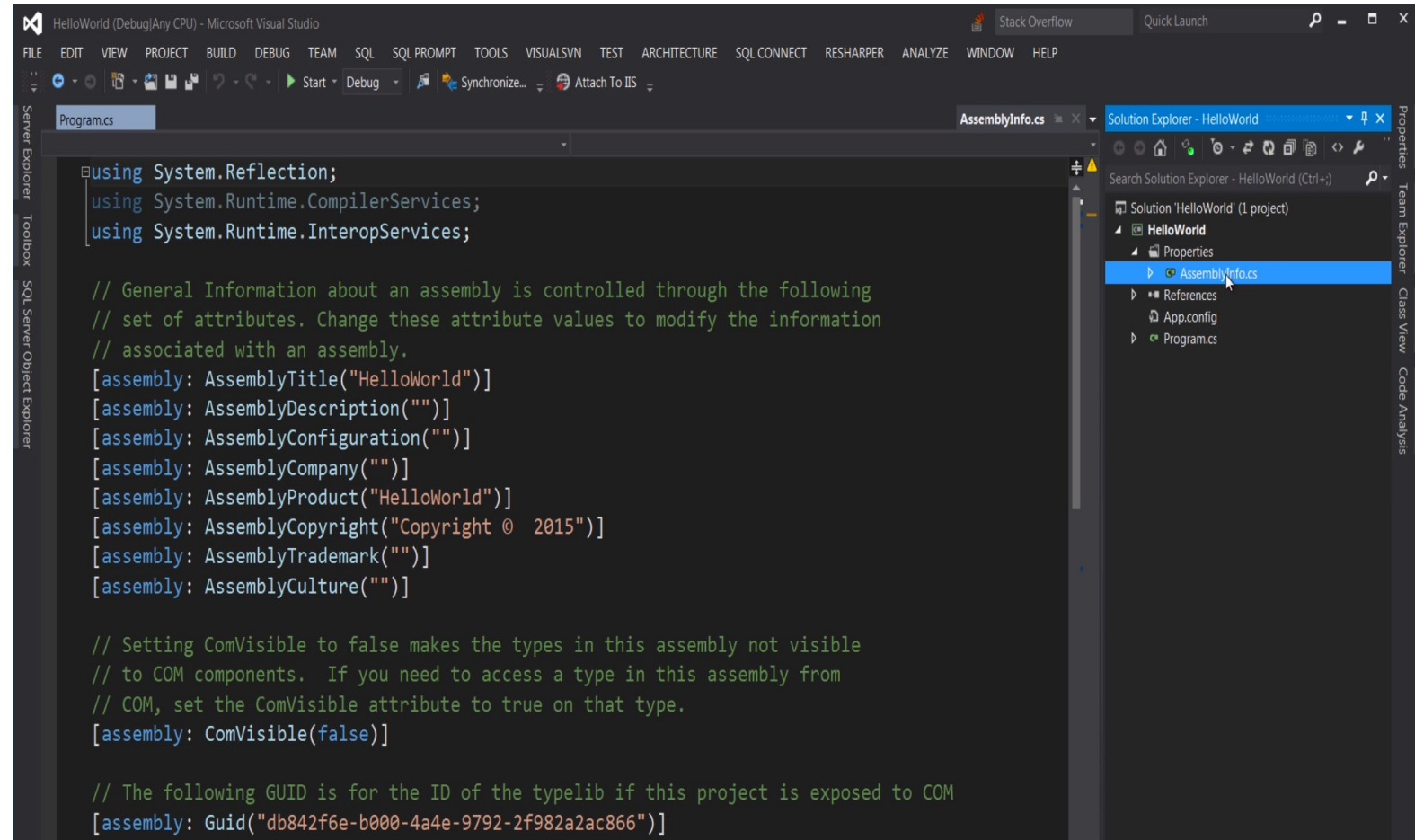
Solution files – Can have multiple projects

HelloWorld - Contains multiple items

1. Properties -> Assemblyinfo.cs
2. Reference
3. App.config
4. Program.cs

# 1. Assemblyinfo.cs

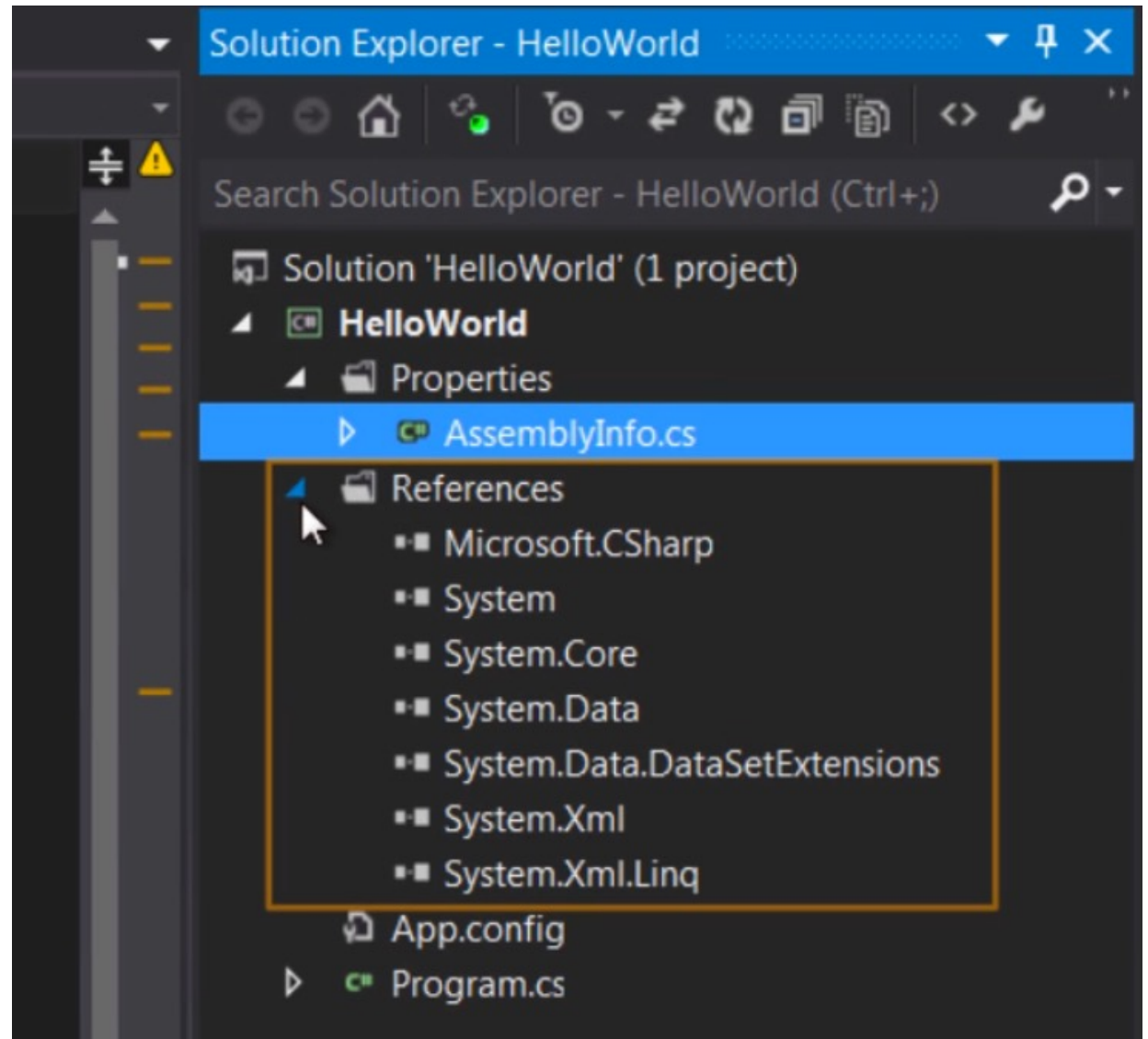
It is the identification for the Assembly that will be produced as result of compiling the project. As result, we will get an executable which is an assembly. It contains identifications and also known as assembly manifest required when we need to distribute the application.



## 2. References

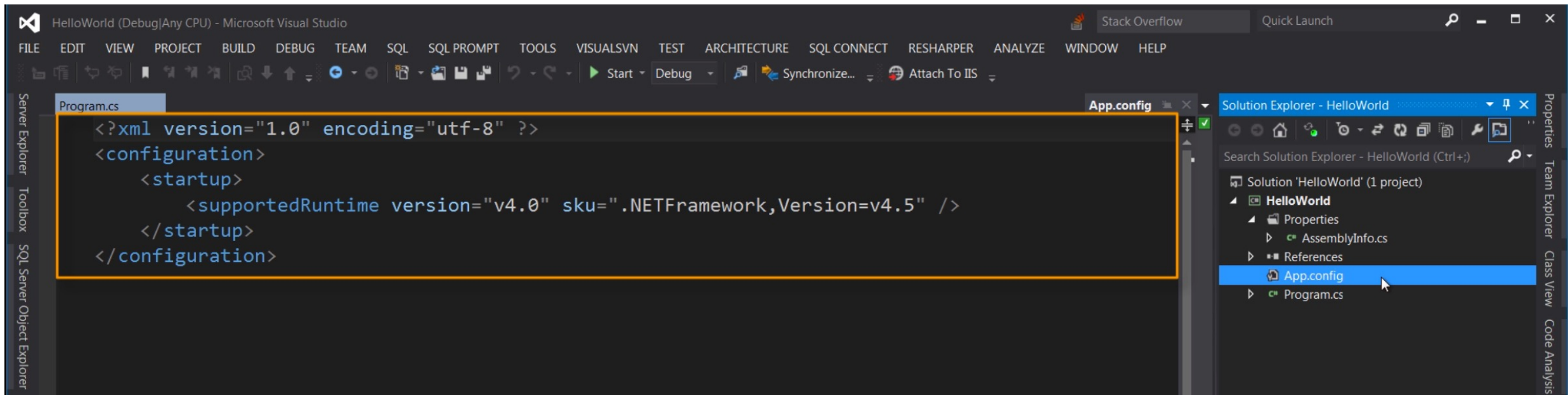
It contains list of the assemblies this project is referencing to do his job.

When you create a project with visual studio by default include reference to a bunch of assemblies that are going to be used by the project like System and others.



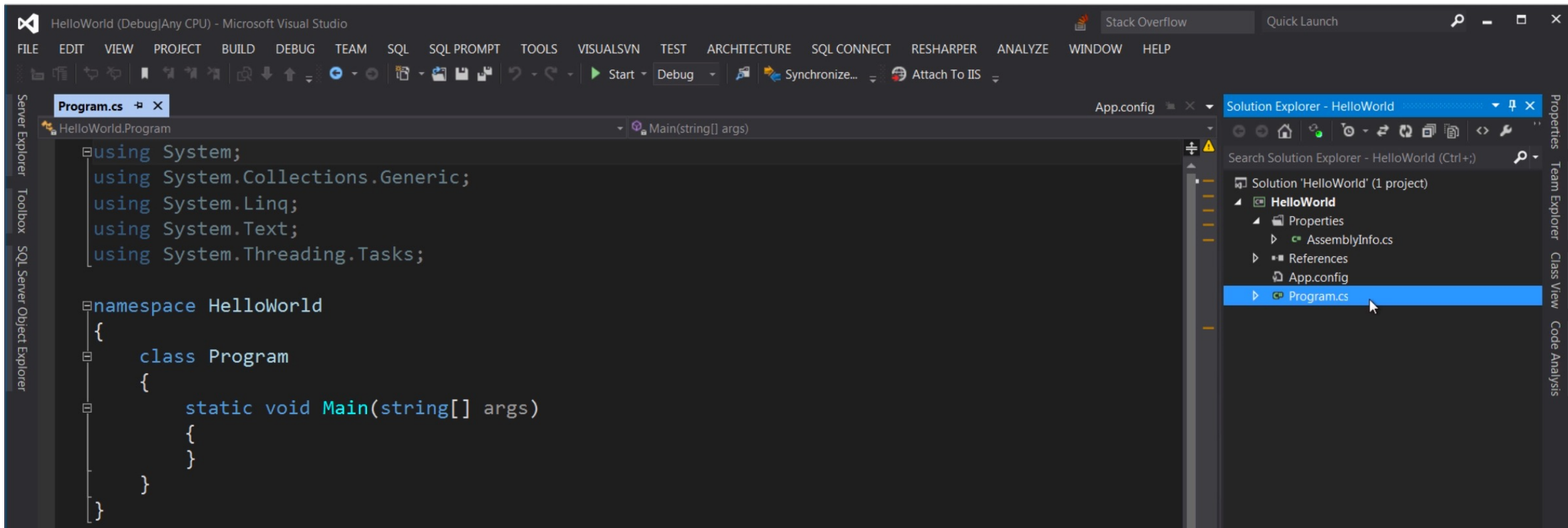
### 3. App.config

An xml file which stores the configuration for this application. Sometimes you may want to store connection strings through the database or you may want to have some about your application.



# 4. Program.cs

This is the file where we will be going to write our code.

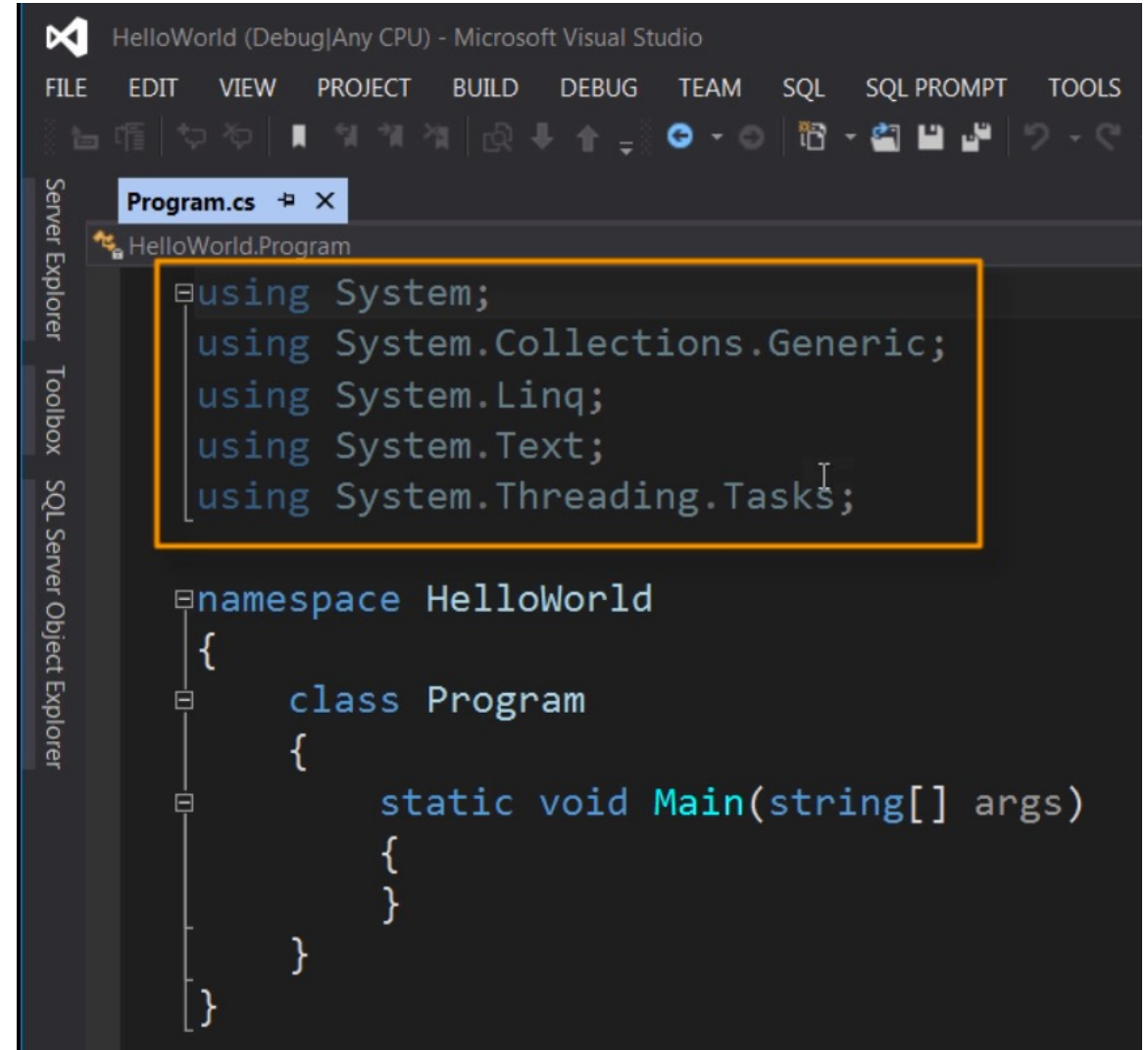




# HelloWorld

**using statements** - by default visual studio creates a namespace called Hello world when you write code in this namespace. we have access to any classes defined in this namespace.

if we want to use a class that is defined in a different namespace we need to import it in our code.

A screenshot of the Microsoft Visual Studio IDE. The title bar reads 'HelloWorld (Debug|Any CPU) - Microsoft Visual Studio'. The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, SQL, SQL PROMPT, and TOOLS. The toolbar shows various icons for file operations and development. On the left, the 'Server Explorer' and 'Toolbox' are visible. The main editor window shows a file named 'Program.cs' with the following C# code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

The 'using' statements are highlighted with an orange rectangle. The 'namespace HelloWorld' is also highlighted with an orange rectangle.

# using system

**system** is a namespace in a Dot Net Framework and contains all basic utility classes and primitive types



# class – Program

By default, we have a class called program so every console application you create with Visual Studio

has a class called program Inside program by default.

Method – Main which is an entry point to the application

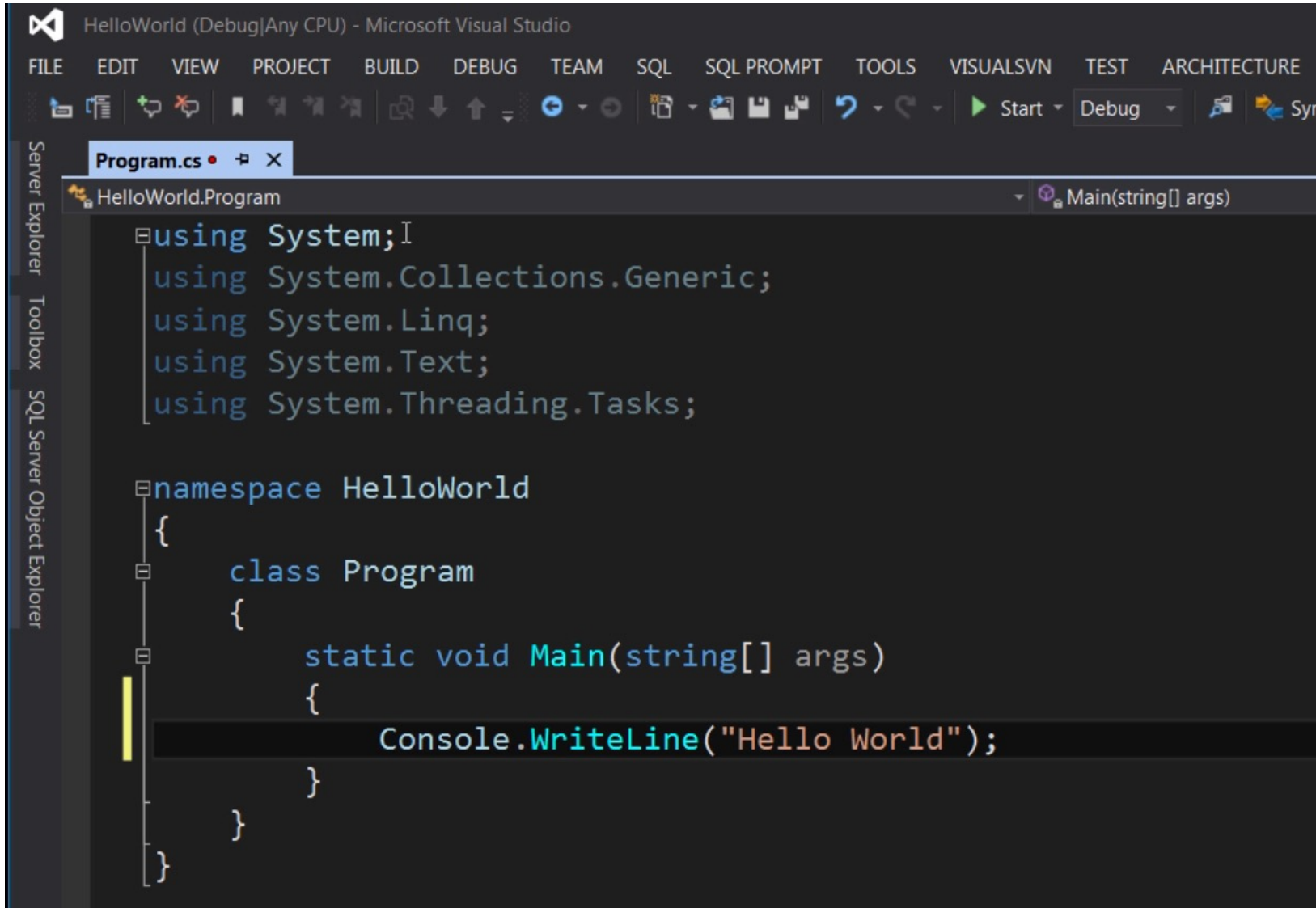
```
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

# class – Program

1. Boilerplate Code
2. Method Names and Conventions
3. Block names and curly braces
4. Indentation
5. Whitespaces ignore
6. Case Sensitive
7. Methods have input and outputs
  1. void – method does not return anything
  2. Arguments – input to the methods (like string array args)

```
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

# Body of the program



```
Microsoft Visual Studio
HelloWorld (Debug|Any CPU)

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL SQL PROMPT TOOLS VISUALSVN TEST ARCHITECTURE

Program.cs
HelloWorld.Program
Main(string[] args)

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

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