

1. Visual Studio 2019 IDE Overview

Different types of windows (solution exp.,properties etc.)

- Code Editor: Where the user will write code.
- Output Window: Here the Visual Studio shows the outputs, compiler warnings, error messages and debugging information.
- Solution Explorer: It shows the files on which the user is currently working.
- Properties: It will give additional information and context about the selected parts of the current project.

Solution,Project

- Project : When you create an app or website in Visual Studio, you start with a project. Project contains all files that are compiled into website. Those files can include source code, icons, images, data files, and so on.
- Solution : A project is contained within a solution. simply a container for one or more related projects. It organizes projects, project items, and solution items in the solution. Solution Explorer is where you can use it to group projects in a solution.

Code editor features

- Develop.
- Build.
- Debug.
- Test.
- Improve Performance.
- Access Data.
- Design User Interfaces.
- Version control etc.

Shortcuts

- Ctrl-Tab : Move from one opened file to another opened file in visual studio.
- F9 : Sets or removes a breakpoint at the current line

- Ctrl-F9 : Enables or disables the breakpoint on the current line of code. The line must already have a breakpoint for this to work
- F5 : Runs the code with invoking the debugger.
- Ctrl-F5 : Runs the code without invoking the debugger.
- F4 or Alt-Enter : Displays the Properties window.
- Ctrl-Alt-L : Displays the Solution Explorer.
- Ctrl-K, Ctrl-D : Do proper alignment of all the code
- F7 : Switches from the design view to the code view in the editor
- Shift-F7 : Switches from the code view to the design view in the editor
- Shift-F8 or F8 : Navigate to compile time errors
- Ctrl-Shift-A : Add New Item to selected project
- F12 : Moves the cursor to the selected method, variable, class definition.
- Shift-F12 : Finds the reference to the selected method, variable, class or the item under the cursor

2. Project Types

Windows App, Class Library

- Windows Application is a user build an application that can run on a Windows platform. The windows application has a graphical user interface that is provided by Windows Forms.
- Windows forms provide a variety of controls including Button, TextBox, Radio Button, CheckBox, and other data and connection controls.
- You can easily design it in Visual Studio using a variety of languages including C#, Visual Basic, C++, J# and many more.
- A class library is a collection of class definitions contained in a .dll or .exe file.

Web Application

- It is a type of application that runs on a browser using a Web server configured with Microsoft Internet Information Services (IIS).
- In .NET, all web applications are built around ASP.NET, which is nothing but a part of the .NET platform
- It includes design-time objects and controls and a run-time execution context.

- ASP.NET helps in building a wide variety of web application from a traditional website that serves HTML pages to a high-end business application that runs on the Internet.

3. Create First C# Program "Hello World"

What is namespace?

- In C#, namespaces are used to logically arrange classes, structs and interfaces.
- The namespaces in C# can be nested.
- Declaring a Namespace : The C# language provide a keyword namespace to create a user defined name space.
- It is not possible to use any access specifiers like private, public etc with a namespace declarations. The namespaces in C# are implicitly have public access and this is not modifiable.

What is class?

- In c#, Class is a data structure, and it will combine various types of data members such as fields, properties, member function into a single unit.
- In c#, classes are declared by using class keyword.
- We have to give access specifier to class it can be any private, public, protected, internal.
- We can create required fields, properties, methods, and events to use in our applications.

Variable & Method Declaration

- Variables declared within classes are called instance variables. They belong to an instance of the class/object.
- The modifier (public, private) permits or restricts direct access to the variable with respect to its scope Variables declared within a class are called fields.
- Variables without a modifier are known as local variables, typically used within a method. They are temporary and only exist within the scope of the where its declared method.
- A method is a group of statements that together perform a task. Every C# program has at least one class with a method named Main.
- To use a method, you need to -Define the method -Call the method

4. Understanding C# Program

Program Flow

- Namespace declaration
- A class
- Class methods
- Class attributes
- A Main method
- Statements and Expressions
- Comments

Understanding Syntax

using System;

namespace HelloWorld {

class Program {

static void Main(string[] args) {

Console.WriteLine("Hello World!");

}

}

}

- using System means that we can use classes from the System namespace.
- namespace is used to organize your code, and it is a container for classes and other namespaces.
- The curly braces {} marks the beginning and the end of a block of code.
- class is a container for data and methods, which brings functionality to your program. Every line of code that runs in C# must be inside a class. In our example, we named the class Program.

- Another thing that always appear in a C# program, is the Main method. Any code inside its curly brackets {} will be executed.
- Console is a class of the System namespace, which has a WriteLine() method that is used to print text. In our example it will output "Hello World!".
- If you omit the using System line, you would have to write System.Console.WriteLine() to print/output text.
- Every C# statement ends with a semicolon ;.
- C# is case-sensitive: "MyClass" and "myclass" has different meaning.

Working with code files, projects & solutions

Understanding structure of solution

- A solution is a container for one or more related projects. Visual Studio uses to organize one or more related projects. Solution contains one or more projects.
- When you open a solution, Visual Studio automatically loads all the projects that the solution contains.
- The solution maintains the state information for projects in two files: .sln file (text-based, shared) .suo file (binary, user-specific solution options)
- The .sln file contains text-based information the environment uses to find and load the name-value parameters for the persisted data and the project VSPackages it references.

Understanding structure of project (Win app, web app, web api, class library)

- **Win app:**
 - When We create windows form App from visual studio we can see Solustion exp on right side of screen.
 - It have sections for Properties, References, and a Program.cs file.
 - Form1.cs file where we'll write most of your code
 - References contain all the references of the project.
 - Project.cs file which starts executing from the entry point public static void Main() in Program class .

- **Web app:**
 - Visual Studio uses .csproj file to manage projects ,<project-name>.csproj.
 - The csproj file includes settings related to targeted .NET Frameworks, project folders, NuGet package references etc.
 - The Properties node includes launchSettings.json file which includes Visual Studio profiles of debug settings. The following is a default launchSettings.json file.
 - By default, the wwwroot folder in the ASP.NET Core project is treated as a web root folder. Static files can be stored in any folder under the web root and accessed with a relative path to that root.
 - ASP.NET Core web application is actually a console project which starts executing from the entry point public static void Main() in Program class where we can create a host for the web application.
 - Web application must include Startup class. It is like Global.asax in the traditional .NET application. As the name suggests, it is executed first when the application starts.
- **Web api:**
 - When We create windows Api from visual studio we can see Solution exp on right side of screen.
 - It have sections for Properties, References, controller, Configuration.
 - we need to add latest Web API references using NuGet Package Manager.
 - We will add Web API controller in the Controllers folder and configuration class in the Configuration folder.
 - We have to add Global.asax file into the project. We need to configure our Web API routes when application starts.
- **Class library:**
 - Class Library have sections for Properties, References and .cs file.
 - In References we can add as many references as we can and use it for our project.
 - References are those file with .dll extension. We can add it in our class libray.

Famlier with different type of file extension

- .sln : sln extension represents a Visual Studio solution file.

- .csproj : csproj file is a C# programming project file created by Microsoft Visual Studio.
- .aspx : aspx extension is a webpage generated using Microsoft ASP.NET framework running on web servers.
- .cs : cs extension are source code files for C#.
- .exe : used on Windows computers to install or run software applications.
- .json : is a standard text-based format for representing structured data based on JavaScript object syntax.
- .dll : DLL is a library that contains code and data that can be used by more than one program at the same time.
- .config : configuration files are files used to configure the parameters and initial settings for some computer programs.

Understanding datatypes & variables with conversion

Base datatype

- In C#, these data types are categorized based on how they store their value in the memory. C# includes the following categories of data types:
- **Value type:** A data type is a value type if it holds a data value within its own memory space. It means the variables of these data types directly contain values.
 - Predefined datatype: bool , byte , char , decimal , double , float , int , long etc.
 - User defined datatype: struct , enum
- **Reference type:** Unlike value types, a reference type doesn't store its value directly. Instead, it stores the address where the value is being stored. In other words, a reference type contains a pointer to another memory location that holds the data.
 - Predefined datatype: String, Array
 - User defined datatype: Class, Interface
- **Pointer type:** A pointer is a variable that stores a memory address. Pointers are used to store the addresses of other variables or memory items.

Datatype Conversion

- Type conversion is converting one type of data to another type. It is also known as Type Casting. In C#, type casting has two forms
- **Implicit type conversion** – These conversions are performed by C# in a type-safe manner. For example, are conversions from smaller to larger integral types and conversions from derived classes to base classes.
- **Explicit type conversion** – These conversions are done explicitly by users using the pre-defined functions. Explicit conversions require a cast operator.
- C# provides the following built-in type conversion methods like ToBoolean, ToByte, ToChar, ToDateTime, ToDecimal, ToDouble and many more.

Boxing/Unboxing

- **Boxing**
 - The process of Converting a Value Type (char, int etc.) to a Reference Type(object) is called Boxing.
 - Boxing is implicit conversion process in which object type (super type) is used.
 - The Value type is always stored in Stack. The Referenced Type is stored in Heap
- **Unboxing**
 - The process of converting reference type into the value type is known as Unboxing.
 - It is explicit conversion process.

Understanding Decision making & statements

if else,switch

- **If else:** An if statement can be followed by an optional else statement, which executes when the boolean expression is false
- Syntax:

```
if(boolean_expression) {  
    /* statement(s) will execute if the boolean expression is true */  
} else {
```



```
    /* statement(s) will execute if the boolean expression is false */  
}
```

- **Switch:** A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case.
- Syntax:

```
switch(expression) {  
    case constant-expression1 :  
        statement(s);  
        break;  
    case constant-expression2 :  
    case constant-expression3 :  
        statement(s);  
        break;  
  
    default : /* Optional */  
        statement(s);  
}
```