Code Academy
أكاديمية البرمجة

# Introduction to .Net

Eng : Yousif mohamed

Code Academy
أكاديمية البرمجة

# Table of Content

*1- Setup Environment*

*2- Exploring Visual Studio*

*3-Solution Concept and Project*

*4- NameSpace and Using KeyWord*

*5- C# under the hood*

*6- Data Types*

# Setup Environment Microsoft Visual Studio IDE

Let’s Do it Together Family

https://learn.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2022

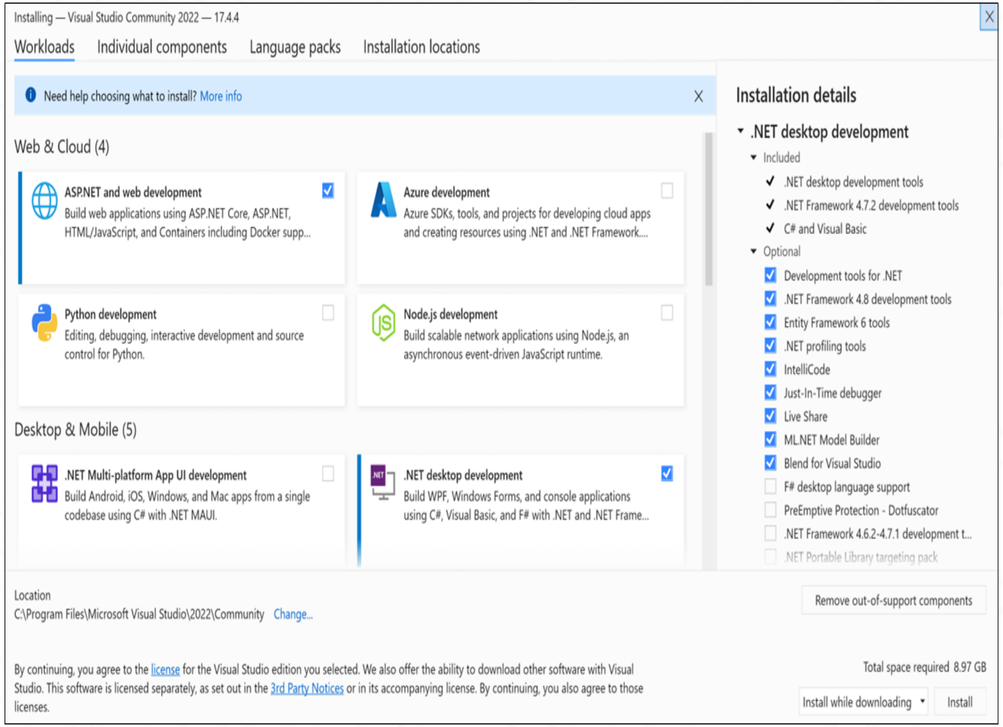
Code Academy
أكاديمية البرمجة

# Installing Packages .Net Packages

We will need to Install Asp.Net and Web Development that will include for us C# and all Packages we will need

**Note** : it might looks different so just Choose C# , .Net Mvc , .Net web Api

If your already installed with less packages you can always install whatever you want as a Separated Package



Installing — Visual Studio Community 2022 — 17.4.4

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

Web & Cloud (4)

- [✓] **ASP.NET and web development** — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- [ ] **Azure development** — Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....
- [ ] **Python development** — Editing, debugging, interactive development and source control for Python.
- [ ] **Node.js development** — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- [ ] **.NET Multi-platform App UI development** — Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.
- [✓] **.NET desktop development** — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

Installation details

- .NET desktop development
  - Included
    - ✓ .NET desktop development tools
    - ✓ .NET Framework 4.7.2 development tools
    - ✓ C# and Visual Basic
  - Optional
    - [✓] Development tools for .NET
    - [✓] .NET Framework 4.8 development tools
    - [✓] Entity Framework 6 tools
    - [✓] .NET profiling tools
    - [✓] IntelliCode
    - [✓] Just-In-Time debugger
    - [✓] Live Share
    - [✓] ML.NET Model Builder
    - [✓] Blend for Visual Studio
    - [ ] F# desktop language support
    - [ ] PreEmptive Protection - Dotfuscator
    - [ ] .NET Framework 4.6.2-4.7.1 development t...
    - [ ] .NET Portable Library targeting pack

Location
C:\Program Files\Microsoft Visual Studio\2022\Community Change...

Remove out-of-support components

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 8.97 GB

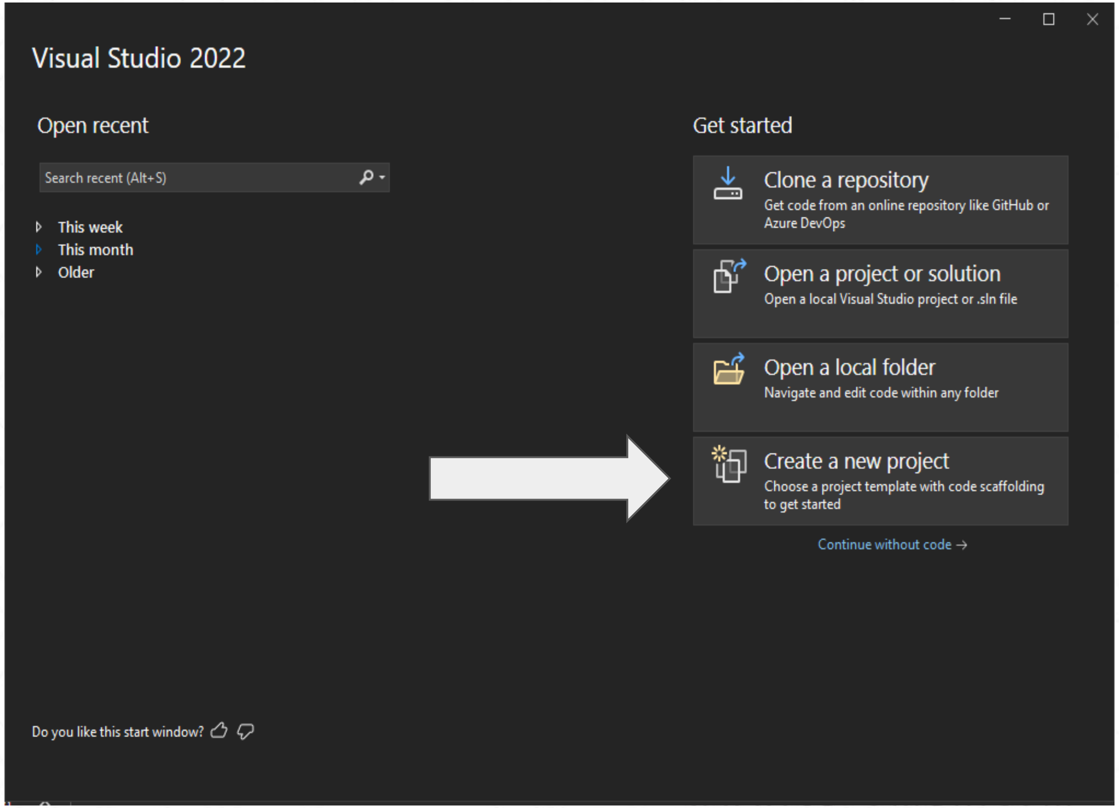
Install while downloading | Install

# Visual Studio Start Window

That's the Start window that will appear for you when you Start the Visual Studio

## Creating a New Project :

To Create Your New Project just Follow The Arrow


Visual Studio 2022

Open recent

Search recent (Alt+S)

- This week
- This month
- Older

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

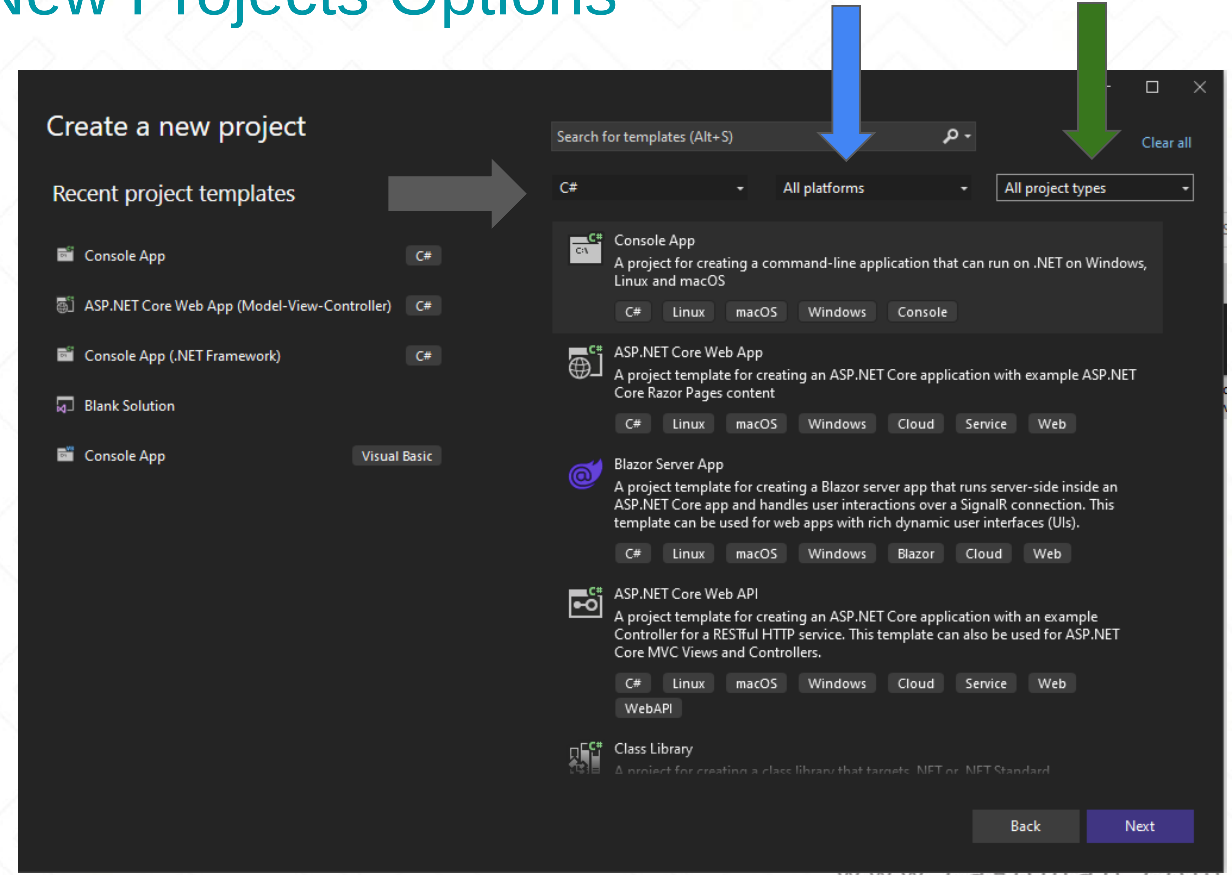
Do you like this start window?

# Let's Discuss the New Projects Options

Arrow 1 :
in Our First Course we will Discuss C# Fundamentals

Arrow 2 :
In this Section you Can Choose which development environment PlatForm you are Targeting

Arrow 3 :
As we are in the basic so we will work on Console Application so select Console



Create a new project

Recent project templates

- Console App — C#
- ASP.NET Core Web App (Model-View-Controller) — C#
- Console App (.NET Framework) — C#
- Blank Solution
- Console App — Visual Basic

Search for templates (Alt+S)

Clear all

C# | All platforms | All project types

**Console App**
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# · Linux · macOS · Windows · Console

**ASP.NET Core Web App**
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content
C# · Linux · macOS · Windows · Cloud · Service · Web

**Blazor Server App**
A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).
C# · Linux · macOS · Windows · Blazor · Cloud · Web

**ASP.NET Core Web API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# · Linux · macOS · Windows · Cloud · Service · Web · WebAPI

**Class Library**
A project for creating a class library that targets .NET or .NET Standard

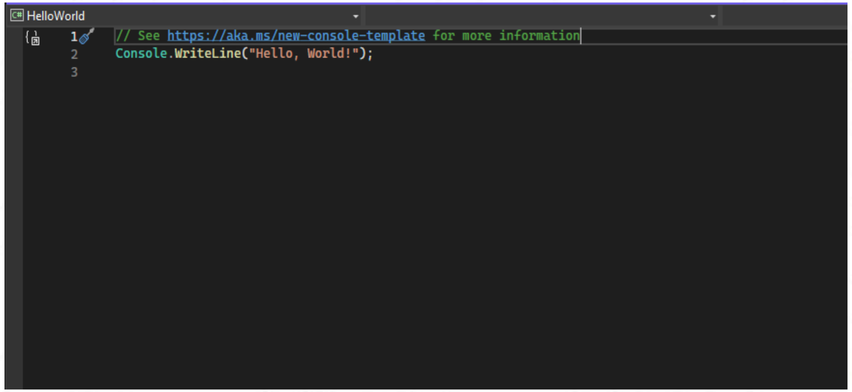
Back | Next

## What is the Console (Base/FrameWork Class Library) :

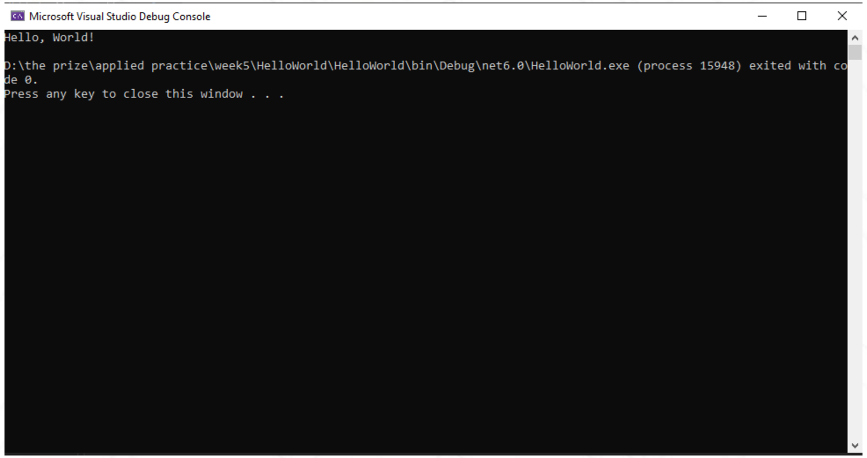
In C#, a console is used as a way for a program to interact with a user through the command-line interface (CLI). The console is a text-based interface where users can enter commands or data and view program output.

In C#, the Console class provides a set of methods and properties that can be used to interact with the console. Some of the most commonly used methods of the Console class include:

- Console.WriteLine(): used to display a line of text on the console.
- Console.ReadLine(): used to read a line of input from the user.
- Console.Write(): used to display text on the console without starting a new line.



```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

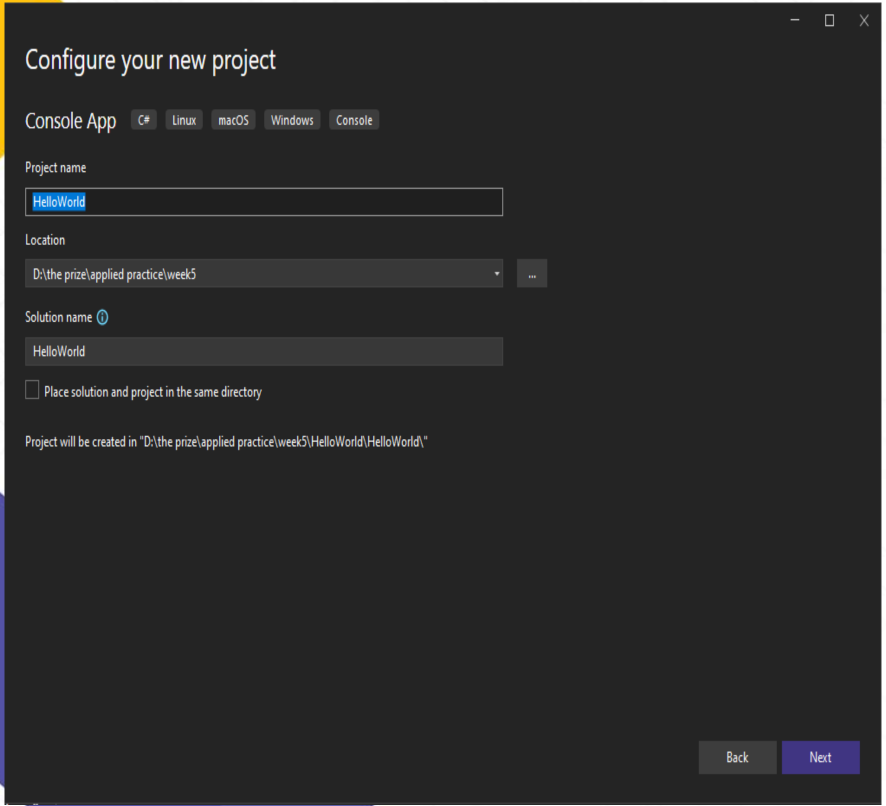


```
Hello, World!

D:\the prize\applied practice\week5\HelloWorld\HelloWorld\bin\Debug\net6.0\HelloWorld.exe (process 15948) exited with co
de 0.
Press any key to close this window . . .
```

Code Academy
أكاديمية البرمجة

## Specifying Project Name and .Net Version



Configure your new project

Console App C# Linux macOS Windows Console

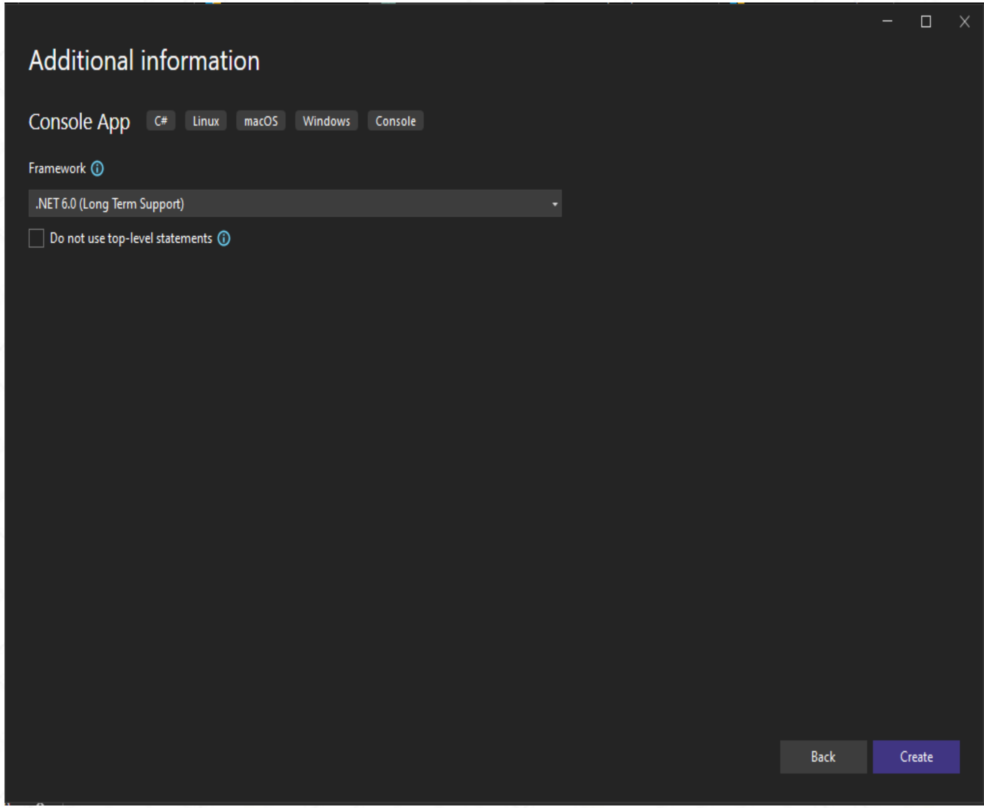
Project name: HelloWorld

Location: D:\the prize\applied practice\week5

Solution name: HelloWorld

☐ Place solution and project in the same directory

Project will be created in "D:\the prize\applied practice\week5\HelloWorld\HelloWorld\"



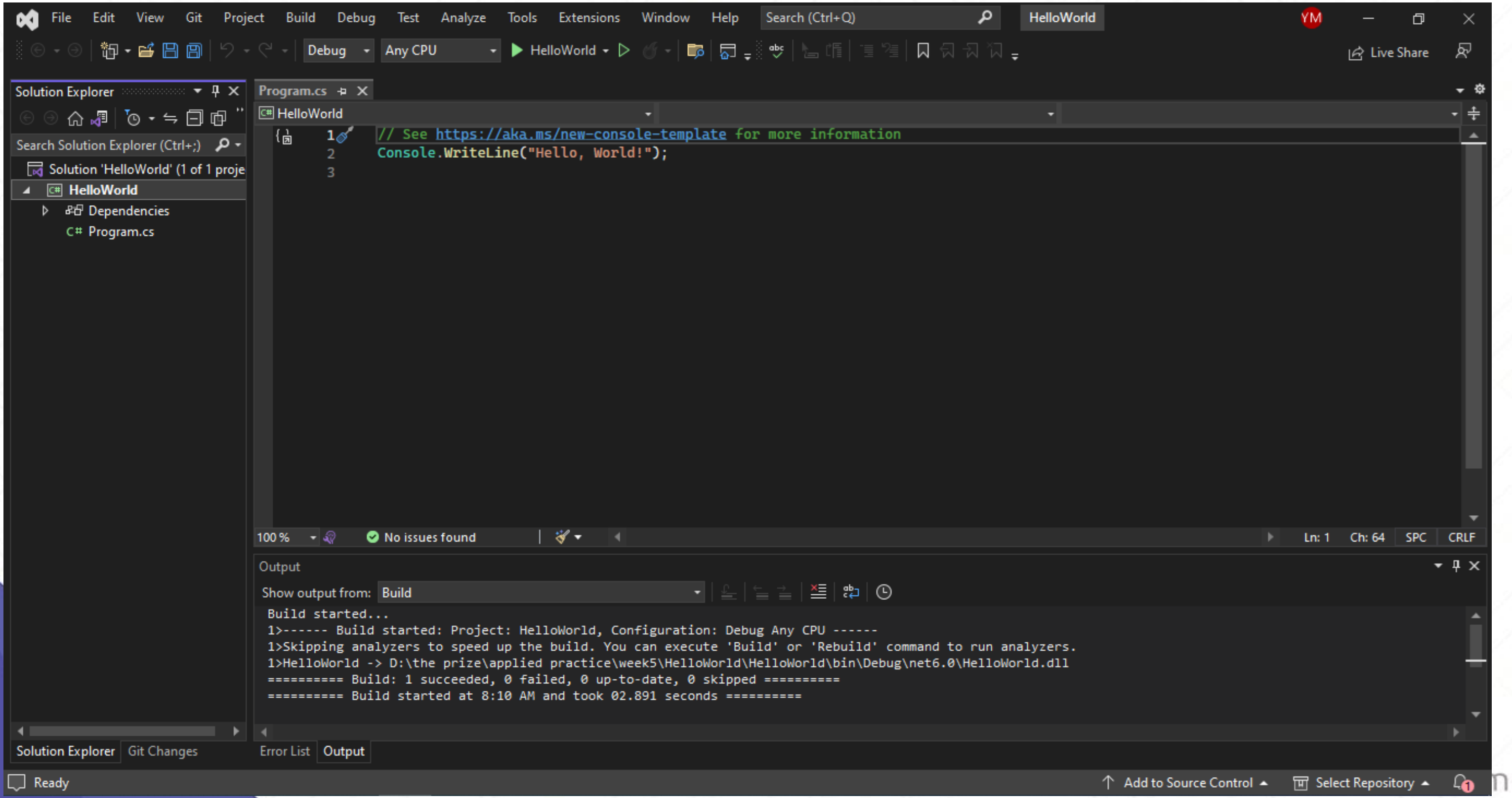
Additional information

Console App C# Linux macOS Windows Console

Framework: .NET 6.0 (Long Term Support)

☐ Do not use top-level statements

# Let’s Discuss Together the Project Structure



```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

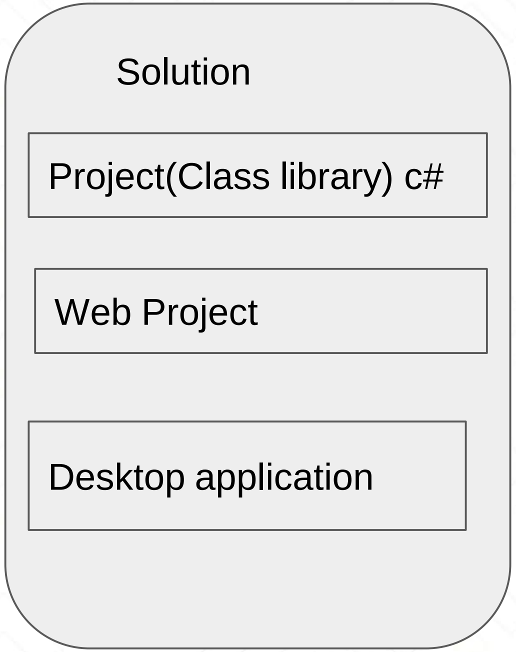
```
Build started...
1>------ Build started: Project: HelloWorld, Configuration: Debug Any CPU ------
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>HelloWorld -> D:\the prize\applied practice\week5\HelloWorld\HelloWorld\bin\Debug\net6.0\HelloWorld.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build started at 8:10 AM and took 02.891 seconds ==========
```

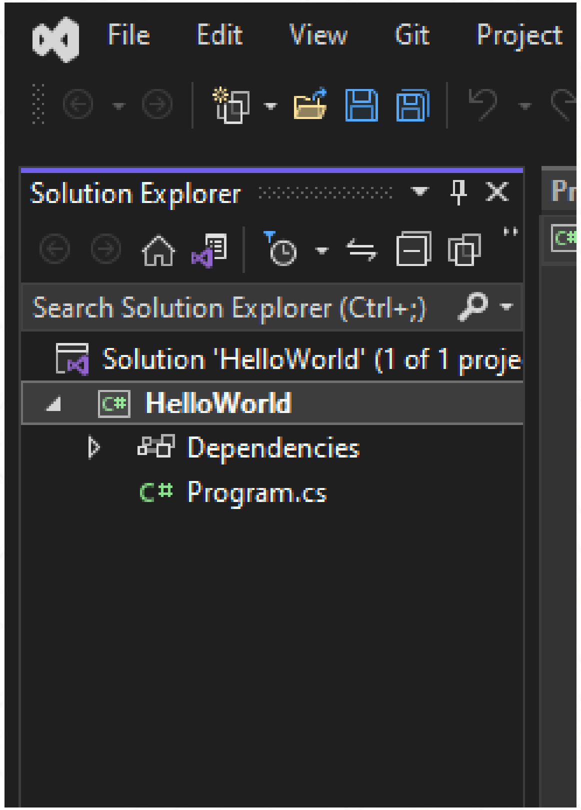
Code Academy
أكاديمية البرمجة

## What is the Solution ?

Solution idea is like a wrapper for big Projects , when you are Developing Big Projects we divide it to small Projects but we Found a Problem that we may repeat some Code so we needed like a big Container or umbrella that would hold all this Projects and if we want to Use Some entities from other Projects We Can Use it without repeating it “DRY” .

In Other Meaning its Like Book Contains Chapters



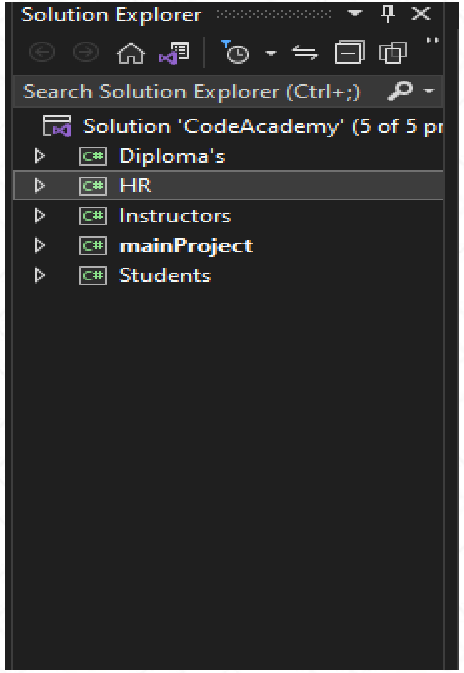


Solution 'HelloWorld' (1 of 1 proje
HelloWorld
Dependencies
Program.cs

Code Academy
أكاديمية البرمجة

## Project Structure With Example

Code Academy Wants to make a fully Functional System , a system that Could manage Scholarship and diploma's that they Can Provide so Diplomas has main Factors Students and Instructors for each one of them they have their Own Functionalities so if they merged everything in One Project the Code will be Hard to Develop and Maintenance so they Follow the Solution Concept that will include Several Projects and they Can Connect with Each Other

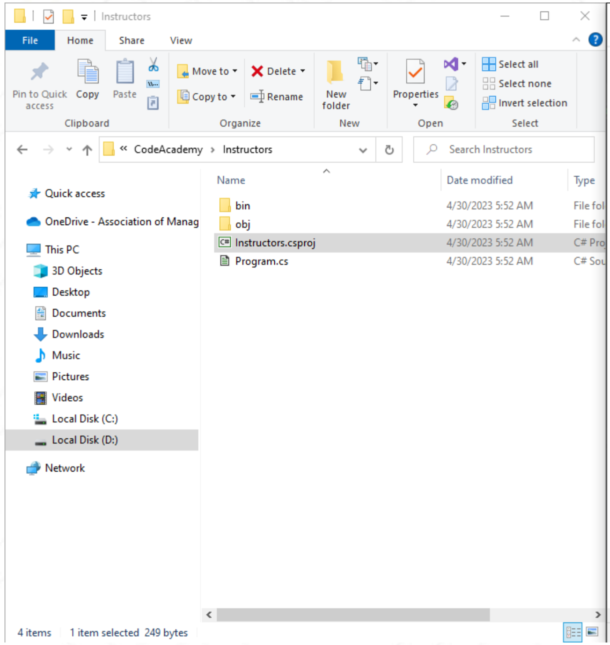


```
Solution 'CodeAcademy' (5 of 5 pr
  Diploma's
  HR
  Instructors
  mainProject
  Students
```

# Projects Components

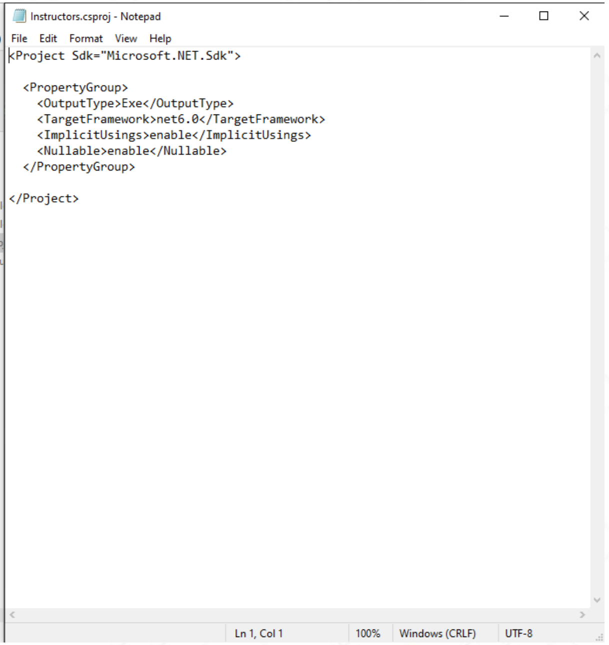
In Each Project Created that’s the Structure that they Follow :

- Bin : Contains the output for the build of the Project , because we are working on Windows after the Build it will contain .exe (excutable)
- Obj : Contain all Project Main Components that we are using in the Project
- .csproj : Contain version Release and the Output version after Build Targeted FrameWork



| Name | Date modified | Type |
|---|---|---|
| bin | 4/30/2023 5:52 AM | File fol |
| obj | 4/30/2023 5:52 AM | File fol |
| Instructors.csproj | 4/30/2023 5:52 AM | C# Pro |
| Program.cs | 4/30/2023 5:52 AM | C# Sou |

4 items  1 item selected 249 bytes



```xml
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

</Project>
```

Ln 1, Col 1  100%  Windows (CRLF)  UTF-8

Code Academy
أكاديمية البرمجة

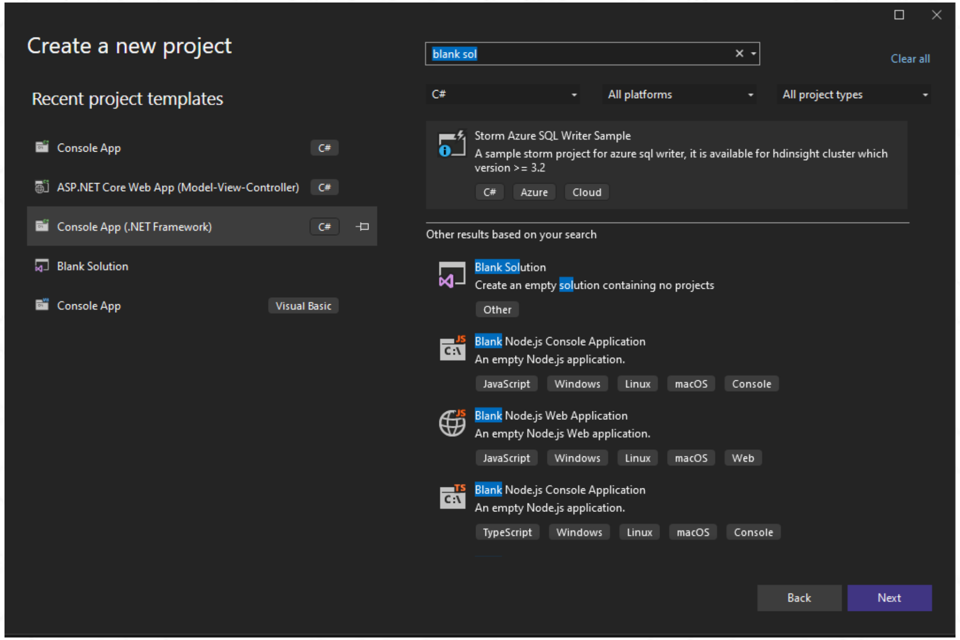
## Let’s Create Blank Solution and add Our Projects

# Blank Solution (For illustration)

The Blank Solution is an empty Book you are intending to Create its own Chapter's , so we will Create a Blank Solution For OmanTel .

After Creating the Solution we will Create Our Own Projects .

So let's Learn how the Pro's Makes it we will name our sol. With the Company Name and each project we will make we will follow the Naming Convention "CompanyName.Projectname" Let's Do it Together Differently



Recent project templates:

- Console App — C#
- ASP.NET Core Web App (Model-View-Controller) — C#
- Console App (.NET Framework) — C#
- Blank Solution
- Console App — Visual Basic

Storm Azure SQL Writer Sample
A sample storm project for azure sql writer, it is available for hdinsight cluster which version >= 3.2
C# Azure Cloud

Other results based on your search

Blank Solution
Create an empty solution containing no projects
Other

Blank Node.js Console Application
An empty Node.js application.
JavaScript Windows Linux macOS Console

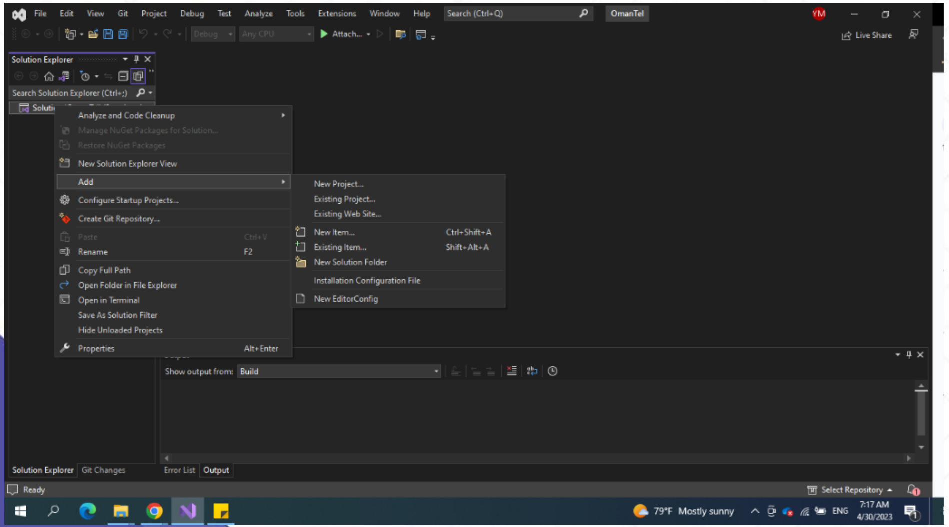
Blank Node.js Web Application
An empty Node.js Web application.
JavaScript Windows Linux macOS Web

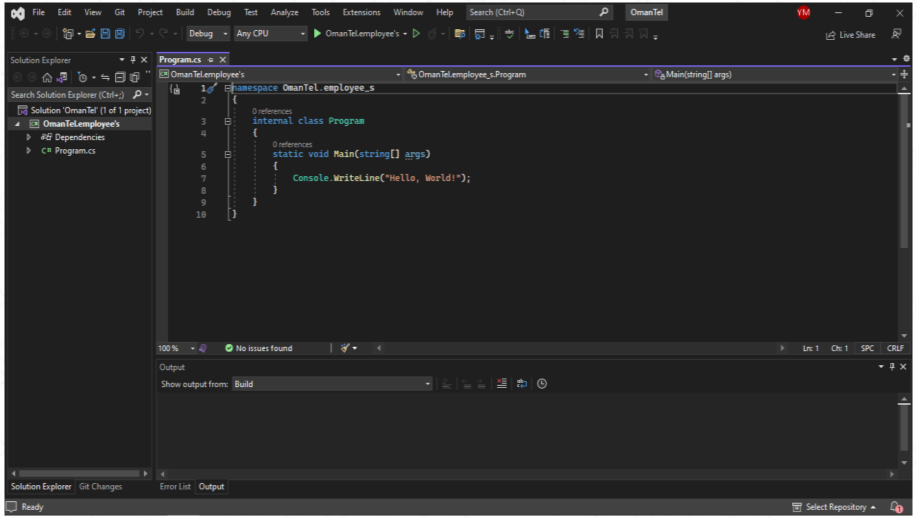
Blank Node.js Console Application
An empty Node.js application.
TypeScript Windows Linux macOS Console

Code Academy
أكاديمية البرمجة

# Adding Project To Solution

Let's Discover Together Why Looks Different





```
namespace OmanTel.employee_s
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

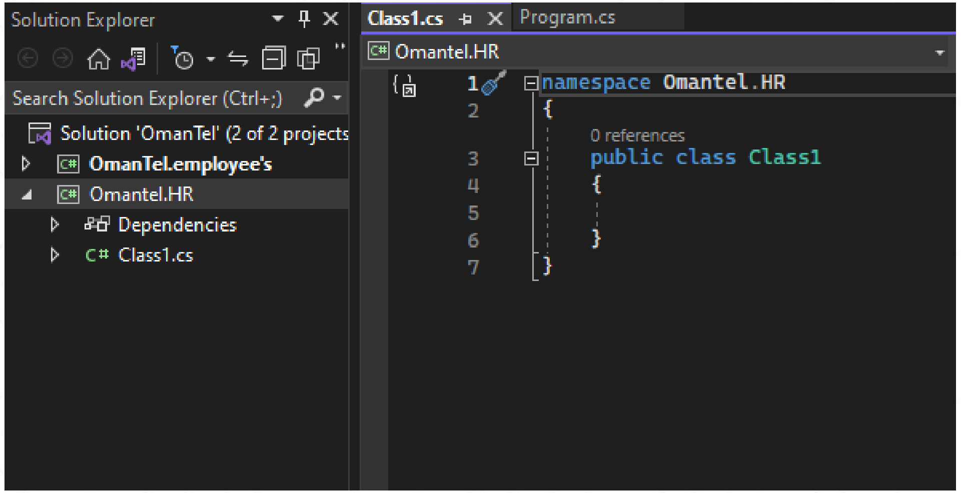
www.ca-oman.com

# Let’s add another another Component To our Project

I Will add a Class Library which will not interact with user directly but we uses it to handle specific Functionality like connecting to DataBase and Don’t Worry we will deal with it in our upcoming Levels . I  i will Create it here for HR

Right Click to solu → add → new Project → ClassLibrary → Setname Conv. → OmanTel.HR

Now how we can Connect Them Together to user HR in Employee’s ??! Project Referencing



```
namespace Omantel.HR
{
    0 references
    public class Class1
    {
    }
}
```

## NameSpace :

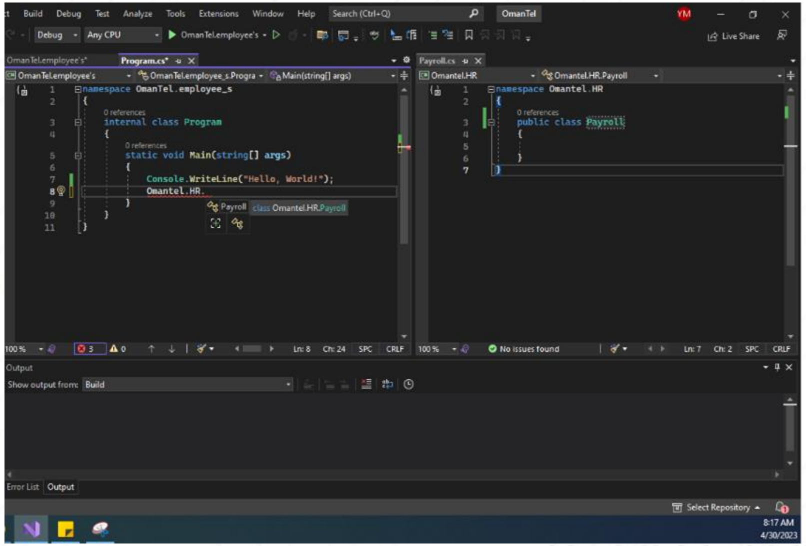
Developers can create their own namespaces to organize their own classes and types, and they can also use the using directive to reference namespaces from other assemblies in their code. This allows them to use classes and types from other namespaces without having to fully qualify the names every time they are used.

In other meaning Name Space is like Folder hierarchy that i am Telling the machine that inside HR folder there is a class that i can use

So now what if i want to use the omantel.Hr.payroll(Class inside hr ) into employee’s Project

I will go to the Project that i want to use in it the other Project and i do this Steps

rightClick (employees)→ add → Project Reference → Select HR (NOW YOU CAN USE IT ) We Call it Using NameSpace



```
namespace OmanTel.employee_s
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
            Omantel.HR.
        }
    }
}
```

Payroll class Omantel.HR.Payroll

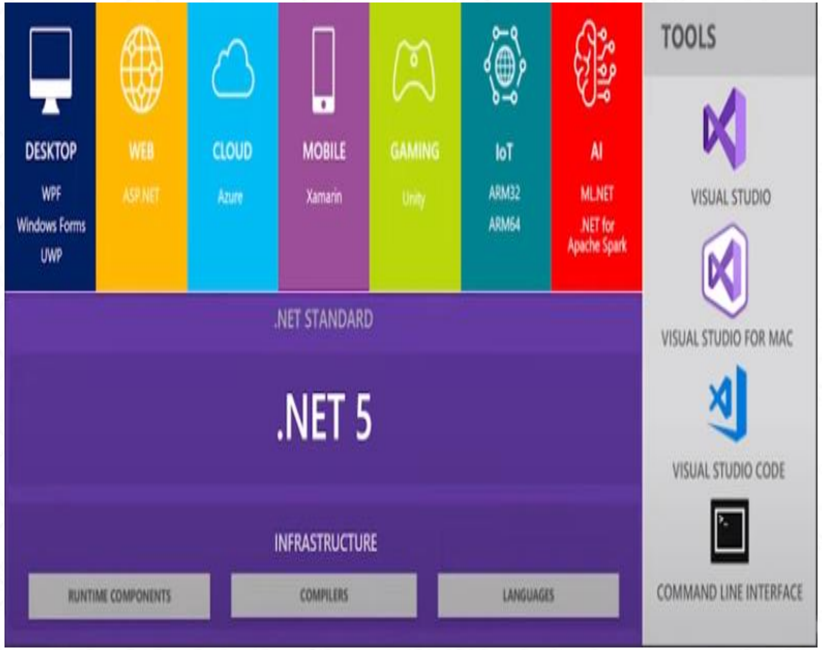
```
namespace Omantel.HR
{
    public class Payroll
    {

    }
}
```

Show output from: Build

## Compatibility

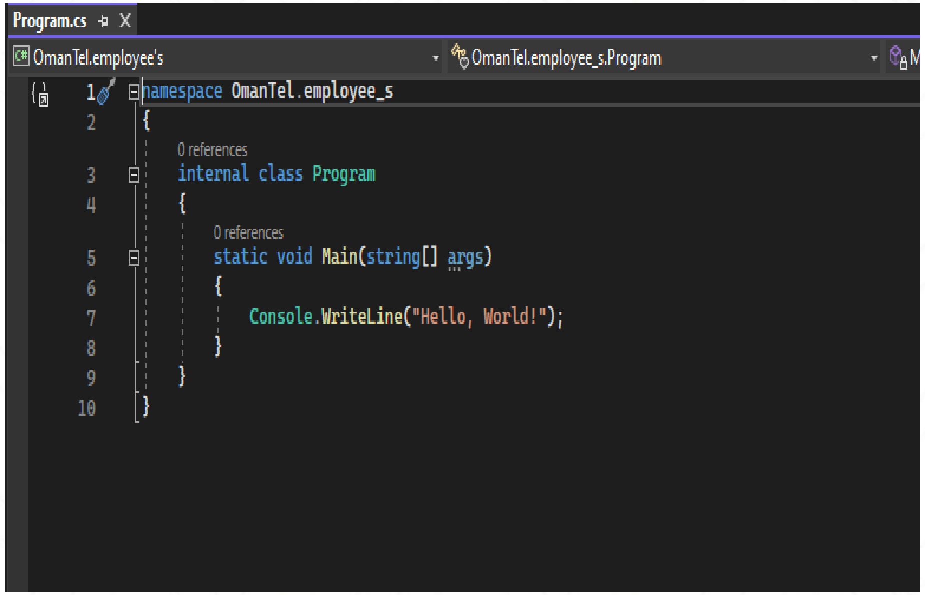
To make Your Projects Work Together you should Be Sure about you are using same Versions



Code Academy
أكاديمية البرمجة

# Let’s Toggle inside Our Project

- We Have discussed what is NameSpace .
- Internal Class Program : is a class Visual Studio Create For Us .
- Internal : it’s access Modifier that will make anything between in its bracket Accessible within the Same Project
- Main() Method : it’s the Starter For the Project **YOUR PROJECT SHOULD HAVE ONLY ONE Main() .**
- When you run the Compiler Search for the Main() to start With .



```
namespace OmanTel.employee_s
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

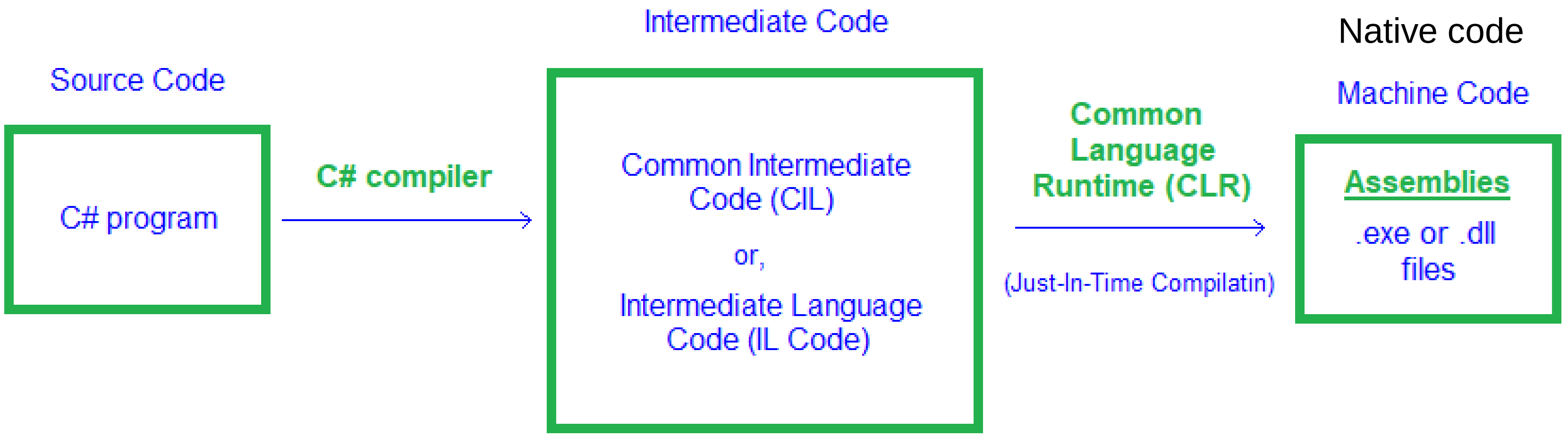
## Let’s Code in Practice

Code Academy
أكاديمية البرمجة

## C# Under the Hood

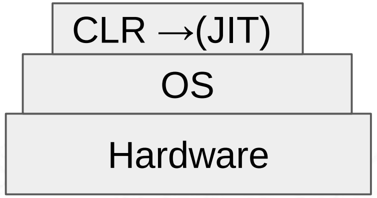
Code Academy
أكاديمية البرمجة

# What happens when we Press Build/ Run (Compilation):



Source Code
C# program
C# compiler
Intermediate Code
Common Intermediate Code (CIL)
or,
Intermediate Language Code (IL Code)
Common Language Runtime (CLR)
(Just-In-Time Compilatin)
Native code
Machine Code
Assemblies
.exe or .dll files

PlatForm independent(OS)



## Computer Memory :

HDD :HARD DISK Drive

It's a Permanent Storage

Slow access but big storage

Ram : Random access Memory

It's A Temporary Storage and Fast

And small

Code Academy
أكاديمية البرمجة

## Session Recap

Code Academy
أكاديمية البرمجة

# Any Questions ?

# Thank You

Email : youssef.mohamed@amit-learning.com

Linked-in : https://www.linkedin.com/in/youssif-mohamed-450795157/