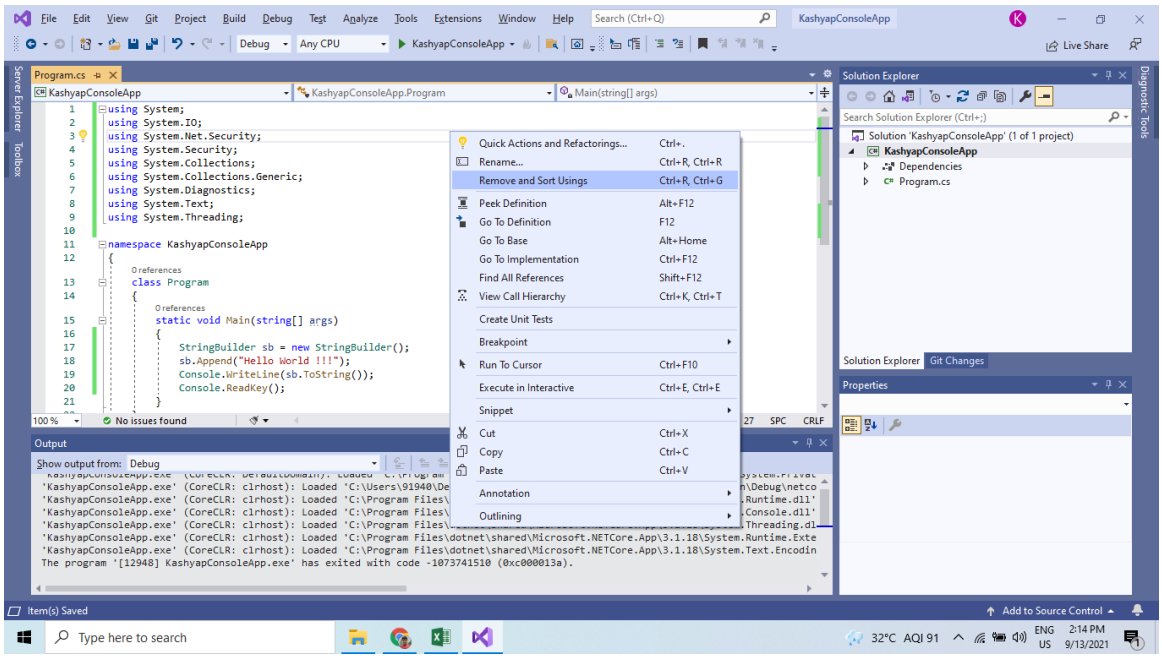
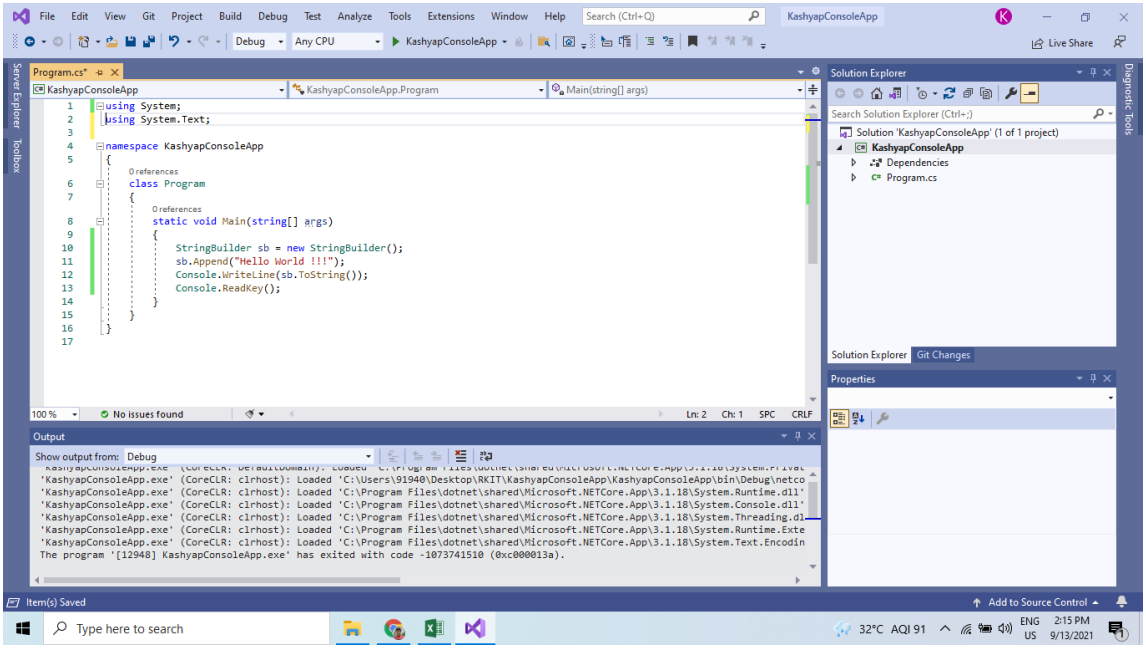
Kashyap Sayani Phase – 1 Submission

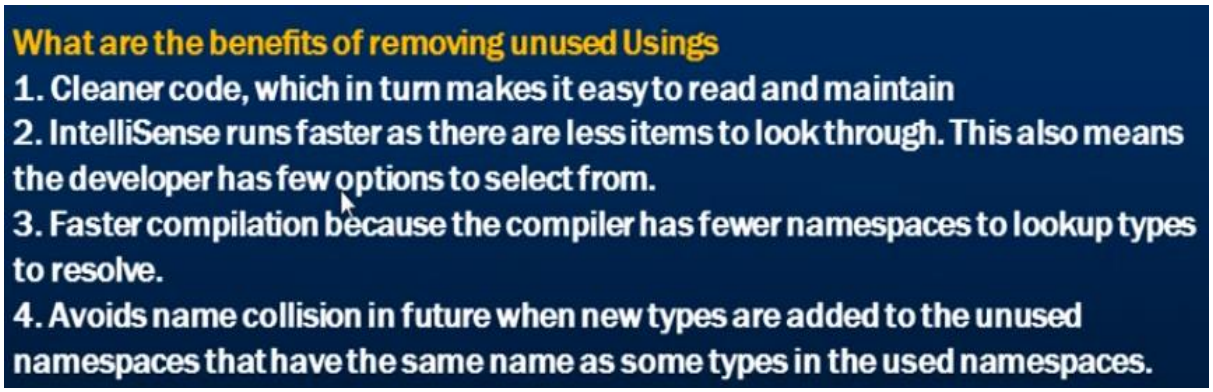
- **Visual Studio 2019 IDE Overview**

- **How to remove unused namespaces :**





Kashyap Sayani Phase – 1 Submission

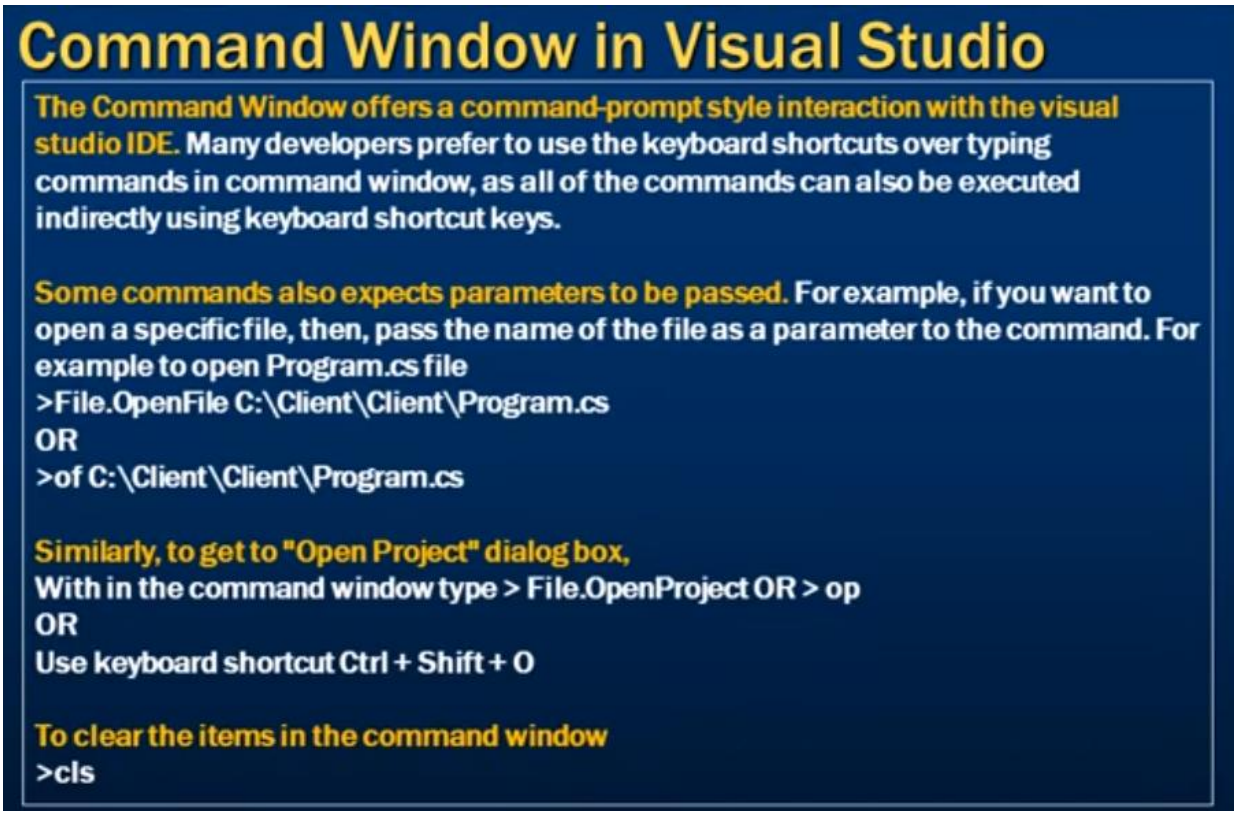


- **Visual Studio Keyboard Shortcuts :**

○ Format Document	:	**Ctrl + K, D**
○ Do Comment	:	**Ctrl + K, C**
○ Do Uncomment	:	**Ctrl + K, U**
○ Show Intellisense	:	**Ctrl + Space**
○ Attach the Debugger to a process	:	**Ctrl + Alt + P**
○ Expand Or Collapse Current Element	:	**Ctrl + M, M**
○ Collapse all	:	**Ctrl + M,O**
○ Toggle All Outlining	:	**Ctrl + M, L**
○ To Uppercase	:	**Ctrl + Shift + U**
○ To Lowercase	:	**Ctrl + U**
○ Toggle Full Screen	:	**Alt + Shift + Enter**
○ Build Your Project	:	**Ctrl + B**
○ Build Solution	:	**Ctrl + Shift + B**
○ Command Window	:	**Ctrl + Alt + A**
○ Immediate Window	:	**Ctrl + Alt + I**

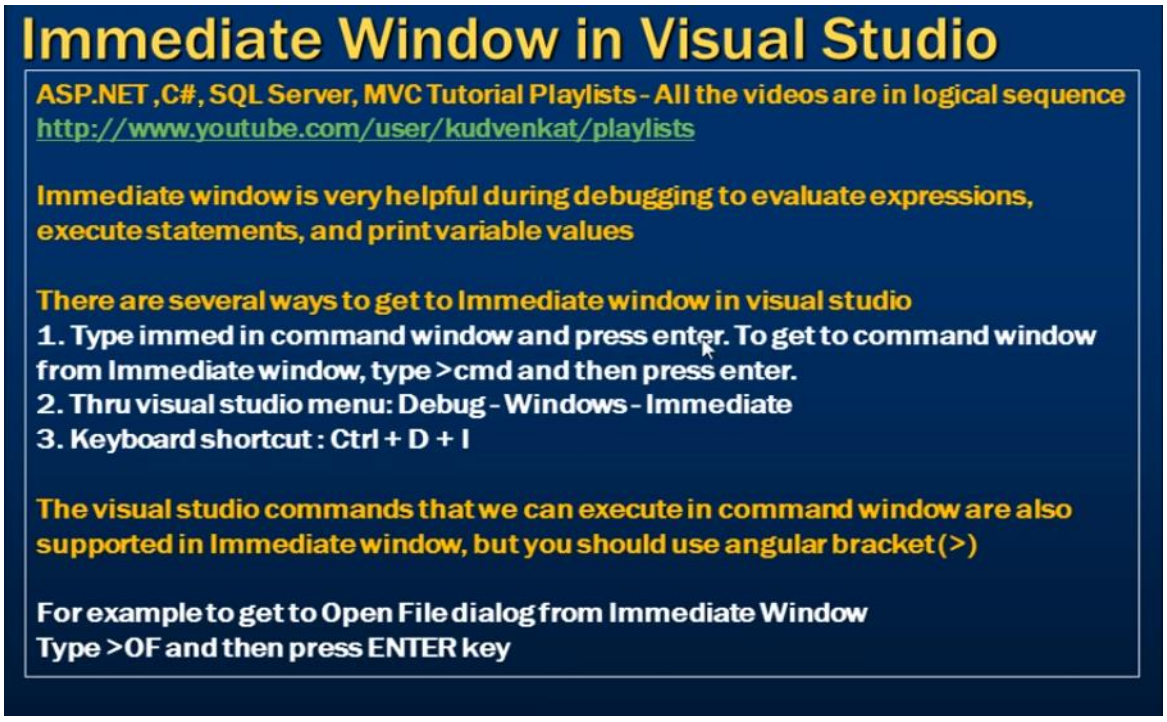
- **Windows :**

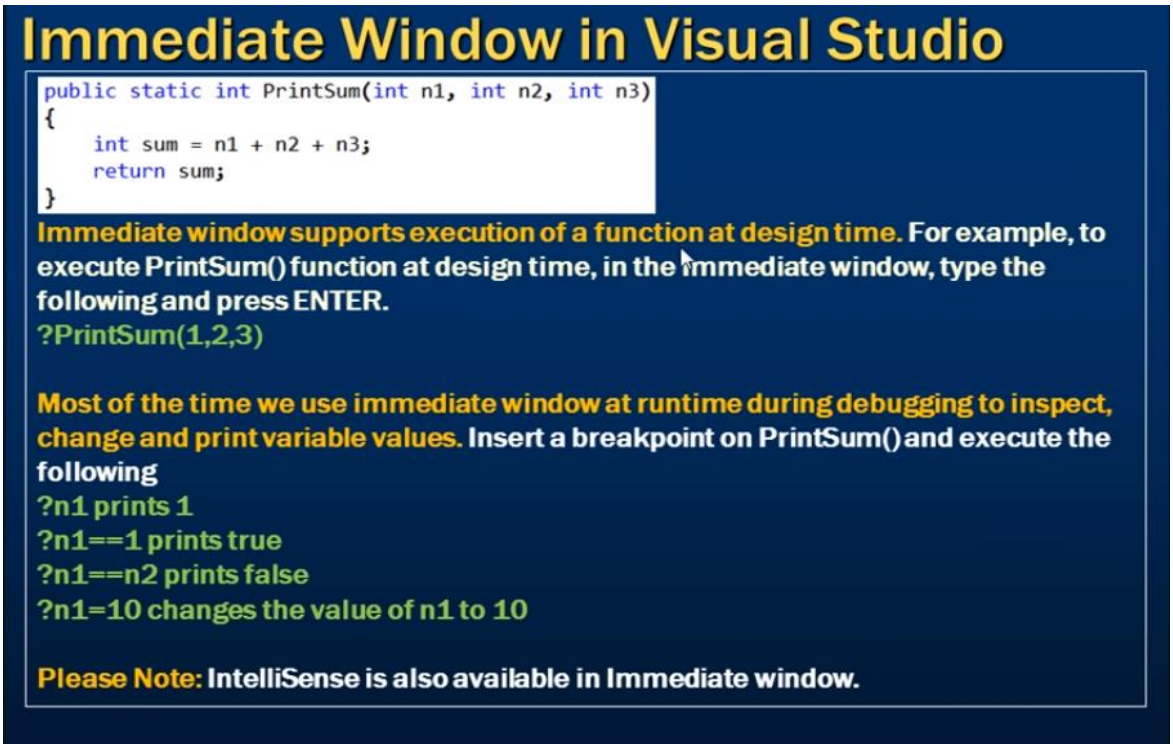
  - Command Window



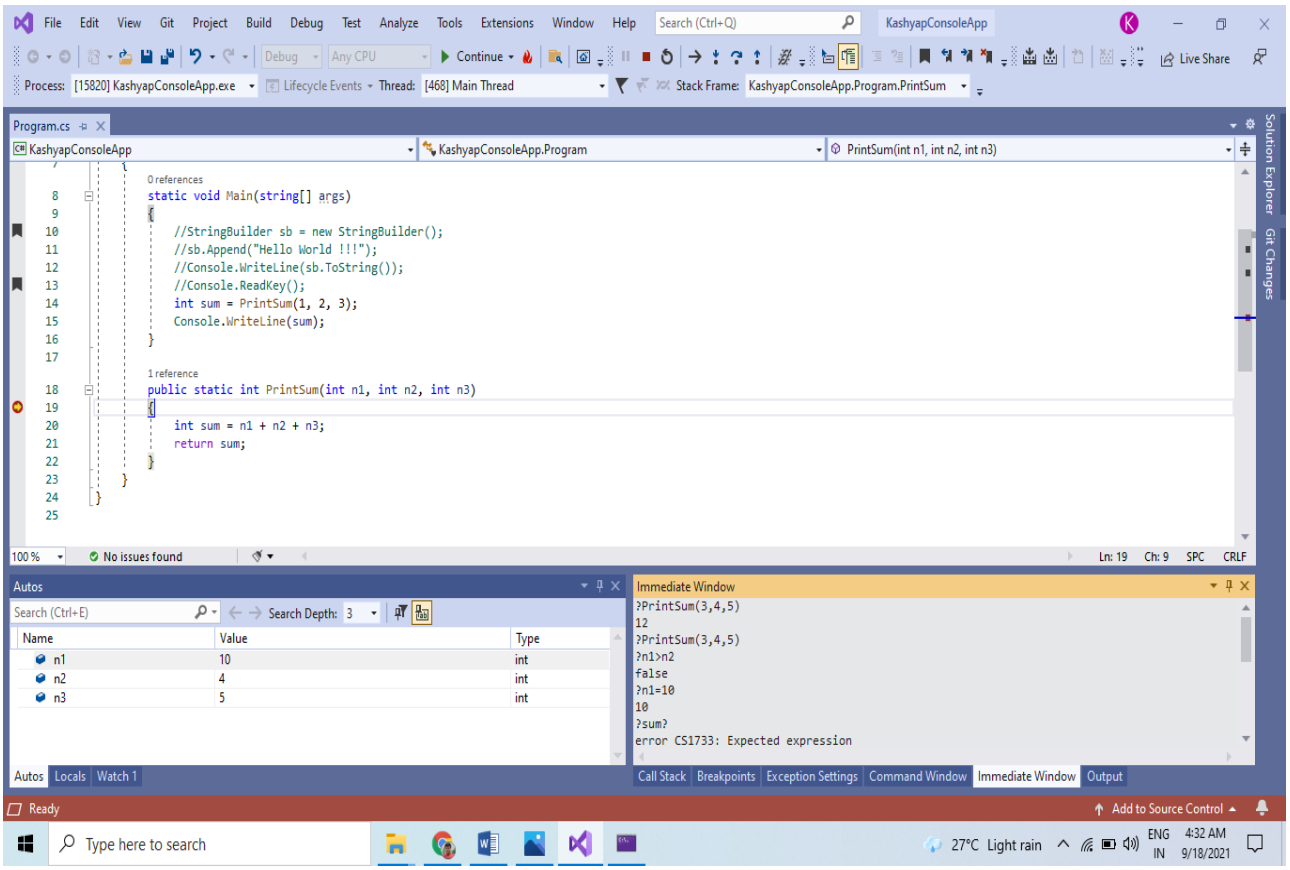
Kashyap Sayani Phase – 1 Submission

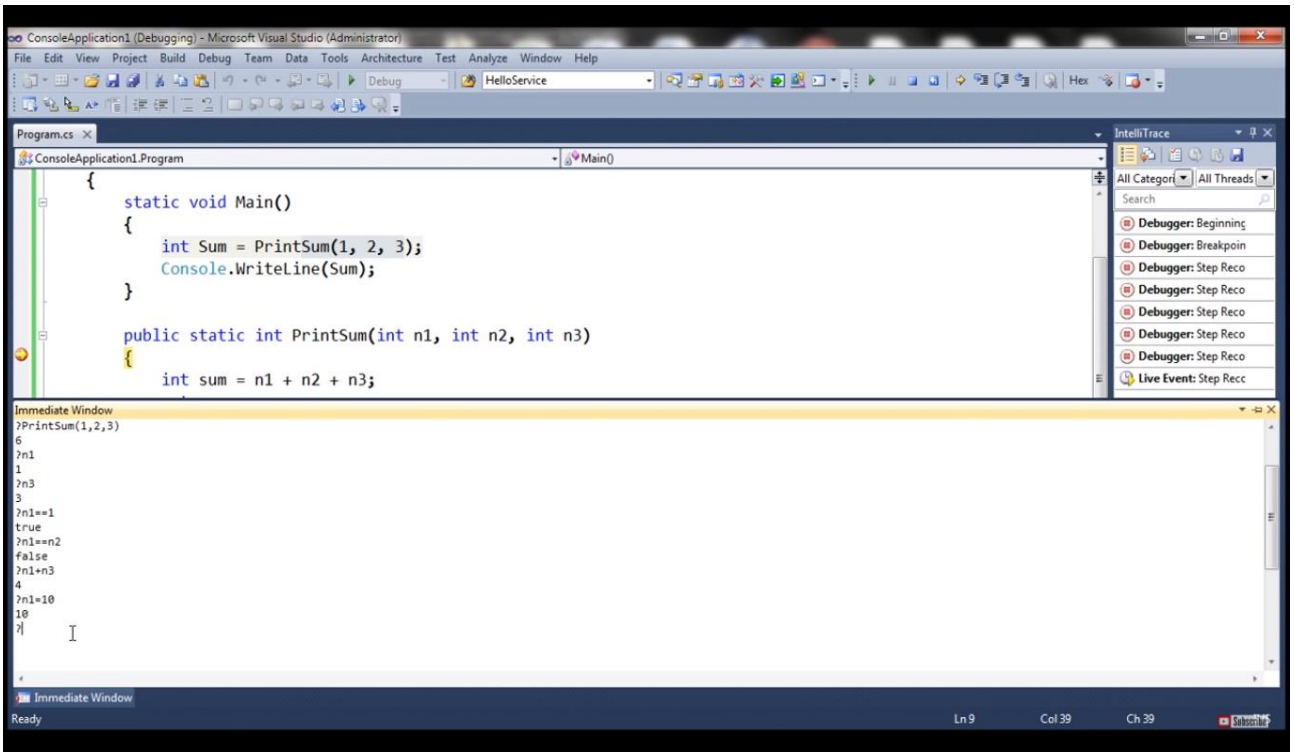
- Immediate Window





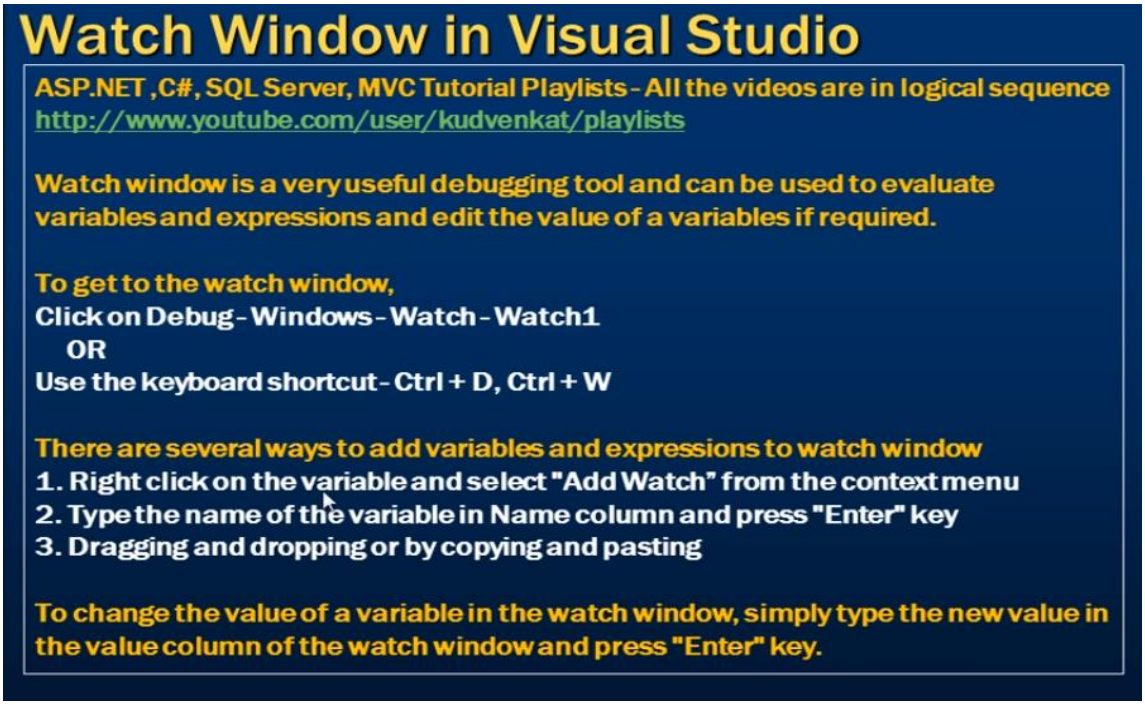
Kashyap Sayani Phase – 1 Submission

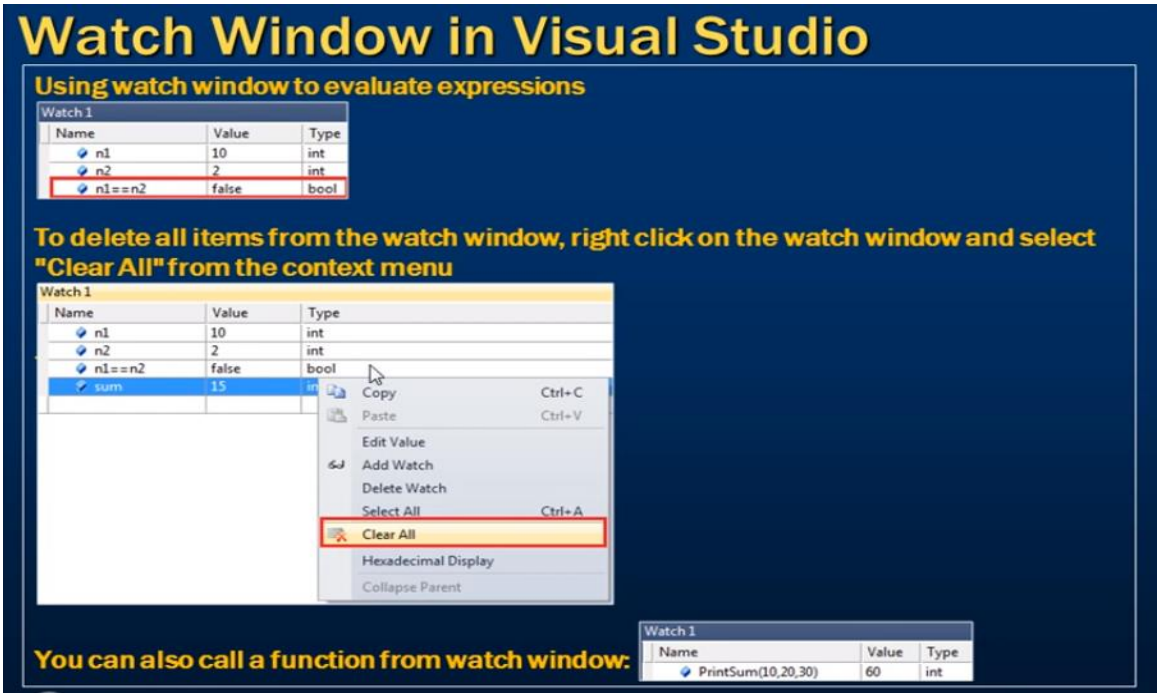




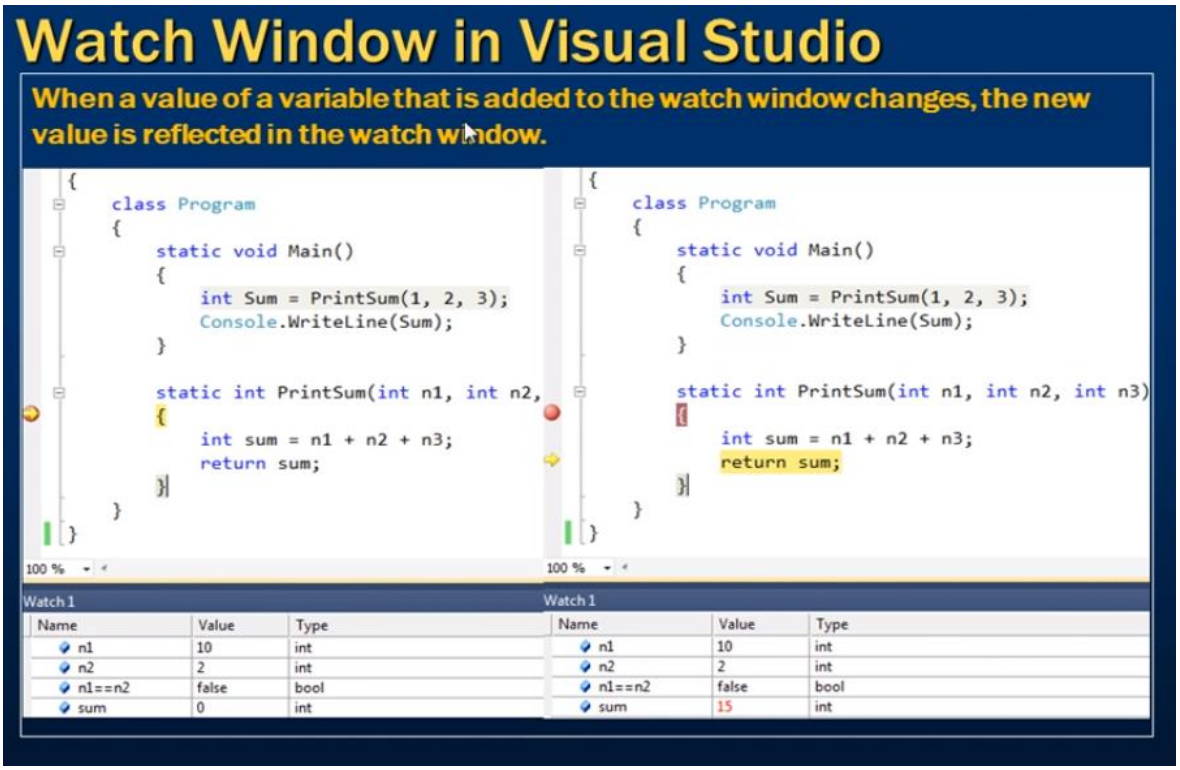
Kashyap Sayani Phase – 1 Submission

- Watch Window



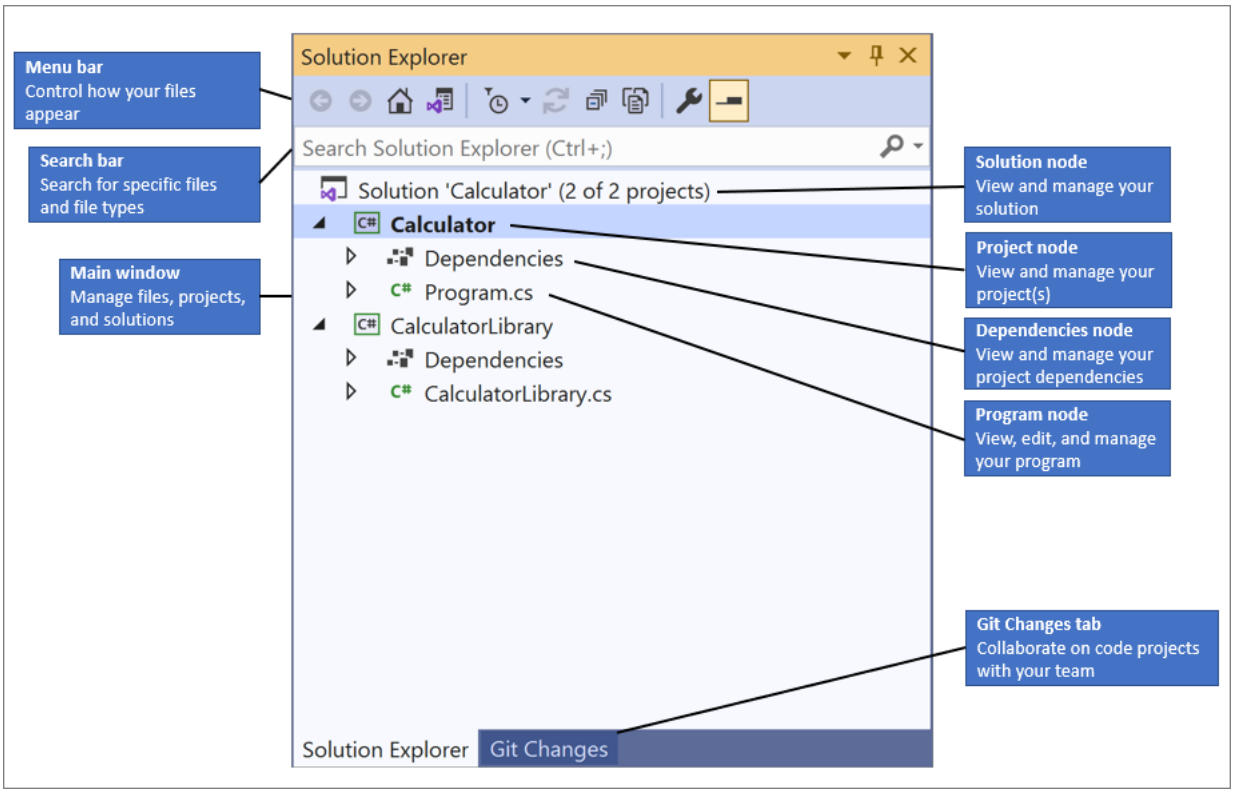


Kashyap Sayani Phase – 1 Submission



Kashyap Sayani Phase – 1 Submission

- Solution Explorer



- **Menu bar**, where you can control how your files appear
- **Search bar**, where you can search for specific files and file types
- **Main window**, where you can view and manage your files, projects, & solutions
- **Solution node**, where you can manage your solution(s)
- **Project node**, where you can manage your project(s)
- **Dependencies node**, where you can manage your solution & project dependencies
- **Program node**, where you can view, edit, and manage your program or application (app)
- **Git changes tab**, where you can use Git & GitHub within Visual Studio to collaborate on projects with your team

Projects

When you create an app or website in Visual Studio, you start with a *project*. In a logical sense, a project contains all files that are compiled into an executable, library, or website. Those files can include source code, icons, images, data files, and so on. A project also contains compiler settings and other configuration files that might be needed by various services or components that your program communicates with.

Kashyap Sayani Phase – 1 Submission

Solutions

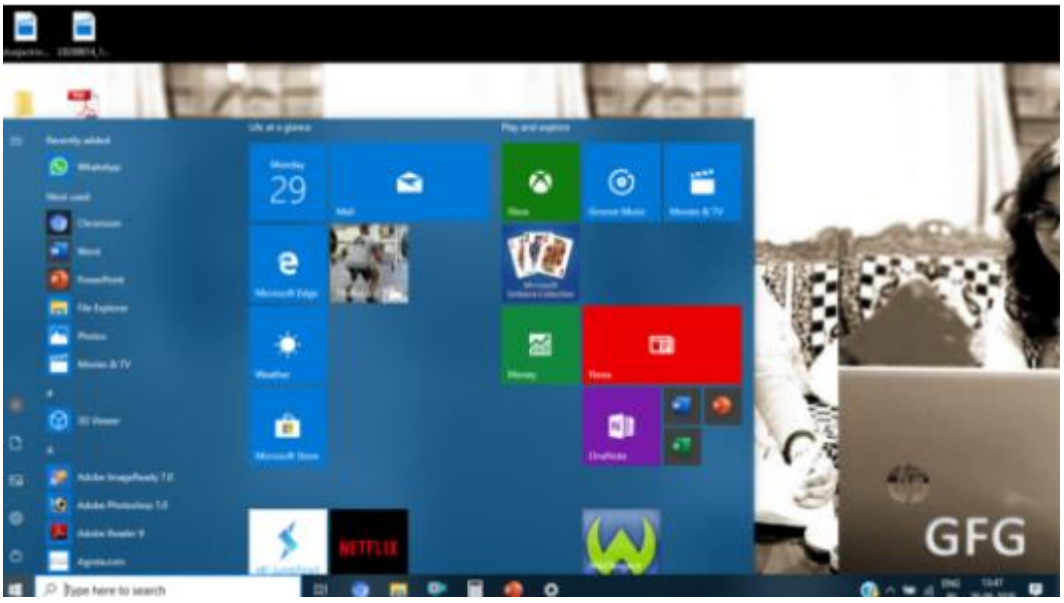
A project is contained within a *solution*. Despite its name, a solution is not an "answer". It's simply a container for one or more related projects, along with build information, Visual Studio window settings, and any miscellaneous files that aren't associated with a particular project.

Code editor features

- Create cloud-enabled Azure apps
- Create Web Apps
- Build cross-platform apps and games
- Connect to databases
- Debug, test and improve your code
- Deploy your finished application
- Manage your source code and collaborate with others

• Project Types

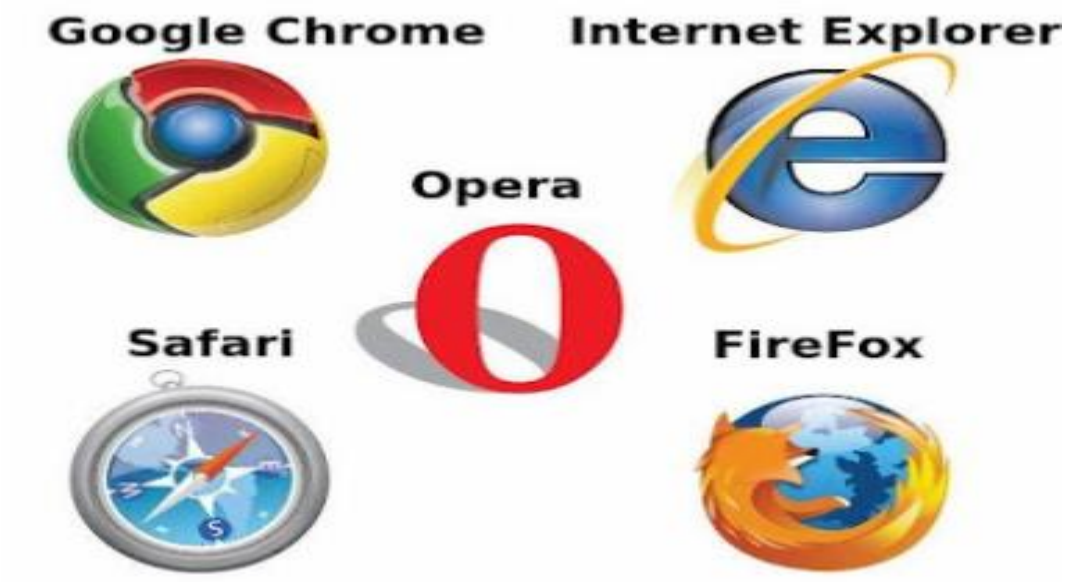
- Windows App: Windows Application is **a user builds an application that can run on a Windows platform**. The windows application has a graphical user interface that is provided by Windows Forms. Windows forms provide a variety of controls including Button, TextBox, Radio Button, CheckBox, and other data and connection controls.



Kashyap Sayani Phase – 1 Submission

- Class Library: A class library is a collection of prewritten classes or coded templates, any of which can be specified and used by a programmer when developing an application program. The **Class Library .DLL** contains program code, data, and resources that can be can used by other programs and are easily implemented into other Visual Studio projects.

- Web Application: Web application is an application that runs on web browser making use of web server. It makes use of is Microsoft IIS configuration i.e., Internet Information Services (in developing web applications). A variety of web applications using .net can be made. These include many ranging from simple HTML pages to highly interactive business applications.



• Create First C# Program "Hello World"

- What is namespace?

  In C#, namespaces are used to logically arrange classes, structs, interfaces, enums and delegates. The namespaces in C# can be nested.

  The biggest advantage of using namespace is that the class names which are declared in one namespace will not clash with the same class names declared in another namespace.

  Syntax: namespace NameOfNameSpace
  {
  …
  }

- What is class?

  A class is like a blueprint of a specific object.

  https://www.tutorialspoint.com/csharp/csharp_classes.htm

  A class defines some properties, fields, events, methods, etc. A class defines the kinds of data and the functionality their objects will have.

  A class enables you to create your custom types by grouping variables of other types, methods, and events.

  In C#, a class can be defined by using the class keyword.

  Syntax: namespace NameOfNameSpace
  {
  class ClassName

```
        {       …       }
    }
```

- Variable & Method Declaration:

  Variables are containers for storing data values.

  https://www.tutorialspoint.com/csharp/csharp_variables.htm

  In C#, there are different **types** of variables (defined with different keywords), for example:

  - int - stores integers (whole numbers), without decimals, such as 123 or -123
  - double - stores floating point numbers, with decimals, such as 19.99 or -19.99
  - char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
  - string - stores text, such as "Hello World". String values are surrounded by double quotes
  - bool - stores values with two states: true or false

  Declaration: type variableName = value

  A **method** is a block of code which only runs when it is called.

  https://www.tutorialspoint.com/csharp/csharp_methods.htm

  You can pass data, known as parameters, into a method.

  Declaration:

  ```
  <Access Specifier> <Return Type> <Method Name> (Parameter List)
  {
    Method Body
  }
  ```

  - **Access Specifier** −This determines the visibility of a variable or a method from another class.

- **Return type** − A method may return a value. The return type is the data type of the value the method returns. If the method is not returning any values, then the return type is **void**.
- **Method name** − Method name is a unique identifier and it is case sensitive. It cannot be same as any other identifier declared in the class.
- **Parameter list** − Enclosed between parentheses, the parameters are used to pass and receive data from a method. The parameter list refers to the type, order, and number of the parameters of a method. Parameters are optional; that is, a method may contain no parameters.
- **Method body** −This contains the set of instructions needed to complete the required activity.

Kashyap Sayani Phase – 1 Submission

- ## Understanding C# Program

- https://www.tutorialspoint.com/csharp/csharp_program_structure.htm

- ## Understanding datatypes & variables with conversion

- Data Types: https://www.tutorialspoint.com/csharp/csharp_data_types.htm

- Type Conversion: https://www.tutorialspoint.com/csharp/csharp_type_conversion.htm

- Boxing/Unboxing: https://www.geeksforgeeks.org/c-sharp-boxing-unboxing/

- ## Understanding Decision making & statements

- https://www.tutorialspoint.com/csharp/csharp_decision_making.htm

• Working with code files, projects & solutions

- Understanding structure of solution

Solutions:

- A Solution contains a collection of projects, along with information on dependencies between those projects.
- The project themselves contain
- files. We can create many projects in solution

Note: We can one open one solution at a time in a particular instance of Visual studio.

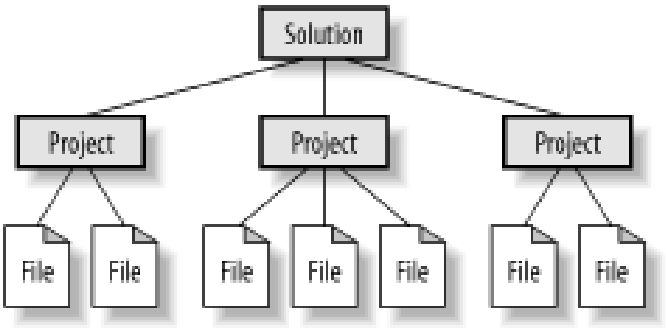


Figure show the Solution

- Understanding structure of Project:

Windows App:

- Windows application is used widely.
- There are many architectural for windows application.

Controls:

- In the Windows application there are many controls like textbox,
- edit text menu etc. This are used to complete the functionally of users.

Forms: This is the actually pages of Windows Application

Dependencies:

- Here we can find the projects dependencies.

Kashyap Sayani Phase – 1 Submission

Program.cs:

- This is the Main Entry Point of the application.
- Web Applications:
  - Using Web application, we can build
  - a website. In the Website Project there is many folders.

Appsettings.json:

- Appsettings json file is an application configuration file used to store configuration settings such as database connections strings.
- Using this file, we can declare the value globally.

Dependencies:

- Here we can find the projects dependencies.

Wwwroot folder:

- CSS: this folder has the css file of the website.
- JS: This folder has the JavaScript files of the website.

Project.json file:

- It contains the information about the project.

- Familiar with different type of file extensions

Name	Use
.html,.htm	Html web file
.asp	Active server page file
.css	Cascading Style Sheets
.txt	A text file
.sln	Solution file
.csproj	C# Project
.png	A image file