

C#-2

A screenshot of the Visual Studio IDE showing the code for a Console Application named 'ConsoleApp'. The code defines a class 'ReadValueDemo' with a static void Main() method. The code reads a name from the console and prints a welcome message. A red curly brace highlights the 'using' statements at the top of the file. A red bracket groups the 'using' statements, and a pink annotation to the right says: 'you can see that we are not using them so we can remove them'.

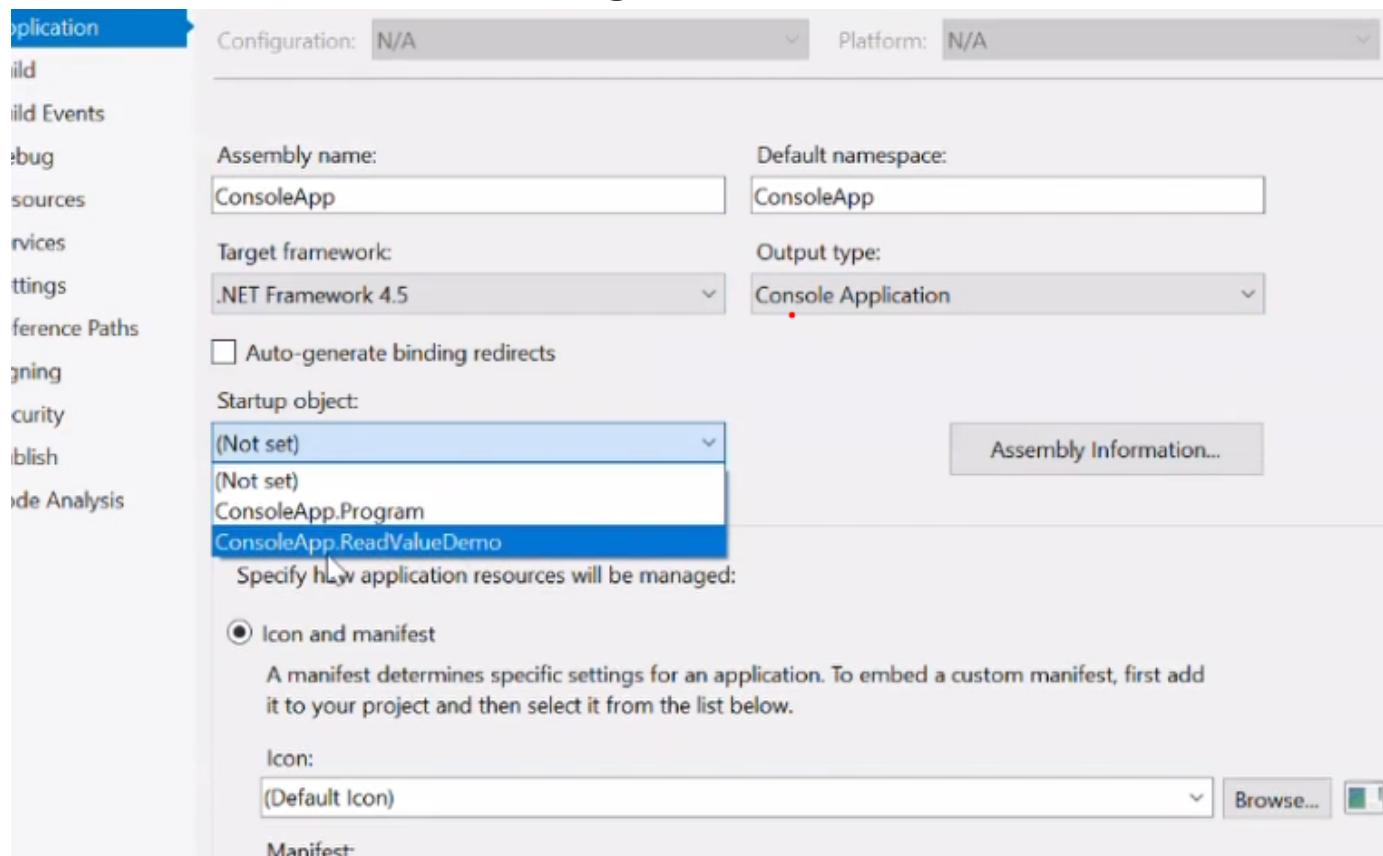
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class ReadValueDemo
10     {
11         static void Main()
12         {
13             string name;
14             Console.WriteLine("Enter your name:");
15             name = Console.ReadLine();
16             Console.WriteLine("Hello " + name + " Welcome to C#");
17         }
18     }
19 }
20
```

A screenshot of the Visual Studio IDE showing the code for 'ConsoleApp'. The code is identical to the one above. A red circle highlights the 'CS0012' error in the Error List, which states: 'Program has more than one entry point defined. Compile with /main to specify the type that contains the entry point.' The Solution Explorer shows the project structure with files like Program.cs and ReadValueDemo.cs.

```
1  using System;
2
3  namespace ConsoleApp
4  {
5      class ReadValueDemo
6      {
7          static void Main()
8      }
9 }
```

when i run this you can see its saying boss you have more than one entry point , does your hose has 2 main door----NO so here i have 2 main door main of my read more demo and main of my program, so my application confuse that both are

main so with which i should go



so for this we just have to select the startup project

```
1 using System;
2
3 namespace ConsoleApp
4 {
5     class Program
6     {
7         static void Main()
8         {
9             Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
10            Console.Write("Welcome");
11            Console.ReadLine();
12        }
13    }
14 }
15
16
17 }
```

if you want to know more about console so just hover on it or hold ctrl key and you will see that it becomes a hyperlink click on this , it will open the library, here you can see what are the methods which are available
you can see that it is a static class under the namespace called system

internally my name space system is also using some of the classes like like [system.IO](#), [system.security](#), [system.text](#).
this is in the assembly mscorelib with the version 4.0

```
ConsoleApp Program.cs* ReadValueDemo.cs
mscorlib System.Console
1  Assembly mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
4
5  using ...
8
9  namespace System
10 {
11     public static class Console
12     {
13         public static int WindowWidth { get; set; }
14         public static bool IsOutputRedirected { get; }
15         public static bool IsErrorRedirected { get; }
16         public static TextReader In { get; }
17         public static TextWriter Out { get; }
18         public static TextWriter Error { get; }
19         public static Encoding InputEncoding { get; set; }
20         public static Encoding OutputEncoding { get; set; }
21         public static ConsoleColor BackgroundColor { get; set; }
22         public static ConsoleColor ForegroundColor { get; set; }
23         public static int BufferHeight { get; set; }
24         public static int BufferWidth { get; set; }
25         public static int WindowHeight { get; set; }
26         public static bool TreatControlCAsInput { get; set; }
27     }
28 }
```

suppose you want to know what my WriteLine does so just come here

you can see that we have write method and WriteLine method
SO , can we have same method name with different parameters
is that possible ?

-----this is called **Method Overloading** -----same
function with different functionality .

so look here how many overloads we have here and they all are
static .

The screenshot shows the Microsoft Visual Studio IDE. The code editor displays the `System.Console` class from the `mscorlib` namespace, listing various static methods like `Write`, `WriteLine`, and `ReadLine`. The Task List window on the right side shows a single item: `Console [from metadata]`. The status bar at the bottom indicates the current file is `ConsoleApp`.

```
public static void Write(char value);
public static void Write(char[] buffer);
public static void Write(char[] buffer, int index, int count);
public static void Write(string format, object arg0, object arg1, object arg2);
public static void Write(decimal value);
public static void Write(float value);
public static void Write(double value);
public static void WriteLine();
public static void WriteLine(float value);
public static void WriteLine(int value);
public static void WriteLine(uint value);
public static void WriteLine(long value);
public static void WriteLine(ulong value);
public static void WriteLine(object value);
public static void WriteLine(string value);
public static void WriteLine(string format, object arg0);
public static void WriteLine(string format, object arg0, object arg1, object arg2);
public static void WriteLine(string format, object arg0, object arg1, object arg2, object arg3);
public static void WriteLine(string format, params object[] arg);
public static void WriteLine(char[] buffer, int index, int count);
public static void WriteLine(decimal value);
public static void WriteLine(char[] buffer);
```

why static?

--so that we can access this method without creating an instance of it

that's a reason before using `WriteLine` we will not write
`console c = new console();`

The screenshot shows the `Main()` method of the `Program` class. The code uses `Console.WriteLine` to print a multi-line string. The `Console` class is highlighted in red, indicating it is not yet defined.

```
using System;
namespace ConsoleApp
{
    class Program
    {
        static void Main()
        {
            Console c = new Console();
            Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
            Console.WriteLine("Hi");
            Console.WriteLine("Hello");
            Console.Write("Welcome");
            Console.ReadLine();
        }
    }
}
```

Remember whenever you write any class and if that particular class has some kind of methods

for example - i have a particular class Demo and it has a show() method

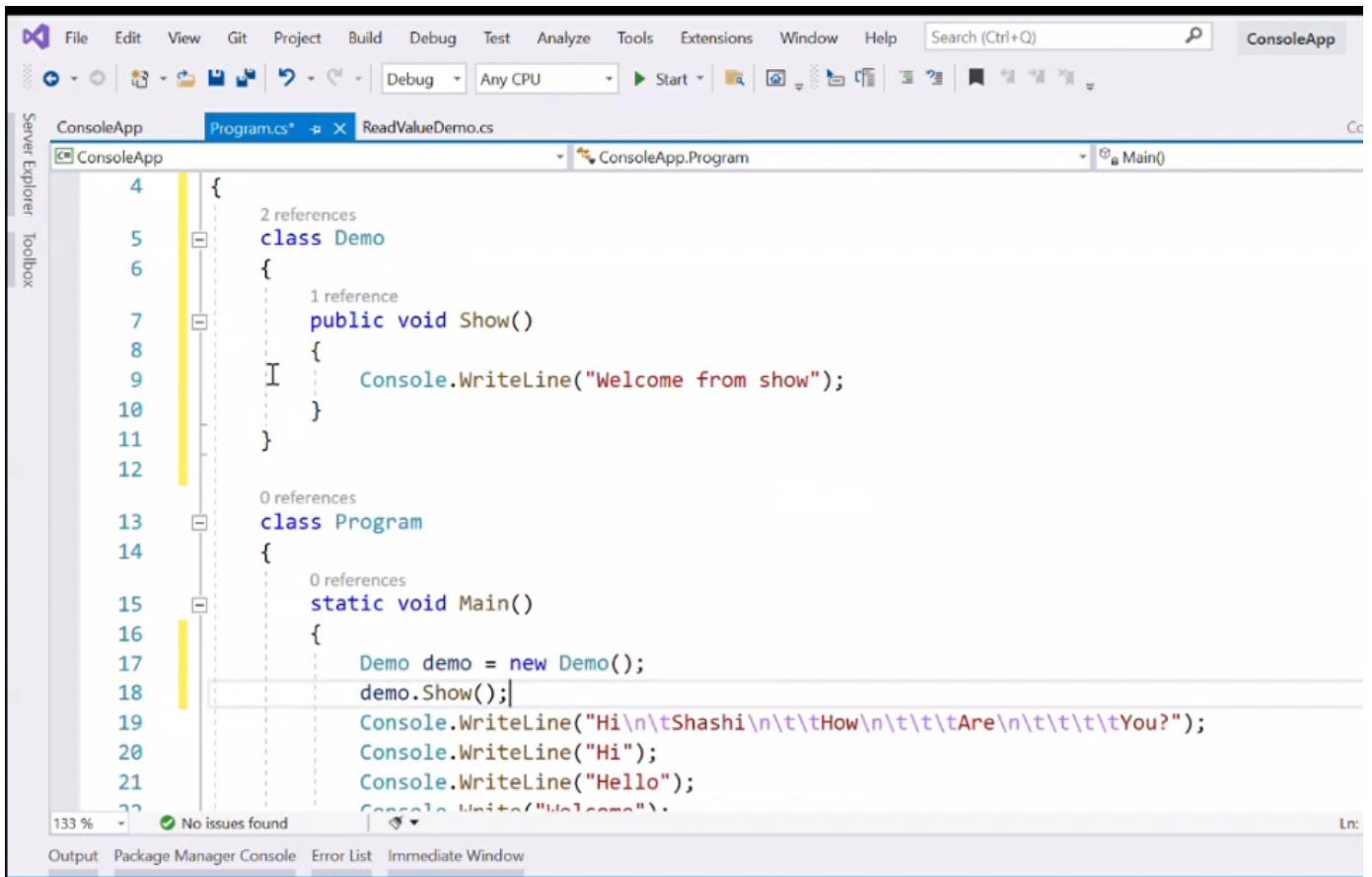
The screenshot shows the code editor in Visual Studio for a project named 'ConsoleApp'. The code is as follows:

```
1 using System;
2
3 namespace ConsoleApp
4 {
5     class Demo
6     {
7         public void Show()
8         {
9             Console.WriteLine("Welcome from show");
10        }
11    }
12
13 class Program
14 {
15     static void Main()
16     {
17         Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
18         Console.WriteLine("Hi");
19         Console.WriteLine("Hello");
20     }
21 }
```

The 'Main()' method in the 'Program' class calls the 'Show()' method in the 'Demo' class. The code editor highlights the 'Show()' method with a yellow vertical bar. The status bar at the bottom left indicates 'No issues found'.

Now i want to call this particular show method() , so where is my show method-----in the demo class

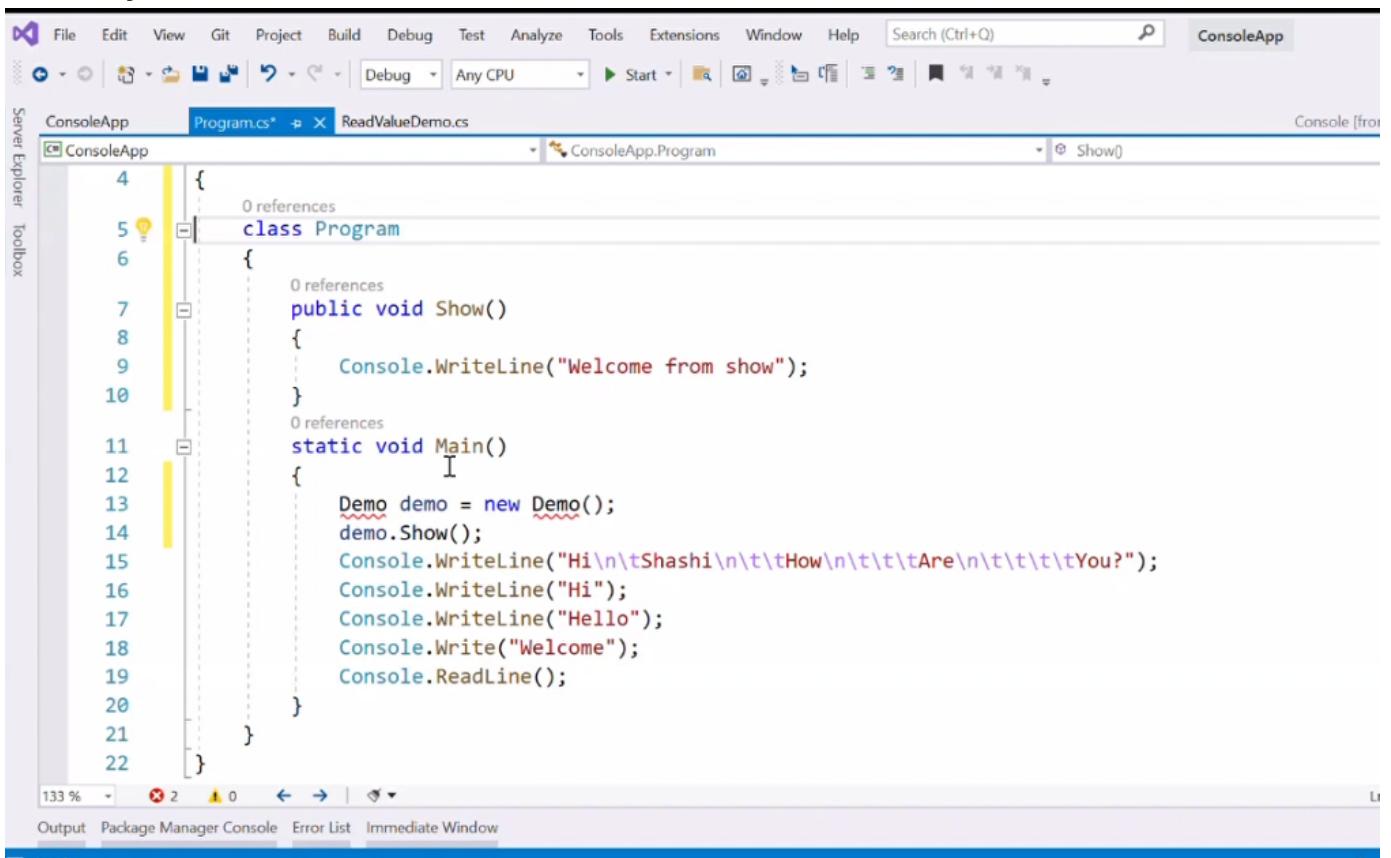
Now, i want to call this demo class --> show method , so i need to create object of my demo class



```
ConsoleApp Program.cs  ReadValueDemo.cs  ConsoleApp.Program  Main()
4 {
5     2 references
6     class Demo
7     {
8         1 reference
9         public void Show()
10        {
11            Console.WriteLine("Welcome from show");
12        }
13    }
14
15    0 references
16    class Program
17    {
18        0 references
19        static void Main()
20        {
21            Demo demo = new Demo();
22            demo.Show();
23            Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
24            Console.WriteLine("Hi");
25            Console.WriteLine("Hello");
26            Console.WriteLine("Welcome");
27            Console.ReadLine();
28        }
29    }
}
133 %  No issues found  Ln:
```

Output Package Manager Console Error List Immediate Window

imagine this `Show()` method is present in this class program only and if i want to access this `Show` method which is in the `Program` class you cannot access them

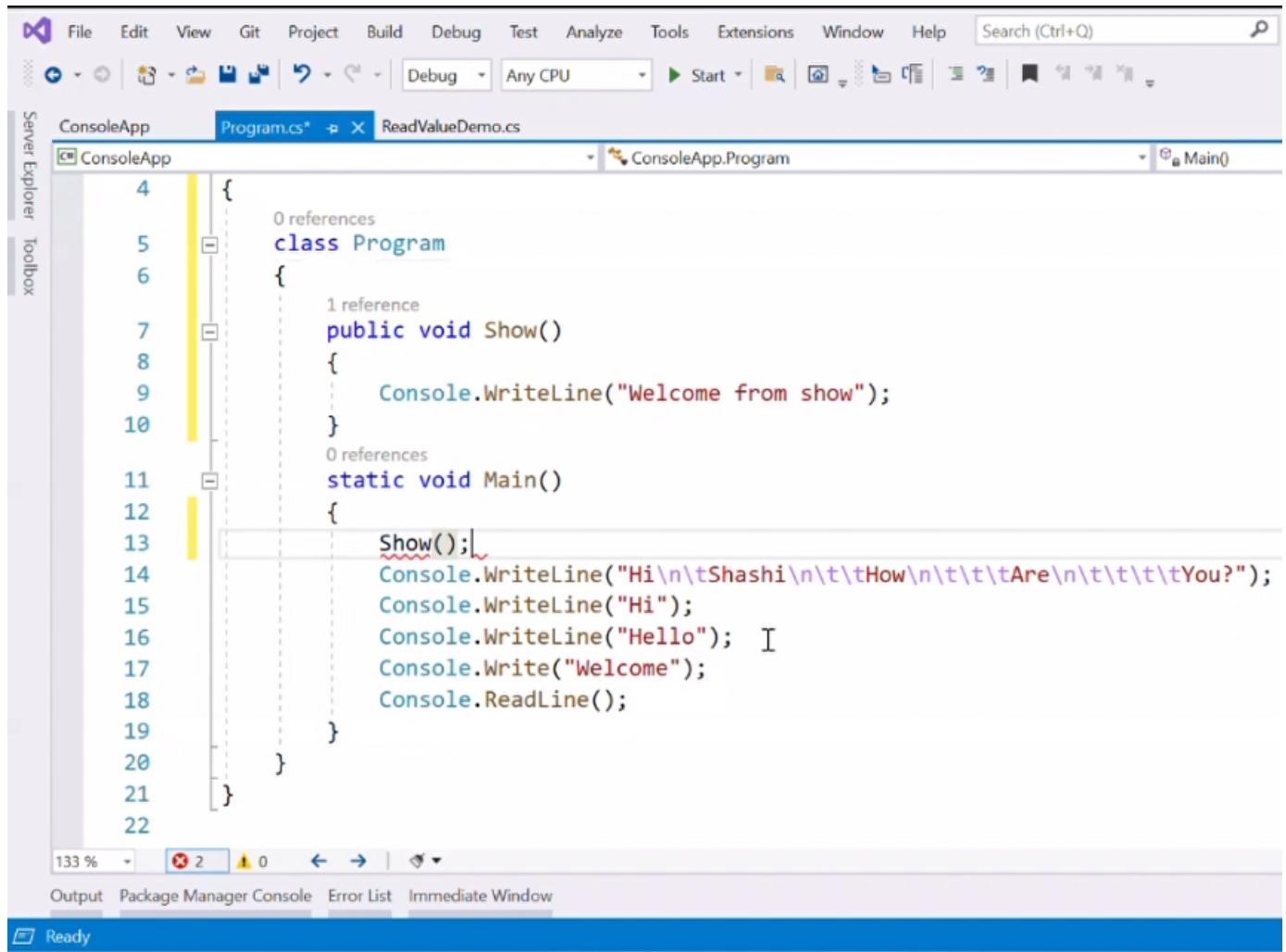


```
ConsoleApp Program.cs  ReadValueDemo.cs  ConsoleApp.Program  Show()
4 {
5     0 references
6     class Program
7     {
8         0 references
9         public void Show()
10        {
11            Console.WriteLine("Welcome from show");
12        }
13    }
14
15    0 references
16    static void Main()
17    {
18        Demo demo = new Demo();
19        demo.Show();
20        Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
21        Console.WriteLine("Hi");
22        Console.WriteLine("Hello");
23        Console.WriteLine("Welcome");
24        Console.ReadLine();
25    }
}
133 %  2  0  Ln:
```

Output Package Manager Console Error List Immediate Window

even it is in same class you cannot directly say `Show`

Why because this method is a static method



The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays a C# code editor with the file 'Program.cs' open. The code defines a class 'Program' with a static void method 'Main()' and a non-static void method 'Show()'. The 'Main()' method contains several console output statements. The code editor has syntax highlighting and line numbers. A yellow vertical bar highlights the first few lines of the 'Main()' method. The status bar at the bottom shows '133 %' zoom level, 2 errors, and 0 warnings.

```
4  {
5      0 references
6      class Program
7      {
8          1 reference
9          public void Show()
10         {
11             Console.WriteLine("Welcome from show");
12         }
13         0 references
14         static void Main()
15         {
16             Show();~
17             Console.WriteLine("Hi\n\tShashi\n\tHow\n\t\tAre\n\t\tYou?");~
18             Console.WriteLine("Hi");
19             Console.WriteLine("Hello"); I
20             Console.Write("Welcome");
21             Console.ReadLine();
22         }
23     }
24 }
```

you have to create instance if it otherwise you cannot access that

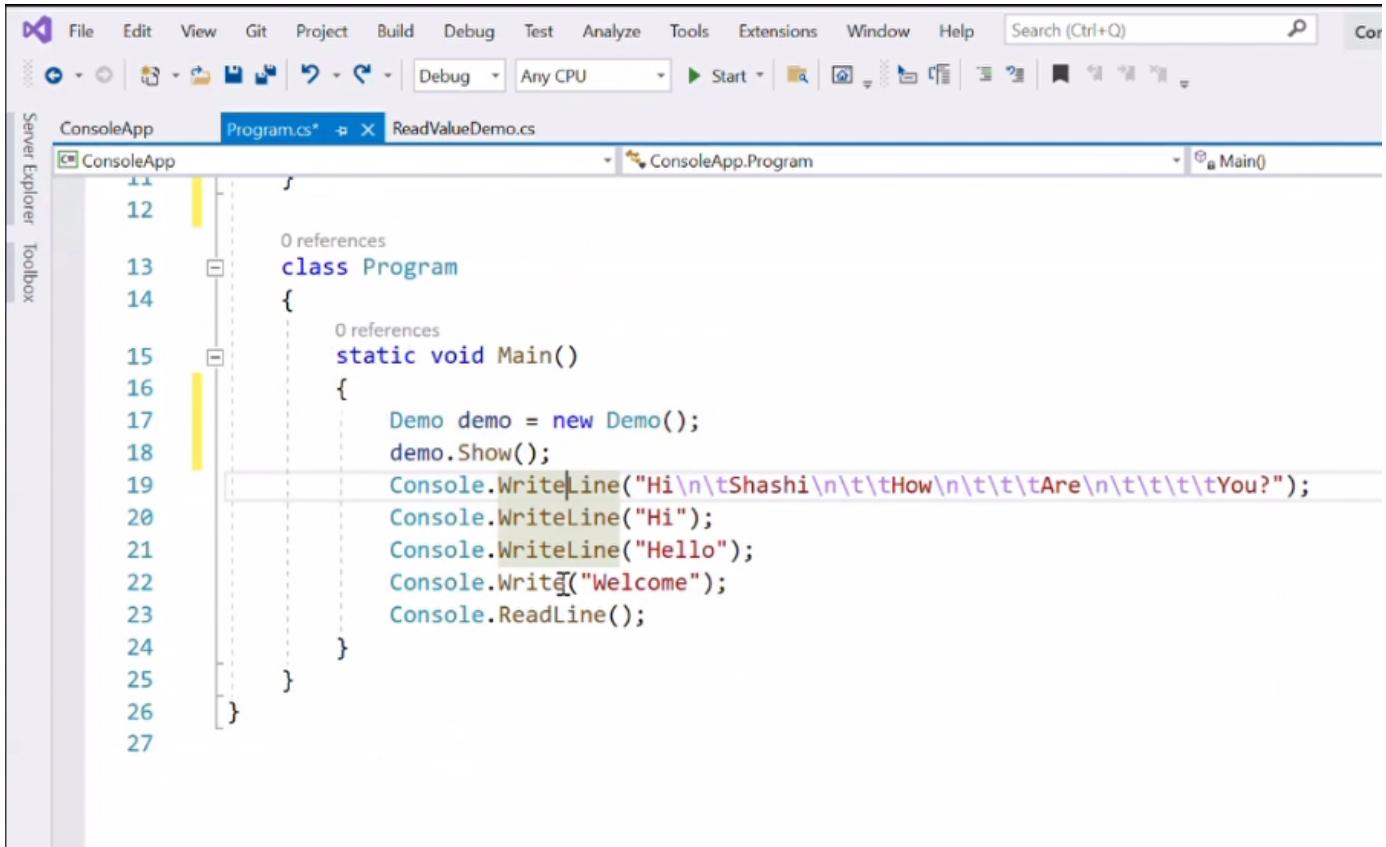
```
ConsoleApp Program.cs* ReadValueDemo.cs
ConsoleApp ConsoleApp.Demo
4 {
2 references
5 class Demo
6 {
1 reference
7     public void Show()
8     {
9         Console.WriteLine("Welcome from show");
10    }
11 }
12
0 references
13 class Program
14 {
15     static void Main()
16     {
17         Demo demo = new Demo();
18         demo.Show();
19         Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
20         Console.WriteLine("Hi");
21         Console.WriteLine("Hello");
22         Console.WriteLine("Welcome");
}
No issues found
```

But where as you look into your console class here , so this is a different class overall

```
ConsoleApp Program.cs* ReadValueDemo.cs
mscorlib System.Console
1 Assembly mscorelib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
4
5 using ...
8
9 namespace System
10 {
11     public static class Console
12     {
13         public static int WindowWidth { get; set; }
14         public static bool IsOutputRedirected { get; }
15         public static bool IsErrorRedirected { get; }
16         public static TextReader In { get; }
17         public static TextWriter Out { get; }
18         public static TextWriter Error { get; }
19         public static Encoding InputEncoding { get; set; }
20         public static Encoding OutputEncoding { get; set; }
21         public static ConsoleColor BackgroundColor { get; set; }
22         public static ConsoleColor ForegroundColor { get; set; }
23         public static int BufferHeight { get; set; }
24         public static int BufferWidth { get; set; }
25         public static int WindowHeight { get; set; }
26         public static bool TreatControlCAsInput { get; set; }
}
No issues found
```

but to access those methods i am not bothered i'll just say
console.WriteLine Why ? -----because it is a static class and

static methods



```
11
12
13 class Program
14 {
15     static void Main()
16     {
17         Demo demo = new Demo();
18         demo.Show();
19         Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
20         Console.WriteLine("Hi");
21         Console.WriteLine("Hello");
22         Console.WriteLine("Welcome");
23         Console.ReadLine();
24     }
25 }
26
27 }
```

So , what is namespace ---- So we'll put all the logical classes together and that becomes my namespace

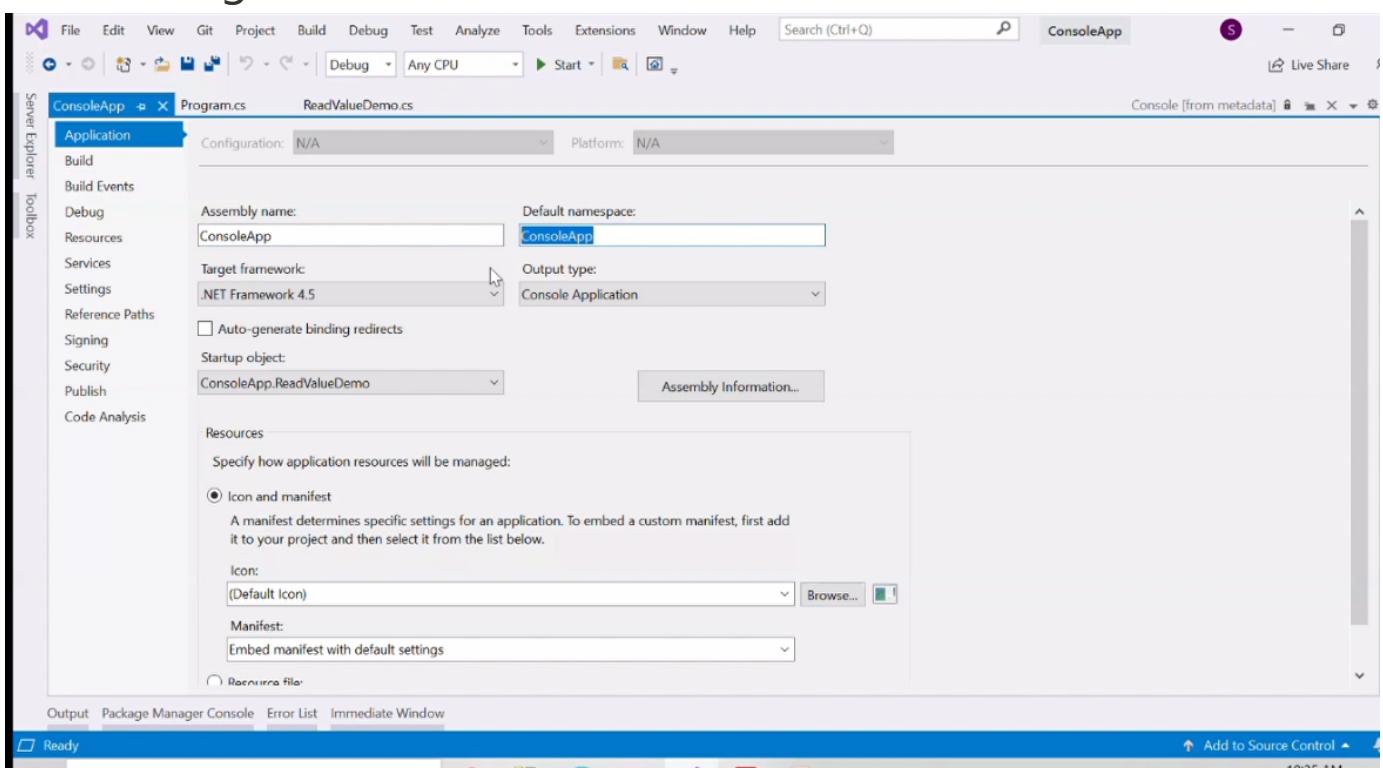
so namespace is not like packages in java where everything will be together all the classes will be part of it , it will be bind together but whereas here this is logical , so logically we put classes together

namespace contains all the logically related classes

```
ConsoleApp Program.cs ReadValueDemo.cs
ConsoleApp
using System;
namespace ConsoleApp
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
            Console.WriteLine("Hi");
            Console.WriteLine("Hello");
            Console.Write("Welcome");
            Console.ReadLine();
        }
    }
}
```

133 % No issues found

now from where did we get the name of our namespace ----So,
usually this will be my project name
Can i change this -----YES



So my namespace has a particular class here named program
and this class has a method called main() in it ,So this becomes a

entry point to this program

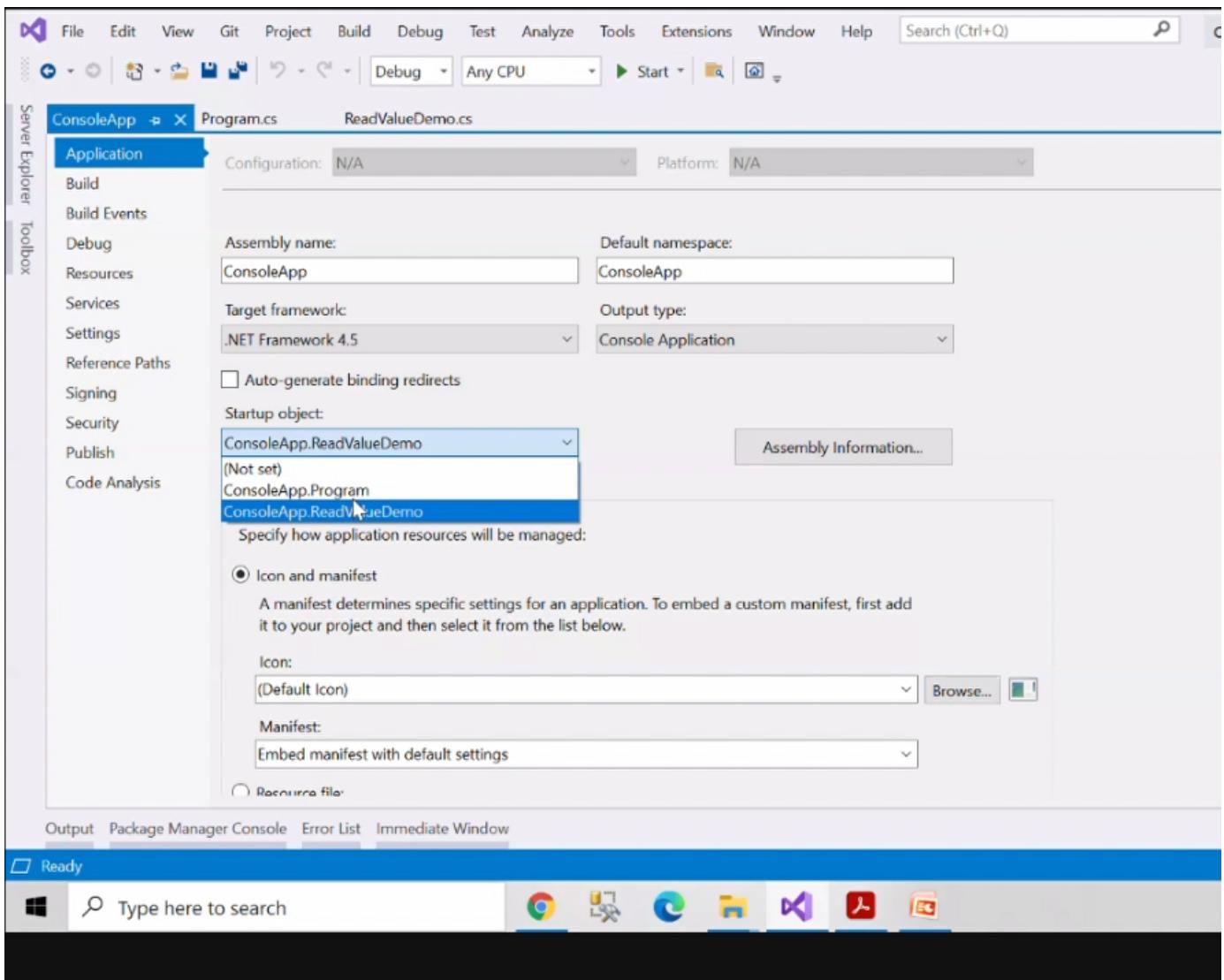
The screenshot shows the Visual Studio IDE with the 'Program.cs' file open. The code is as follows:

```
1  using System;
2
3  namespace ConsoleApp
4  {
5      class Program
6      {
7          static void Main()
8          {
9              Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
10             Console.WriteLine("Hi");
11             Console.WriteLine("Hello");
12             Console.Write("Welcome");
13             Console.ReadLine();
14         }
15     }
16 }
17
```

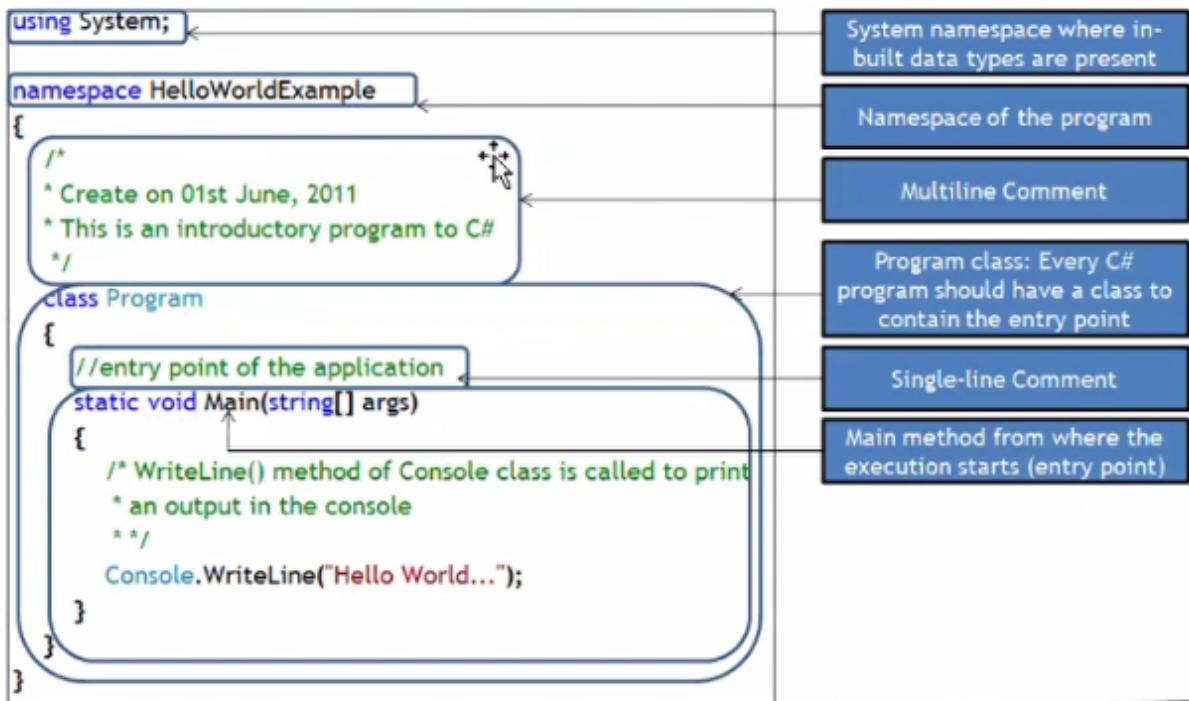
A blue selection bar highlights the entire content of the `Main()` method. The status bar at the bottom indicates "No issues found".

Now , how will i make sure this particular program gets executed
----by selecting that particular file in startup object

Remember---sometime it happens even tho you have
written main there ,sometimes this file/class will not reflect here
in that case what you have to do is
**IF the class is not reflecting there in startup objects -----
close this property window and reopen this window even if it
is not reflecting then make sure that it is static void Main() ---
---M capital , if you write small m then also that class will not
reflect in the startup object, sill not reflecting restart visual
studio**



Structure of C# Program



Writing Code Using Notepad

Different editors can be used to write code, such as Visual Studio Editor and Notepad.

A screenshot of the Visual Studio code editor showing a C# program. A tooltip window is open over the word `Console` in the `Console.WriteLine` call, displaying a list of suggestions including `checked`, `class`, `CLSCompliantAttribute`, `Comparison<T>`, `Console` (which is highlighted), `ConsoleCancelEventArgs`, `ConsoleCancelEventHandler`, `ConsoleColor`, `ConsoleKey`, and `ConsoleKeyInfo`.

```
using System;  
  
namespace NotepadExample  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Con  
        }  
    }  
}
```

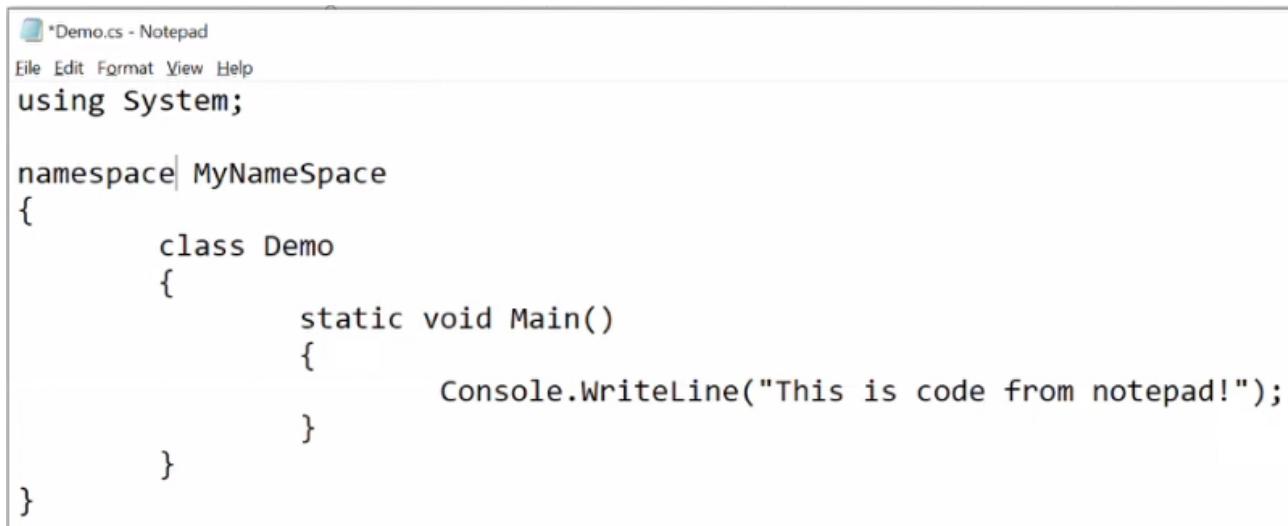
Visual Studio provides Intellisense while writing code

A screenshot of a Windows Notepad window titled "Untitled - Notepad". It contains the same C# code as the Visual Studio example. There is no Intellisense or any other code assistance visible in the Notepad interface.

```
File Edit Format View Help  
using System;  
  
namespace NotepadExample  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Notepad code...");  
        }  
    }  
}
```

Notepad does not provide Intellisense while writing code

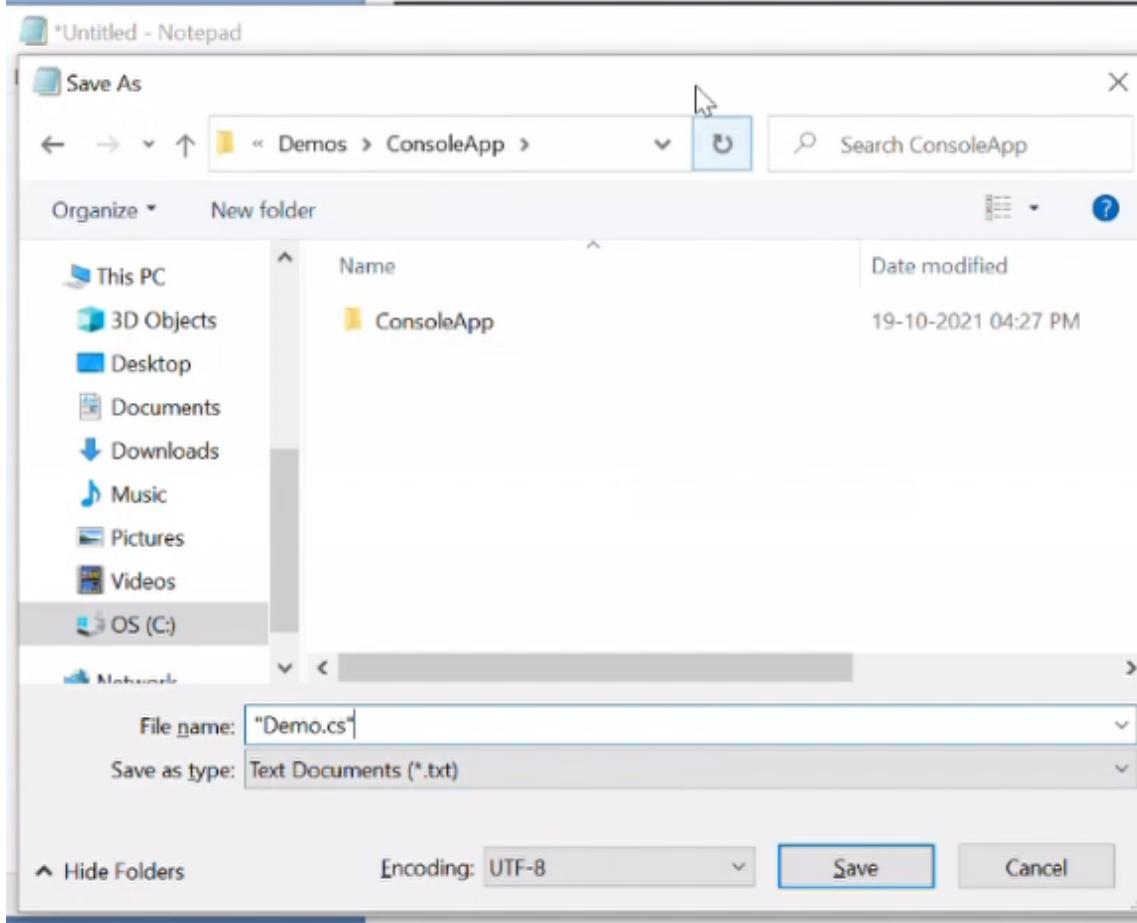
you can write code in NotePad it is possible



```
*Demo.cs - Notepad
File Edit Format View Help
using System;

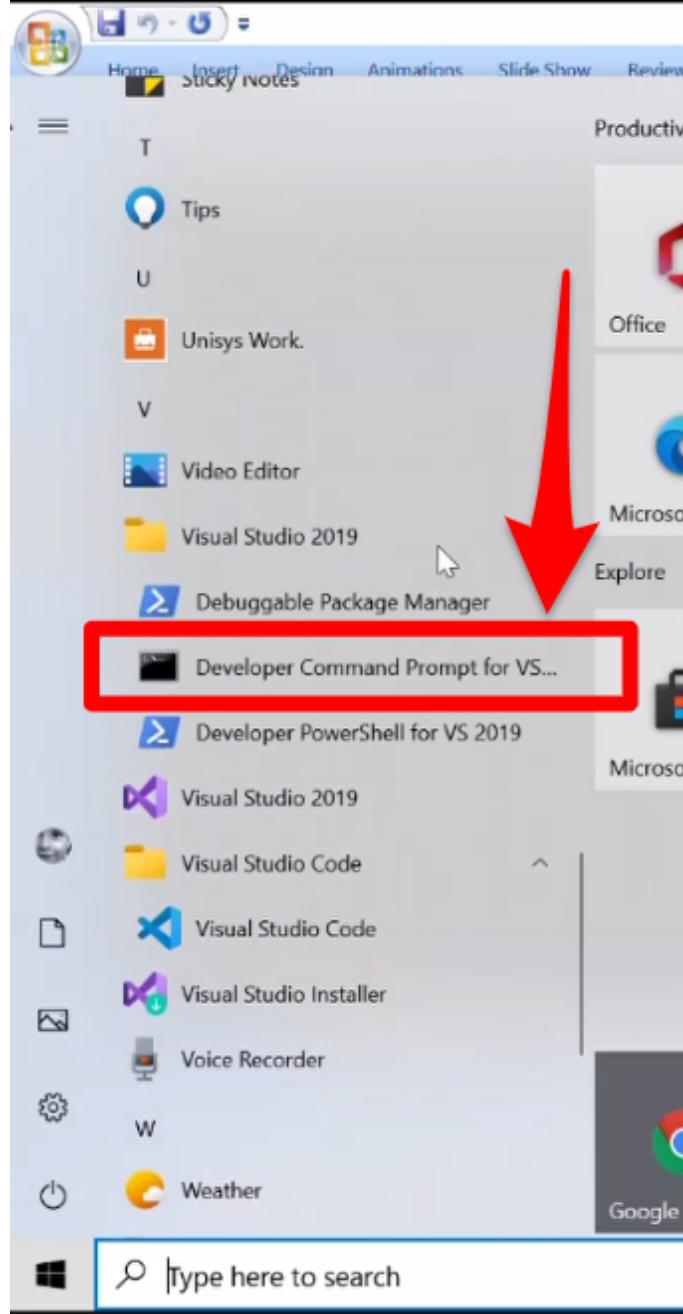
namespace MyNameSpace
{
    class Demo
    {
        static void Main()
        {
            Console.WriteLine("This is code from notepad!");
        }
    }
}
```

now when ever you use not pad make sure that you save file with extension .cs and you can give any name to it it is not like you have to give class name file name same



now you have console window of visual studio which will help you to compile your files and the particular file

How do you compile?----So we'll use the C# compiler

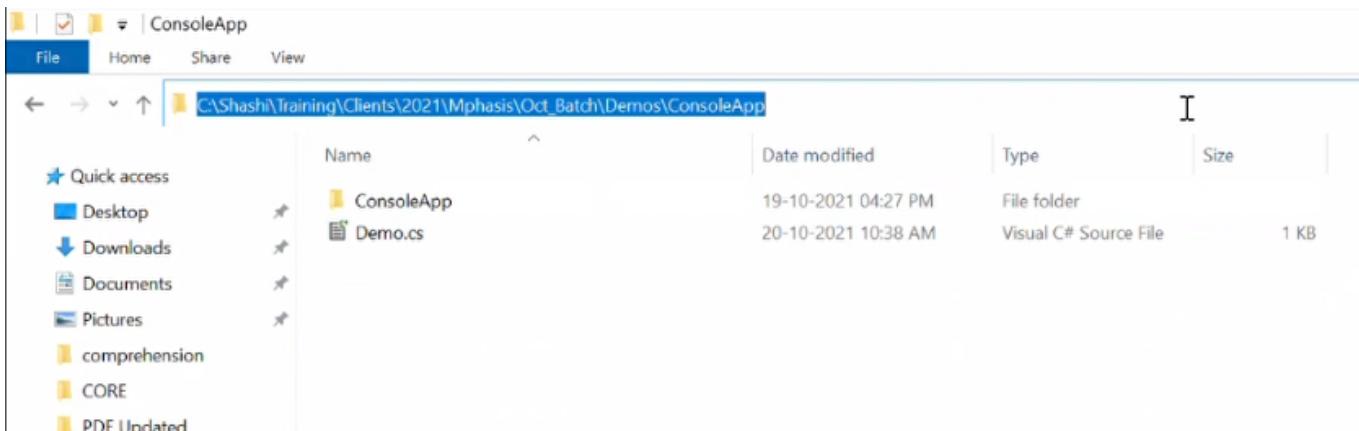


```
Developer Command Prompt for VS 2019 - "C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\Tools\VsDevC... —
*****
** Visual Studio 2019 Developer Command Prompt v16.9.4
** Copyright (c) 2021 Microsoft Corporation
*****
```

```
Developer Command Prompt for VS 2019
*****
** Visual Studio 2019 Developer Command Prompt v16.9.4
** Copyright (c) 2021 Microsoft Corporation
*****

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>cd\

C:\>
```



```
Developer Command Prompt for VS 2019
=====
** Visual Studio 2019 Developer Command Prompt v16.9.4
** Copyright (c) 2021 Microsoft Corporation
=====

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>cd\

C:\>cd C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demo\ConsoleApp

C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demo\ConsoleApp>
```

```
Developer Command Prompt for VS 2019
=====
** Visual Studio 2019 Developer Command Prompt v16.9.4
** Copyright (c) 2021 Microsoft Corporation
=====

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>cd\

C:\>cd C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demo\ConsoleApp

C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demo\ConsoleApp>dir
Volume in drive C is OS
Volume Serial Number is 1456-6BCD

Directory of C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demo\ConsoleApp

20-10-2021  10:38    <DIR>          .
20-10-2021  10:38    <DIR>          ..
19-10-2021  16:27    <DIR>          ConsoleApp
20-10-2021  10:38            145 Demo.cs
                           1 File(s)      145 bytes
                           3 Dir(s)   121,574,232,064 bytes free

C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demo\ConsoleApp>
```

now to compile the program we have

CSC filename

C sharp compiler



```
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>csc Demo.cs
Microsoft (R) Visual C# Compiler version 3.9.0-6.21160.10 (59eedc33)
Copyright (C) Microsoft Corporation. All rights reserved.

Demo.cs(4,2): error CS1513: } expected
Demo.cs(12,1): error CS1022: Type or namespace definition, or end-of-file expected
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>_
```

```
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>csc Demo.cs
Microsoft (R) Visual C# Compiler version 3.9.0-6.21160.10 (59eedc33)
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>_
```

now you can see it got compiled and i got my demo.exe created

```
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>csc Demo.cs
Microsoft (R) Visual C# Compiler version 3.9.0-6.21160.10 (59eedc33)
Copyright (C) Microsoft Corporation. All rights reserved.

C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>dir
Volume in drive C is OS
Volume Serial Number is 1456-6BCD

Directory of C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp

20-10-2021  10:41    <DIR>          .
20-10-2021  10:41    <DIR>          ..
19-10-2021  16:27    <DIR>          ConsoleApp
20-10-2021  10:41            150 Demo.cs
20-10-2021  10:41            3,584 Demo.exe
                  2 File(s)       3,734 bytes
                  3 Dir(s)   121,572,409,344 bytes free

C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>_
```

now you have to write demo and you will get the output , so you can write notepad to write the code but you will not get intellisense here

```
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>Demo
This is code from notepad!

C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Datas\ConsoleApp>_
```

Main method

- Main method is the entry point for a C# Program.
- Main method is a static function defined inside a class or a structure.
- Example:

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

```
namespace MaininStructExample
{
    struct Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hi...");
        }
    }
}
```

Different Signatures of Main method

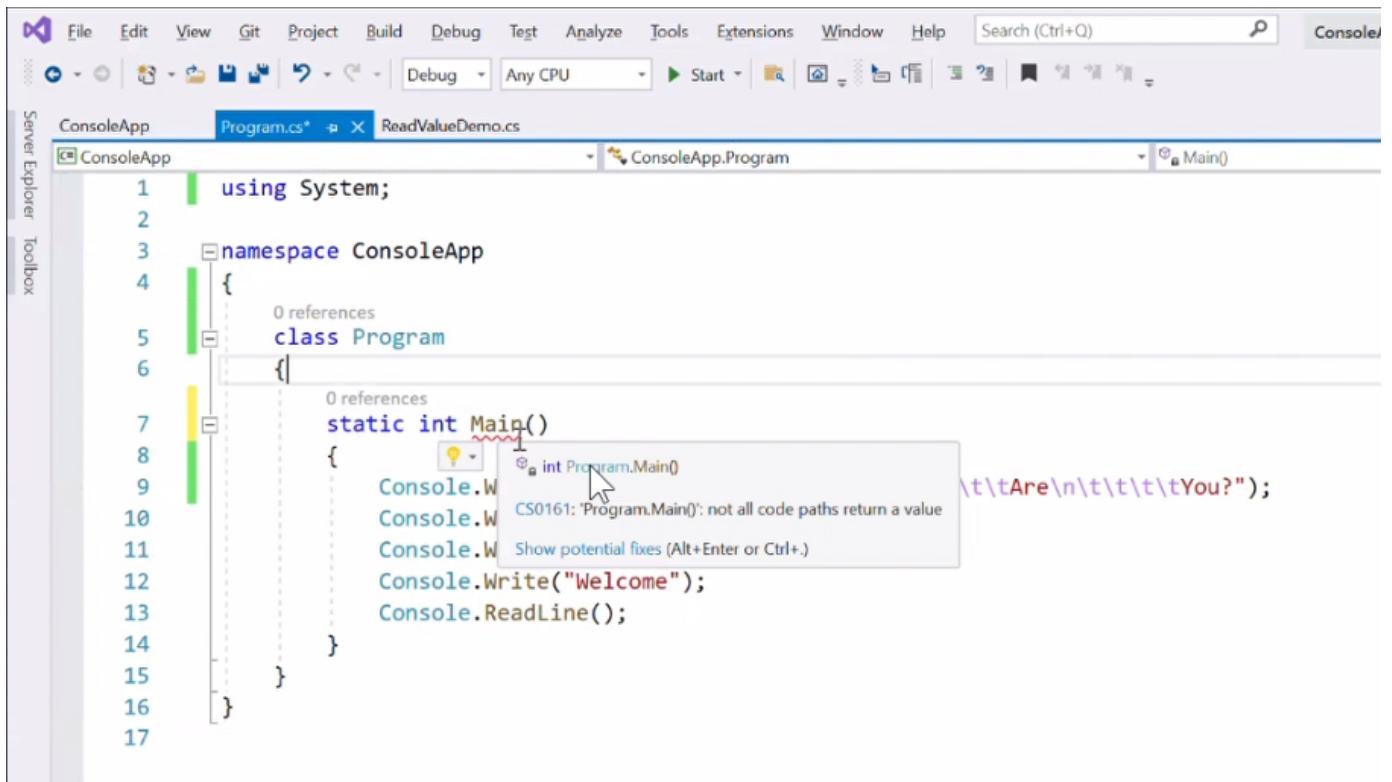
- Different ways to write Main method
 - static void Main()
 - static int Main(string[] args)
 - static void Main(string[] args)
 - static int Main()
- “args” parameter is used to take command-line parameters, i.e. values to be passed into application from outside
- The return value “int” can be used to signify how the application terminates, often used as an indication of an error although this is by no means mandatory

```

1  using System;
2
3  namespace ConsoleApp
4  {
5      class Program
6      {
7          static void Main()
8          {
9              Console.WriteLine("Hi\n\tShashi\n\tHow\n\tAre\n\tYou?");
10             Console.WriteLine("Hi");
11             Console.WriteLine("Hello");
12             Console.Write("Welcome");
13             Console.ReadLine();
14         }
15     }
16 }
17

```

when you write static int Main() so it will ask you that boss what i have to return



A screenshot of the Visual Studio IDE. The code editor shows a C# file named 'Program.cs' with the following code:

```
1 using System;
2
3 namespace ConsoleApp
4 {
5     class Program
6     {
7         static int Main()
8         {
9             Console.WriteLine("Hello");
10            Console.Write("Welcome");
11            Console.ReadLine();
12        }
13    }
14 }
15
16
17 }
```

An inspection tool window is open on the right side, displaying a warning for line 6: "IDE0040 Accessibility modifiers required". It suggests adding accessibility modifiers and provides a preview of the changes:

```
IDE0040 Accessibility modifiers required
Lines 6 to 8
{
    static int Main()
    private static int Main()
}
```

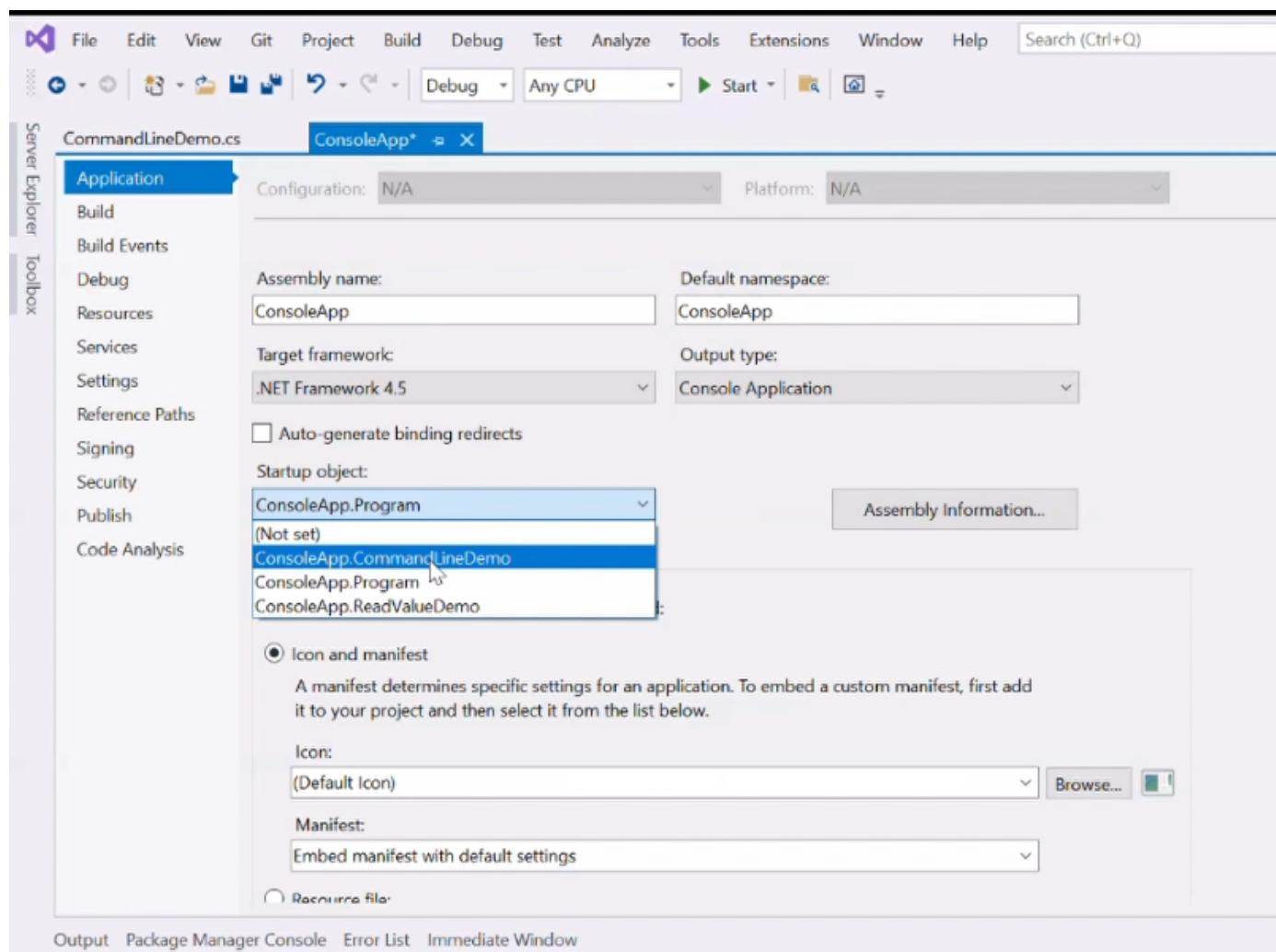
What is Command Line?

A screenshot of the Visual Studio IDE showing a C# file named 'CommandLineDemo.cs' with the following code:

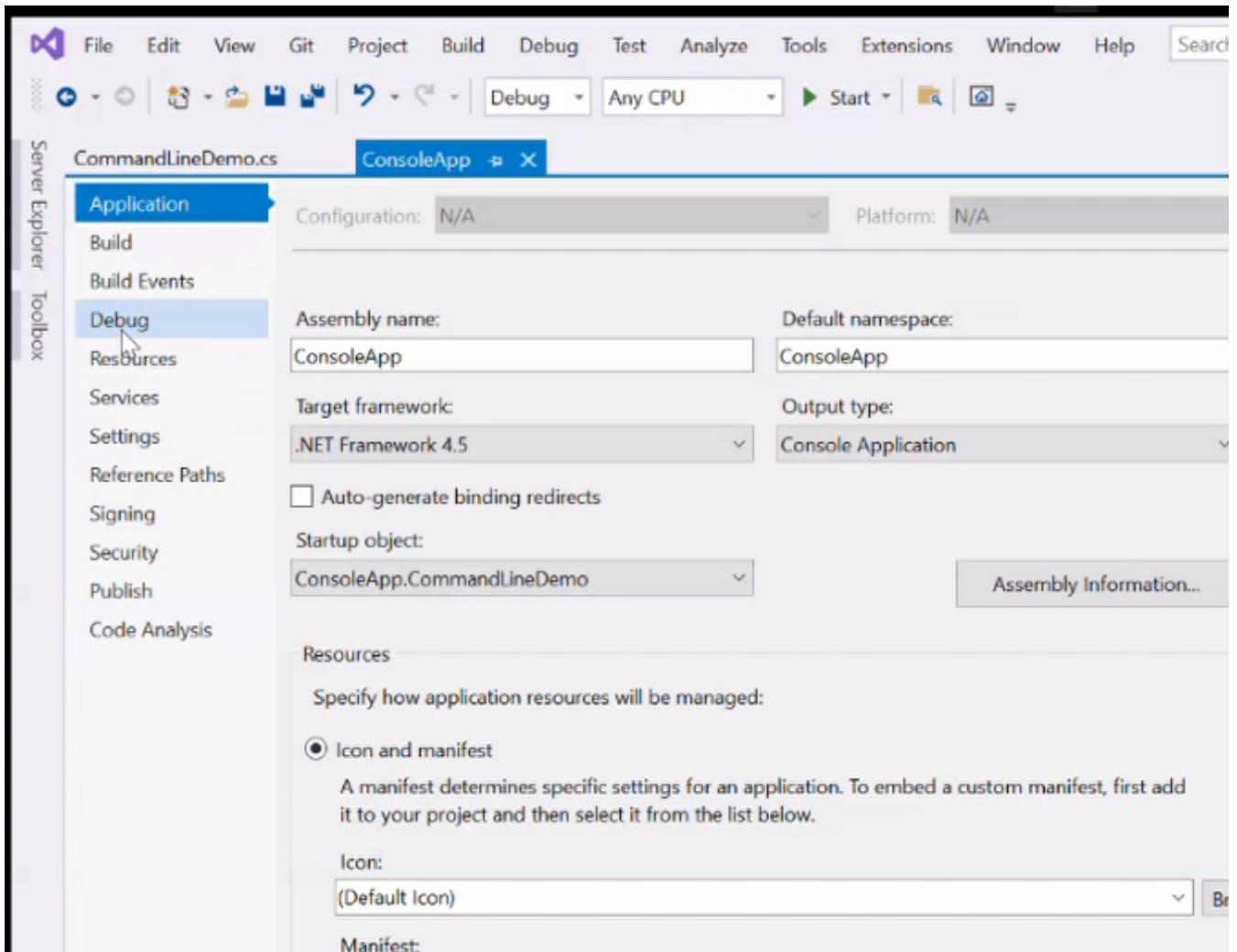
```
1 using System;
2
3 namespace ConsoleApp
4 {
5     class CommandLineDemo
6     {
7         static void Main(string[] args)
8         {
9             if(args.Length > 0)
10             {
11                 foreach (var item in args)
12                 {
13                     Console.WriteLine(item);
14                 }
15             }
16             else
17             {
18                 Console.WriteLine("No args passed");
19             }
20         }
21     }
22 }
```

The status bar at the bottom indicates "133 % No issues found".

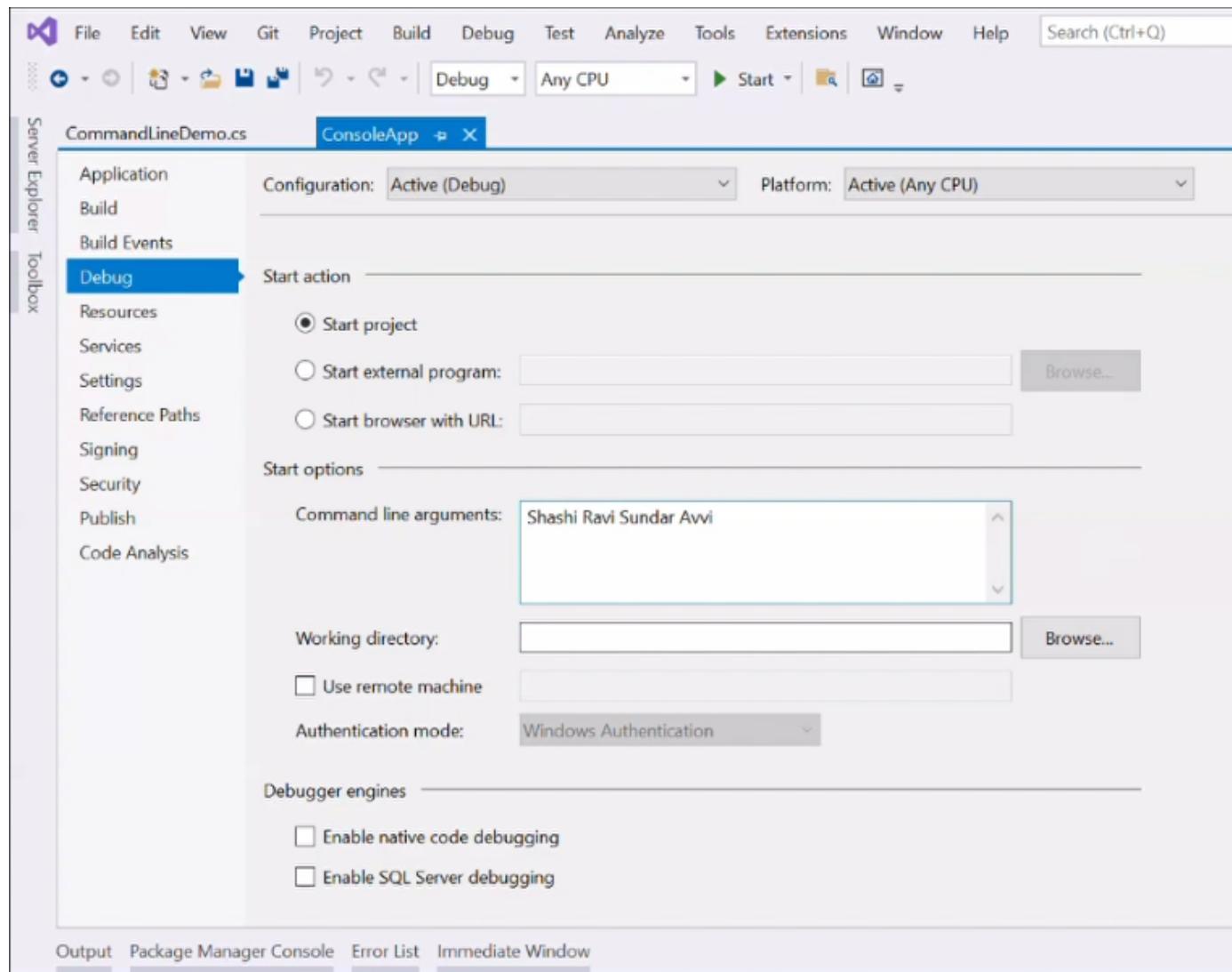
to run set startup object



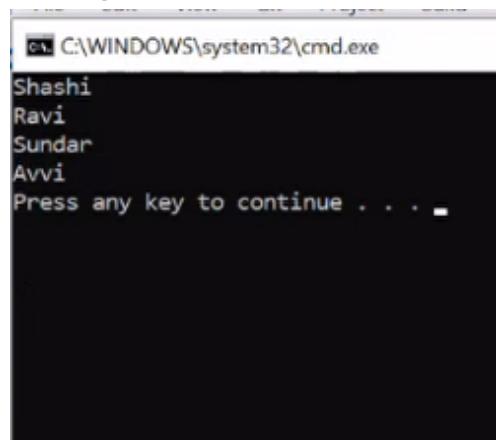
now to pass argument goto---- debug



you can pass command line argument here

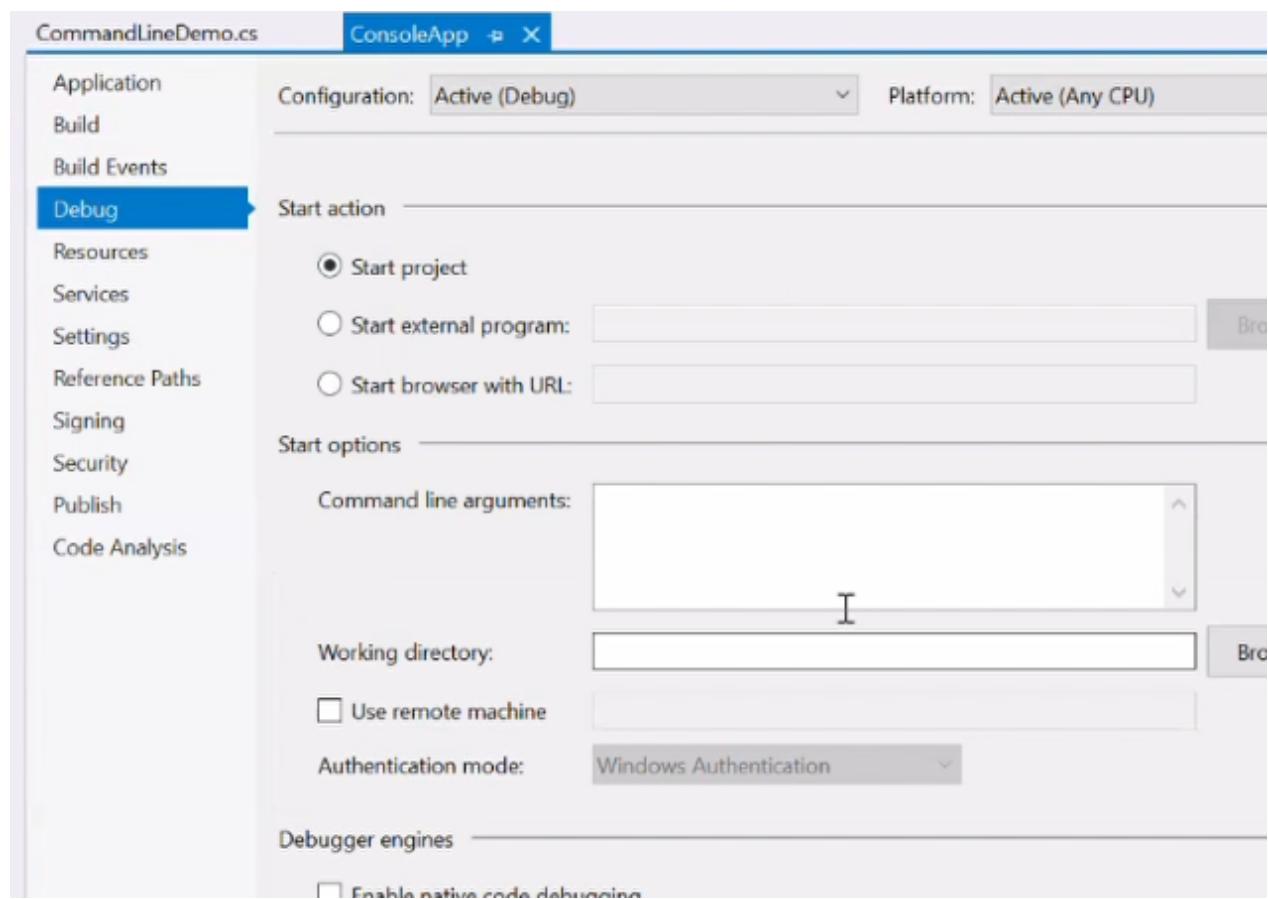


and press **ctrl+f5** to run the code



```
C:\WINDOWS\system32\cmd.exe
Shashi
Ravi
Sundar
Avvi
Press any key to continue . . .
```

if i don't pass any argument i should get message no args passed



see

```
C:\WINDOWS\system32\cmd.exe
No args passed
Press any key to continue . . .
```

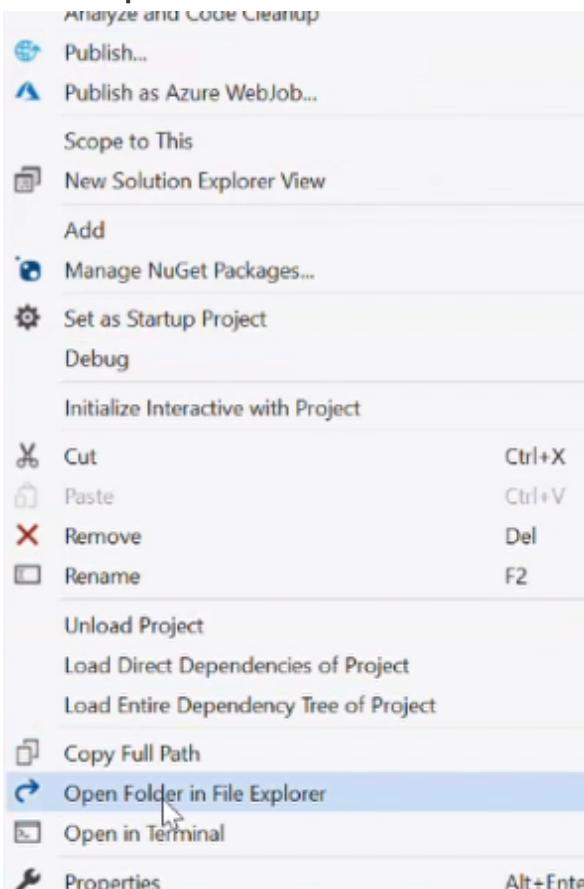
my CommandLineDemo is having a command line argument
meaning when i am running this you can always pass a argument
So we'll go to the exe file

```
7  namespace ConsoleApp
8  {
9      class CommandLineDemo
10     {
11         static void Main(string[] args)
12         {
13             if(args.Length > 0)
14             {
15                 foreach (var item in args)
16                 {
17                     Console.WriteLine(item);
18                 }
19             }
20             else
21             {
22                 Console.WriteLine("No args passed");
23             }
24         }
25     }
26 }
```

No issues found

Solution Explorer Properties Git Changes Notifications

so to open file right click on project and click on open folder in file explorer

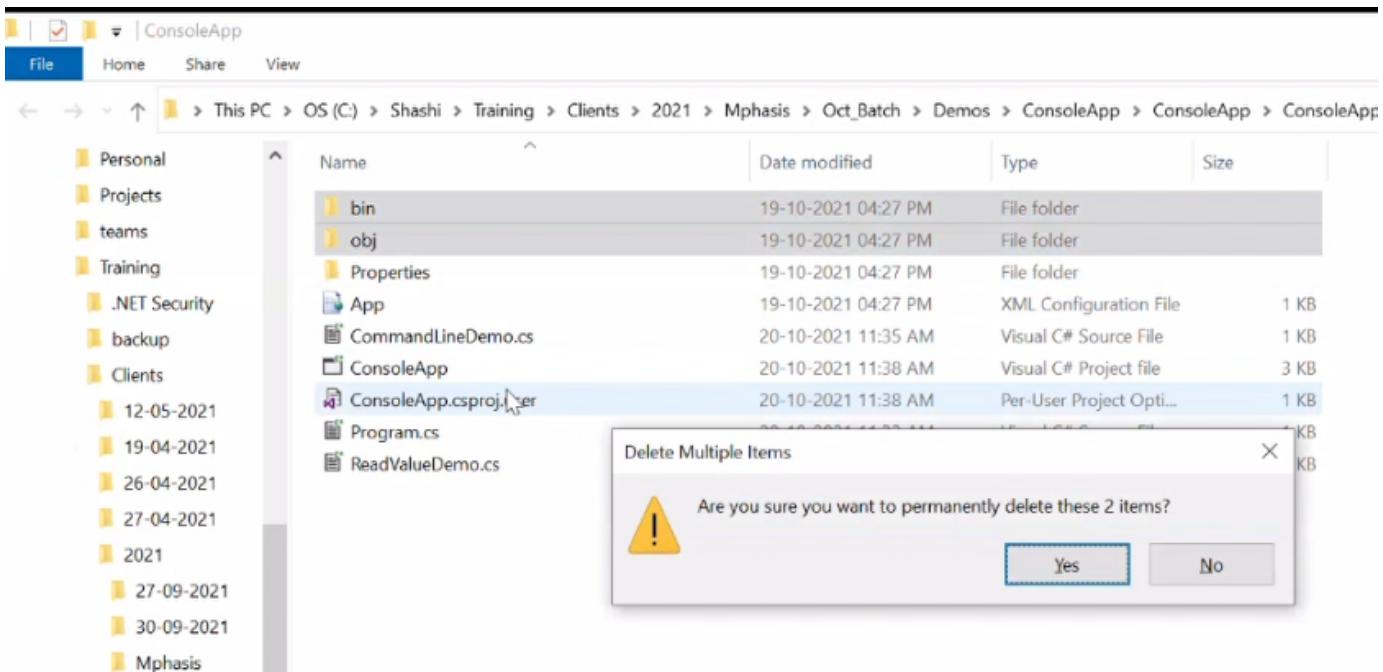


you can see that this is my root folder all this files are present here

Name	Date modified	Type	Size
bin	19-10-2021 04:27 PM	File folder	
obj	19-10-2021 04:27 PM	File folder	
Properties	19-10-2021 04:27 PM	File folder	
App	19-10-2021 04:27 PM	XML Configuration File	1 KB
CommandLineDemo.cs	20-10-2021 11:35 AM	Visual C# Source File	1 KB
ConsoleApp	20-10-2021 11:38 AM	Visual C# Project file	3 KB
ConsoleApp.csproj.user	20-10-2021 11:38 AM	Per-User Project Opti...	1 KB
Program.cs	20-10-2021 11:33 AM	Visual C# Source File	1 KB
ReadValueDemo.cs	19-10-2021 05:46 PM	Visual C# Source File	1 KB

you can see i have bin folder here, this will have all the executable files in it

if you want to share your project to any one you should not follow these 2 folders , its not required



you can delete these folders and then pass this file

Name	Date modified	Type	Size
Properties	19-10-2021 04:27 PM	File folder	
App	19-10-2021 04:27 PM	XML Configuration File	1 KB
CommandLineDemo.cs	20-10-2021 11:35 AM	Visual C# Source File	1 KB
ConsoleApp	20-10-2021 11:38 AM	Visual C# Project file	3 KB
ConsoleApp.csproj.user	20-10-2021 11:38 AM	Per-User Project Opti...	1 KB
Program.cs	20-10-2021 11:33 AM	Visual C# Source File	1 KB
ReadValueDemo.cs	19-10-2021 05:46 PM	Visual C# Source File	1 KB

and i run this again

```
C:\WINDOWS\system32\cmd.exe
No args passed
Press any key to continue . . .
```

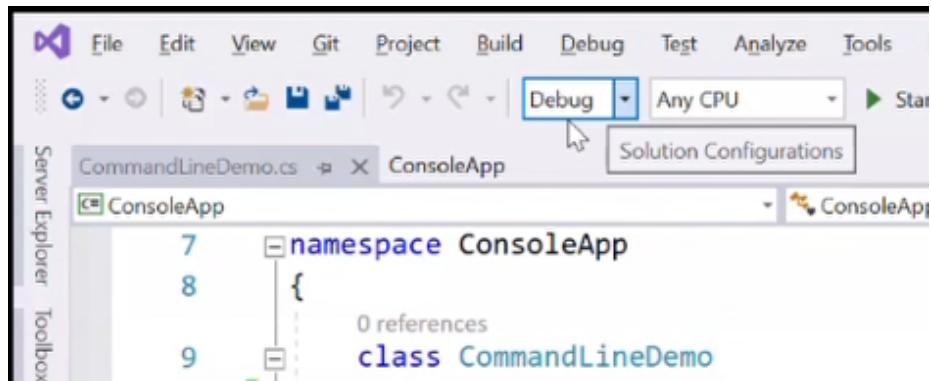
as soon as i run this you can see this 2 folders are again get created -----so this 2 folders have the info associated to the native code means where your application will be running

Name	Date modified	Type	Size
bin	20-10-2021 11:40 AM	File folder	
obj	20-10-2021 11:40 AM	File folder	
Properties	19-10-2021 04:27 PM	File folder	
App	19-10-2021 04:27 PM	XML Configuration File	1 KB
CommandLineDemo.cs	20-10-2021 11:35 AM	Visual C# Source File	1 KB
ConsoleApp	20-10-2021 11:38 AM	Visual C# Project file	3 KB
ConsoleApp.csproj.user	20-10-2021 11:38 AM	Per-User Project Opti...	1 KB
Program.cs	20-10-2021 11:33 AM	Visual C# Source File	1 KB
ReadValueDemo.cs	19-10-2021 05:46 PM	Visual C# Source File	1 KB

so bin has debug folder

Name	Date modified	Type	Size
Debug	20-10-2021 11:40 AM	File folder	

you can see that i am working in debug environment here ,so i get debug folder

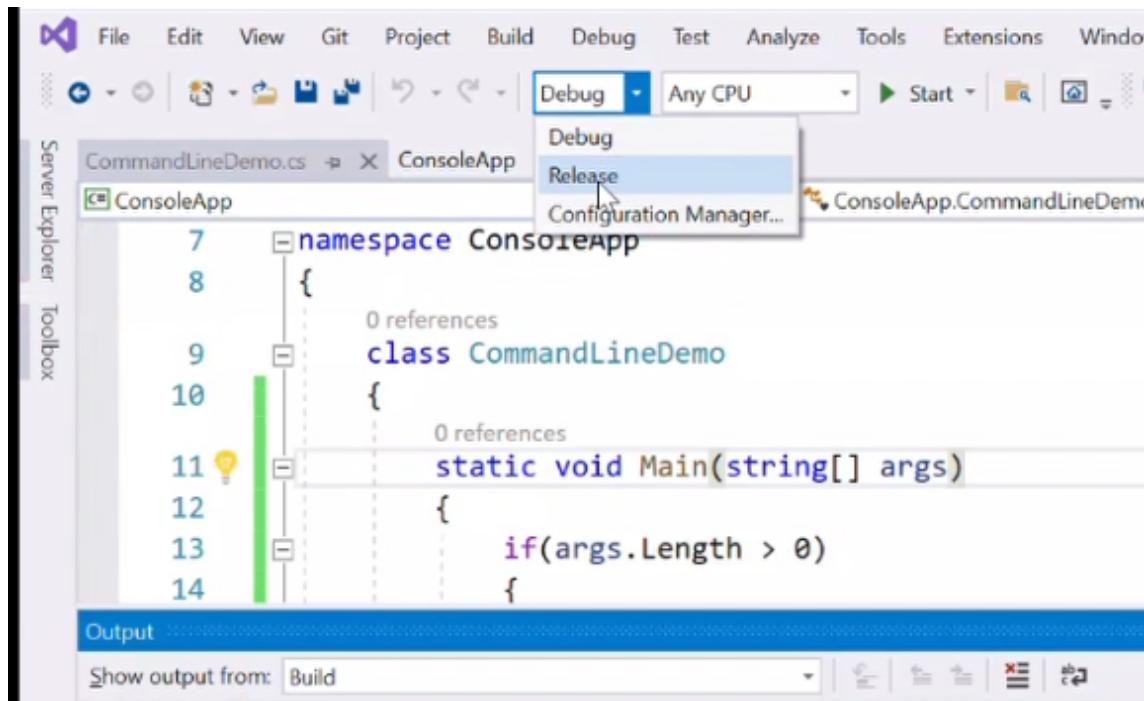


so if go to the release environment here then i ll get the release folder here

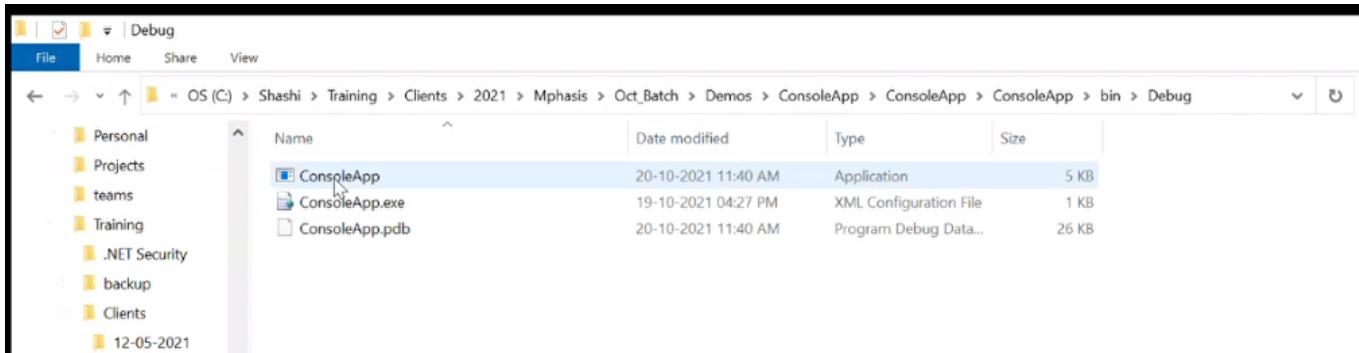
What is a difference between debug and release ?

release is for the production, so it will run in the production environment

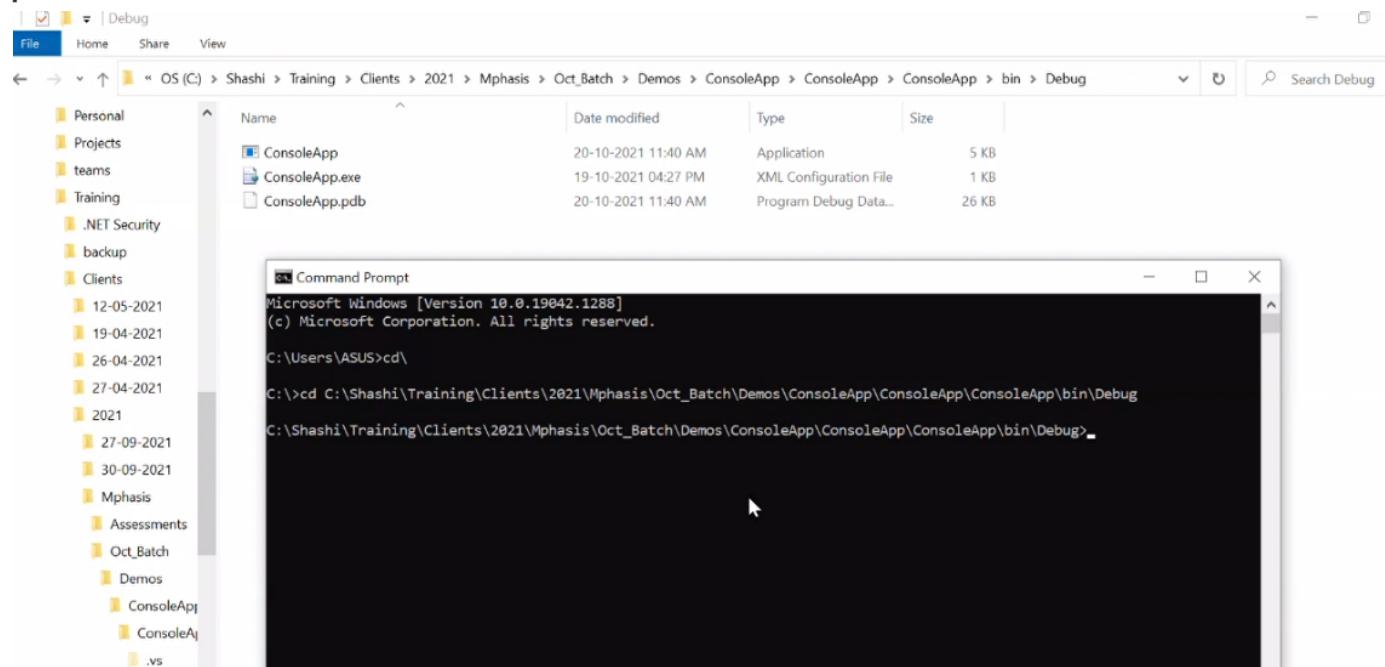
and debug is for development



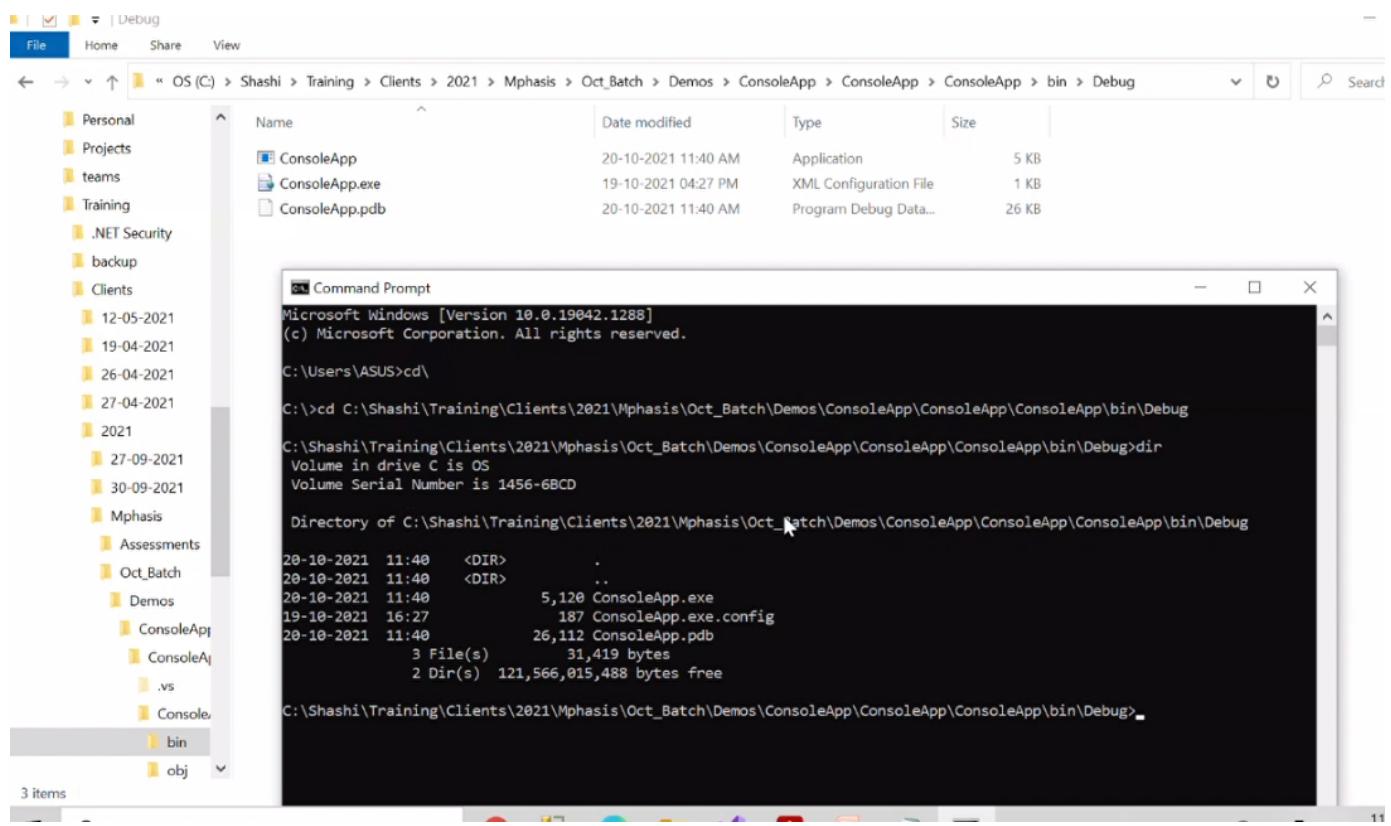
now you can see in the debug i have console app



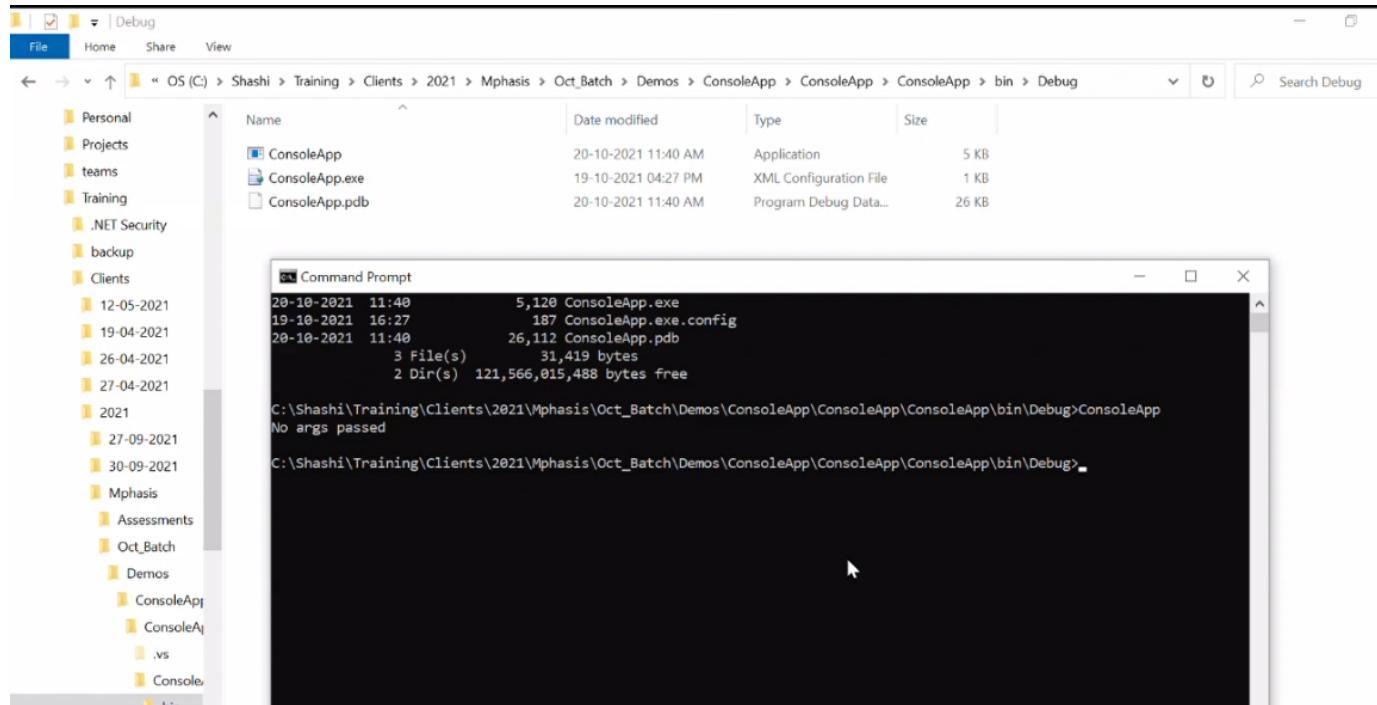
so copy the path and in cmd put cd\ enter and then paste the path



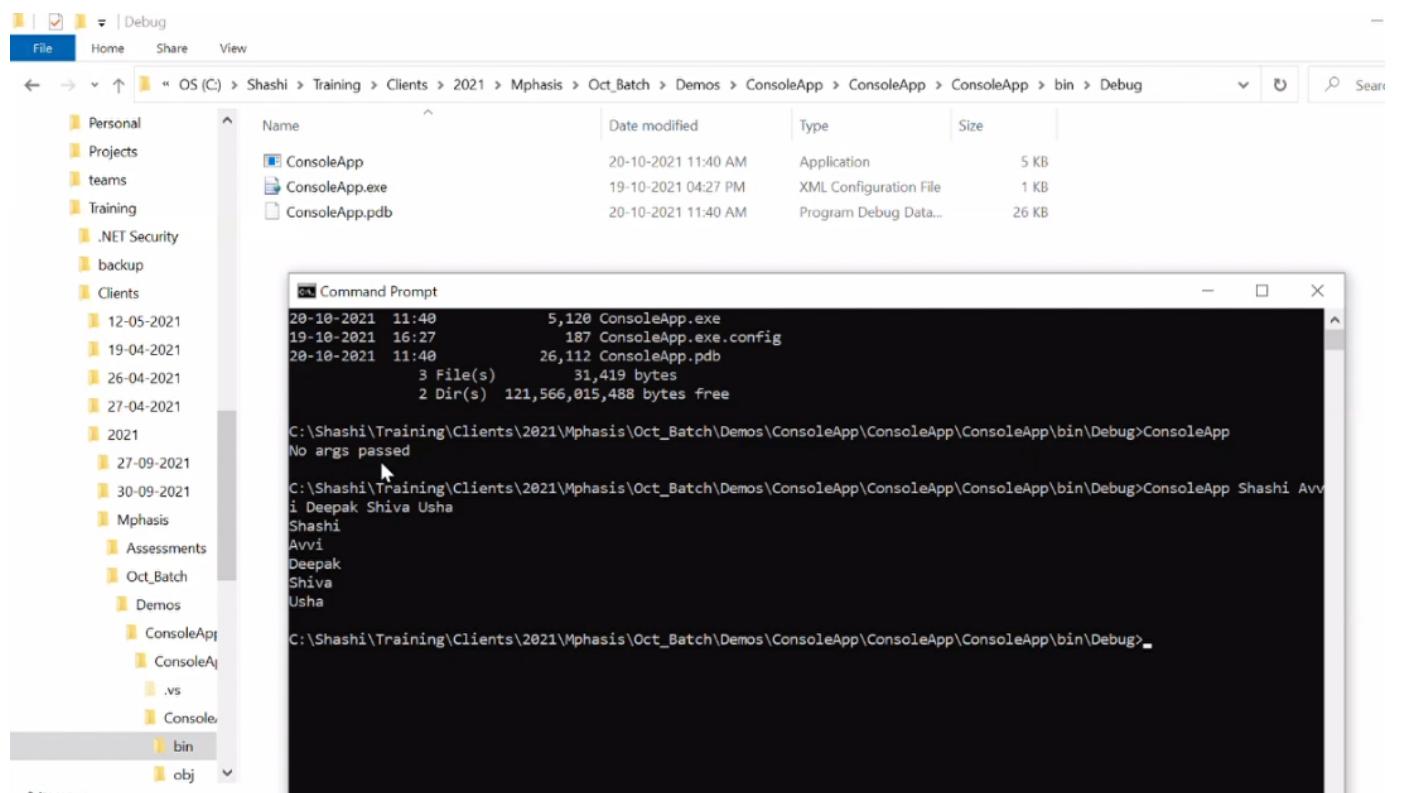
now you can see we are in debug folder ,so i'll say dir



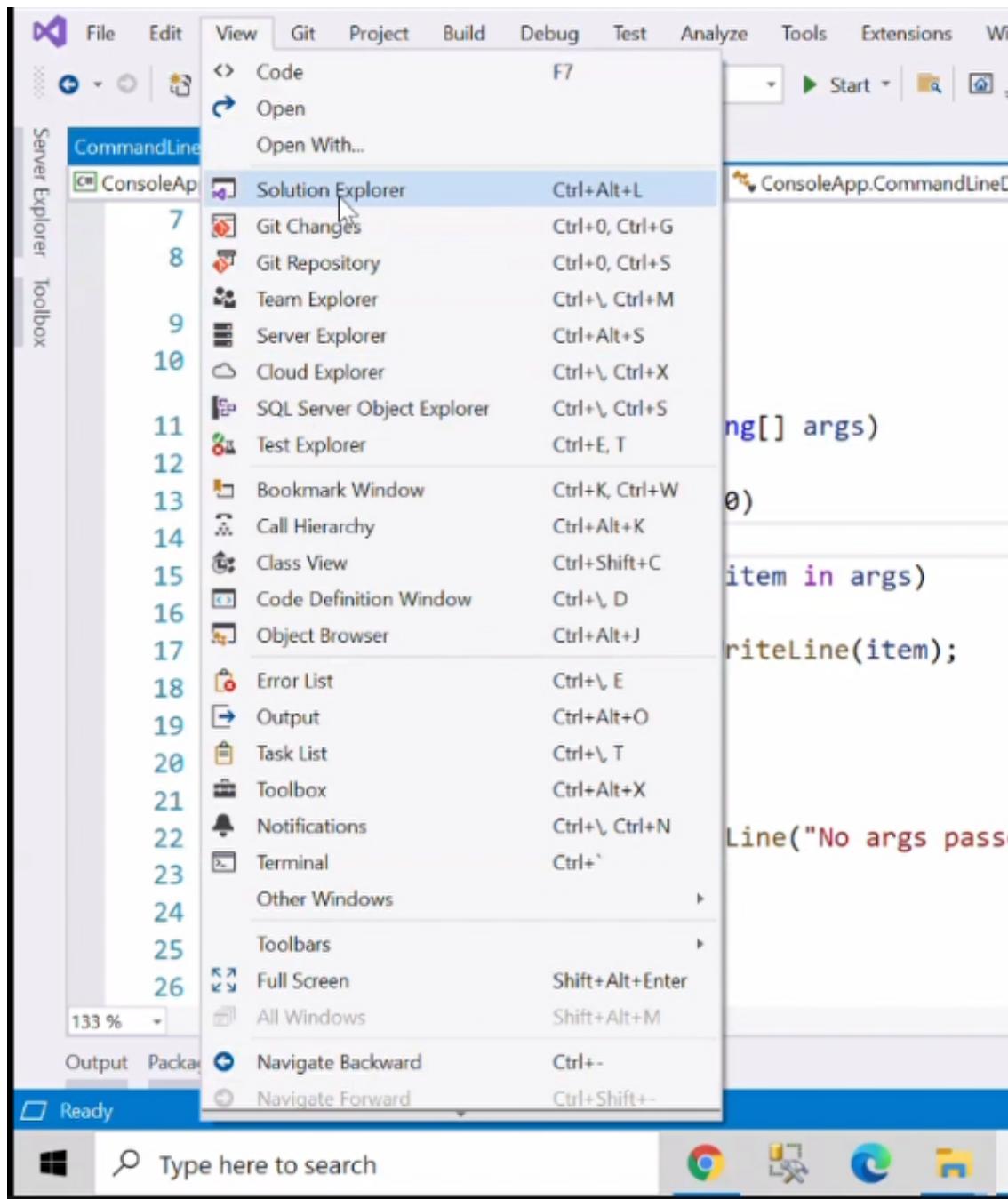
I'll type consoleapp.exe or else you can type console app that is still ok and press enter



You can see it says no argument passed, let's pass some argument after consoleapp I am giving space and passing the arguments



when you open visual studio this things will always be there like solution explorer, property windowif it is not there you can goto view



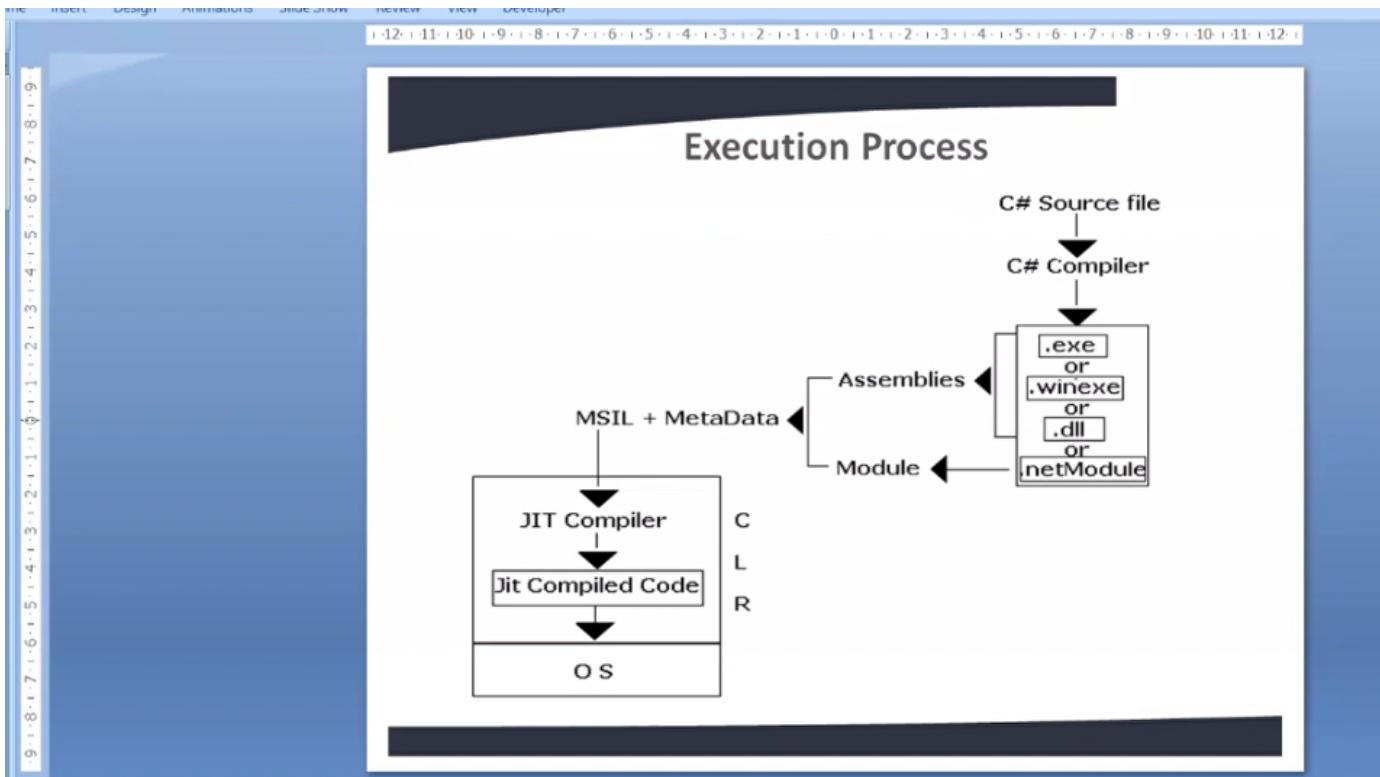
and here you can get them

Basic Input/Output - The Console Class

- Provides access to the standard input, standard output, and standard error streams
- Only meaningful for console applications
 - Standard input – keyboard
 - Standard output – screen
 - Standard error – screen
- All streams may be redirected

Write and WriteLine, Read and ReadLine Methods

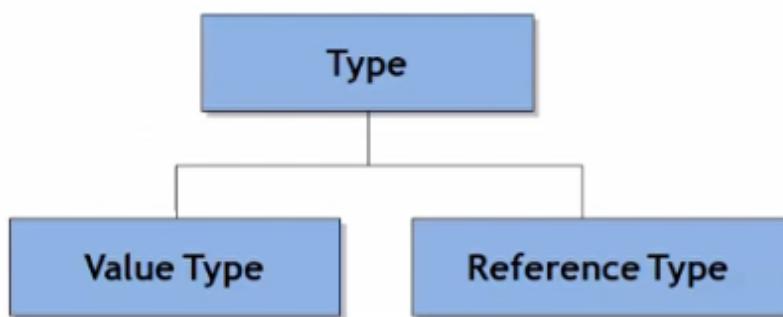
- Console.Write() and Console.WriteLine() display information on the console screen
 - Write method displays output without a carriage return/line feed.
 - WriteLine outputs a line feed/carriage return
 - Example: `Console.WriteLine("What is your name? ");`
- Console.Read() and Console.ReadLine() read user input
 - Read reads the next character **read only one character at a time**
 - ReadLine reads the entire input line **read multiple character at a time**
 - Example: `string name = Console.ReadLine();`



C# Compiler (or any .NET Language compiler) can generate different types of files: **.exe / .winexe / .dll / .netModule.**

Overview of Data Type System

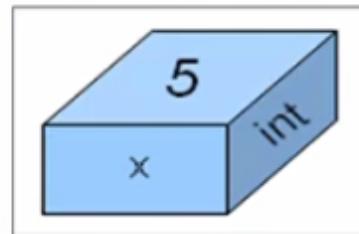
- There are two different categories of data types. They are value and reference types



Value Type Variable

- In a variable that holds a **value type**, the data itself is directly contained within the memory allotted to the variable
- Example:
 - `int x = 5;`
 - The above code declares an 32-bit signed integer variable, called x, initialized with a value of 5. The following figure represents the corresponding variable diagram:

The following figure represents the corresponding variable diagram:



Example of value type

```
File Edit View Insert Build Debug Tools Analyze Tools Window Help Search (Ctrl+Q) ConsoleApp  
ConsoleApp.cs* ConsoleApp  
ConsoleApp PassbyValueDemo.cs Main()  
Server Explorer Toolbox Solution Explorer  
ConsoleApp  
7  namespace ConsoleApp  
8  {  
9  }  
0 references  
10 class PassbyValueDemo  
11 {  
12 }  
0 references  
13 static void Main()  
14 {  
15     int no1 = 10, no2 = 20;  
16     Console.WriteLine("Before Swapping");  
17     Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);  
18     Swap(no1, no2);  
19 }  
20 }  
CS0103: The name 'Swap' does not exist in the current context  
Show potential fixes (Alt+Enter or Ctrl+.)  
when ever you see this line your brain should say  
areey yaar we have'nt created  
instance of it  
Output Package Manager Console Error List Immediate Window  
Ready Type here to search
```

But if you are directly accessing this method so you have to write static

```
PassbyValueDemo.cs*  X ConsoleApp*
ConsoleApp
7   namespace ConsoleApp
8   {
9       class PassbyValueDemo
10      {
11          static void Main()
12          {
13              int no1 = 10, no2 = 20;
14              Console.WriteLine("Before Swapping");
15              Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
16              Swap(no1, no2);
17          }
18
19          static void Swap(int no1, int no2)
20          {
21          }
22      }
23  }
24
25
```

now instead of writing this

```
PassbyValueDemo.cs*  X ConsoleApp*
ConsoleApp
7   namespace ConsoleApp
8   {
9       class PassbyValueDemo
10      {
11          static void Main()
12          {
13              int no1 = 10, no2 = 20;
14              Console.WriteLine("Before Swapping");
15              Console.WriteLine("No1 = " + no1 + "\tNo2 = ");
16              Swap(no1, no2);
17          }
18
19          static void Swap(int no1, int no2)
20          {
21          }
22      }
23  }
24
25
```

CS0103: The name 'Swap' does not exist in the current context
Show potential fixes (Alt+Enter or Ctrl+.)

you can see a small bulb here
when you click on the arrow jo iske side me hai

A screenshot of Microsoft Visual Studio showing a C# code editor. The code in the Main() method contains a call to 'Swap(no1, no2);'. A tooltip appears over the 'Swap' method name, containing the text 'Generate method 'PassbyValueDemo.Swap'' with a red arrow pointing to it. Another option in the tooltip says 'Introduce local for 'Swap(no1, no2)''. Below the tooltip, an error message is displayed: 'CS0103 The name 'Swap' does not exist in the current context'. The code editor shows lines 18 to 20 of the swap method definition:

```
private static void Swap(int no1, int no2)
{
    throw new NotImplementedException();
}
```

this is exactly what is in my mind which i want to write so click on this and you get a method created for you

if you are soo lazy to do this also so just press **ctrl + .**
and press enter and you will get this method automatically
created for you

```
ConsoleApp
    7  namespace ConsoleApp
    8  {
    9      0 references
   10     class PassbyValueDemo
   11     {
   12         0 references
   13         static void Main()
   14         {
   15             int no1 = 10, no2 = 20;
   16             Console.WriteLine("Before Swapping");
   17             Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
   18             Swap(no1, no2);
   19         }
   20
   21         private static void Swap(int no1, int no2)
   22         {
   23             throw new NotImplementedException();
   24         }
   25     }
```

```
No issues found | Search (Ctrl+Q) | L: 16 C: 8 Col: 17 TABS
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PassbyValueDemo.cs  ConsoleApp*
ConsoleApp
    9  class PassbyValueDemo
    10 {
    11     0 references
    12     static void Main()
    13     {
    14         int no1 = 10, no2 = 20;
    15         Console.WriteLine("Before Swapping");
    16         Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
    17         Swap(no1, no2);
    18     }
    19
    20     private static void Swap(int no1, int no2)
    21     {
    22         no1 = no1 + no2;
    23         no2 = no1 - no2;
    24         no1 = no1 - no2;
    25         Console.WriteLine("After Swapping");
    26         Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
    27     }
    28 }
```

I'll set this as startup program and run

The screenshot shows a Microsoft Visual Studio interface. The top part is a terminal window titled 'cmd.exe' showing the output of a program. The bottom part is the code editor for 'ConsoleApp.cs'.

Output Window:

```
C:\WINDOWS\system32\cmd.exe
Before Swapping
No1 = 10      No2 = 20
After Swapping
No1 = 20      No2 = 10
Press any key to continue . . .
```

Code Editor (ConsoleApp.cs):

```
class PassbyValueDemo
{
    static void Main()
    {
        int no1 = 10, no2 = 20;
        Console.WriteLine("Before Swapping");
        Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
        Swap(no1, no2);
        Console.WriteLine("After Swapping");
        Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
    }

    private static void Swap(int no1, int no2)
    {
        no1 = no1 + no2;
        no2 = no1 - no2;
        no1 = no1 - no2;
        Console.WriteLine("After Swapping");
        Console.WriteLine("No1 = " + no1 + "\tNo2 = " + no2);
    }
}
```

The code defines a class 'PassbyValueDemo' with a 'Main' method. It initializes two integers 'no1' and 'no2' to 10 and 20 respectively. It prints their initial values, calls a 'Swap' method, and then prints their values again after the swap. The 'Swap' method takes two integers and swaps them by performing arithmetic operations on their values.

can you write the output for this program

```
leApp          ConsoleApp.Pa
9   class PassbyValueDemo
10  Select C:\WINDOWS\system32\cmd.exe
11  Before Swapping
12  No1 = 10      No2 = 20
13  After Swapping
14  No1 = 20      No2 = 10
15  After Swapping
16  No1 = 10      No2 = 20
17  Press any key to continue . . .
```

you can see again i am getting 10 and 20 because swapping is done in swap method not in main

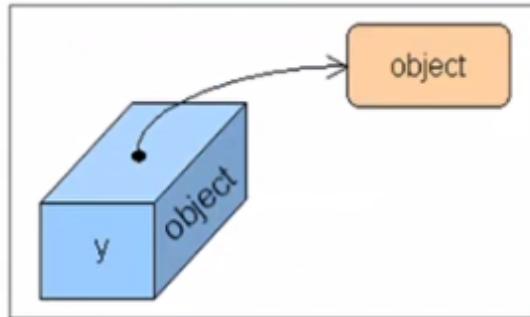
SO , this is pass by value , we are not passing the references

here -----so this are value type

Reference Type

- On the other hand, a variable that holds a **reference type** contains the address of an object stored in the heap.
- Example:
 - *object y = new object();*
 - The above code declares a variable called y of type object which gets initialized, thanks to the new operator, so that it refers to a new heap allocated object instance (object is the base class of all C# types, but more of this latter).

The following figure represents the corresponding variable diagram:



next we have reference type

What is reference type , where you can pass the particular references

here we are going with little advance topic so pay attention

I want to store the student information , how do you store the student information

using class -----why should i store the information using class --
-----because it is easy to handle

so instead of me putting it in a class i'll create a structure

```
PassbyValueDemo2.cs*  X  ConsoleApp*
```

```
ConsoleApp
```

```
ConsoleApp.PassbyValueDemo2
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      struct Student
10     {
11         public int id;
12         public string name;
13     }
14
15
16     class PassbyValueDemo2
17     {
18         static void Main()
19     }
}
```

Output Package Manager Console Error List Immediate Window

Ready Type here to search

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Menu Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q).
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Shows "PassbyValueDemo2.cs*" and "ConsoleApp*".
- Toolbox:** Standard development tools.
- Code Editor:** Displays the following C# code:

```
PassbyValueDemo2.cs*  X  ConsoleApp*
ConsoleApp
13 }
14
15
16 class PassbyValueDemo2
17 {
18     static void Main()
19     {
20         Student student1 = new Student();
21         student1.
22     }
23 }
24 }
```
- IntelliSense Pop-up:** A tooltip is open over the `student1.` part of the code, listing the following properties and methods:
 - Equals
 - GetHashCode
 - GetType
 - id
 - name
 - ToString

The "Equals" method is highlighted. The tooltip also contains the following text:

bool ValueType.Equals(object obj)
Indicates whether this instance and a specified object are equal.
Note: Tab twice to insert the 'Equals' snippet.

Text Overlay: A red arrow points from the text "can you see a small brick symbol here this symbol is for your field" to the small blue square icon next to the "id" entry in the IntelliSense list.

there is lot of difference between the field and the property we'll see that

so i'll pass value in id and name
and class displaystudent method

```
L3 }  
L4  
L5  
L6 0 references  
L7 class PassbyValueDemo2  
L8 {  
L9 0 references  
L10 static void Main()  
L11 {  
L12     Student student1 = new Student();  
L13     student1.id = 1001;  
L14     student1.name = "Shashi";  
L15     DispalyStudent(student1);  
L16 }  
L17  
L18 1 reference  
L19 private static void DispalyStudent(Student student)  
L20 {  
L21     Console.WriteLine("ID = " + student.id);  
L22     Console.WriteLine("Name = " + student.name);  
L23 }  
L24 }  
L25  
No issues found | 🔍 | ⏪ ⏴ | ⏵ | ⏹ | Ln: 29  
Package Manager Console Error List Immediate Window
```

output

```
ID = 1001  
Name = Shashi  
Press any key to continue . . .
```

now i'll create one more instance and i am assigning what ever there is in student 1 to the student 2

now in the student 2 i'll change the id then again call dispalystudent by passing student2 as parameter

```
0 references
class PassbyValueDemo2
{
    0 references
    static void Main()
    {
        Student student1 = new Student();
        student1.id = 1001;
        student1.name = "Shashi";
        DispalyStudent(student1);
        Student student2 = student1;
        student2.id = 1002;
        DispalyStudent(student2);
        DispalyStudent(student1);
    }
}

3 references
private static void DispalyStudent(Student student)
{
    Console.WriteLine("ID = " + student.id);
    Console.WriteLine("Name = " + student.name);
}
```

output

```
C:\WINDOWS\system32\cmd.exe
ID = 1001
Name = Shashi
ID = 1002
Name = Shashi
ID = 1001
Name = Shashi
Press any key to continue . . . -
```

now from struct i'll change this to class now what will be the output

```
leDemo2.cs  X  ConsoleApp
ConsoleApp
ConsoleApp.PassbyValueDemo2
5     using System.Threading.Tasks;
6
7     namespace ConsoleApp
8     {
9         class Student
10        {
11            public int id;
12            public string name;
13        }
14
15    }
16
17    class PassbyValueDemo2
18    {
19        static void Main()
20        {
21            Student student1 = new Student();
22            student1.id = 1001;
23            student1.name = "Shashi";
24            DispalyStudent(student1);
25        }
26    }
27
28    void DispalyStudent(Student s)
29    {
30        Console.WriteLine("ID = " + s.id);
31        Console.WriteLine("Name = " + s.name);
32    }
33
34    static void Main(string[] args)
35    {
36        PassbyValueDemo2 pbd = new PassbyValueDemo2();
37        pbd.Main();
38    }
39
40    // Output:
41    // ID = 1001
42    // Name = Shashi
43    // ID = 1002
44    // Name = Shashi
45    // Press any key to continue . . .
46
47    // Output:
48    // ID = 1001
49    // Name = Shashi
50    // ID = 1002
51    // Name = Shashi
52    // Press any key to continue . . .
53
54    // Output:
55    // ID = 1001
56    // Name = Shashi
57    // ID = 1002
58    // Name = Shashi
59    // Press any key to continue . . .
60
61    // Output:
62    // ID = 1001
63    // Name = Shashi
64    // ID = 1002
65    // Name = Shashi
66    // Press any key to continue . . .
67
68    // Output:
69    // ID = 1001
70    // Name = Shashi
71    // ID = 1002
72    // Name = Shashi
73    // Press any key to continue . . .
74
75    // Output:
76    // ID = 1001
77    // Name = Shashi
78    // ID = 1002
79    // Name = Shashi
80    // Press any key to continue . . .
81
82    // Output:
83    // ID = 1001
84    // Name = Shashi
85    // ID = 1002
86    // Name = Shashi
87    // Press any key to continue . . .
88
89    // Output:
90    // ID = 1001
91    // Name = Shashi
92    // ID = 1002
93    // Name = Shashi
94    // Press any key to continue . . .
95
96    // Output:
97    // ID = 1001
98    // Name = Shashi
99    // ID = 1002
100   // Name = Shashi
101
102   // Output:
103   // ID = 1001
104   // Name = Shashi
105   // ID = 1002
106   // Name = Shashi
107
108   // Output:
109   // ID = 1001
110   // Name = Shashi
111   // ID = 1002
112   // Name = Shashi
113
114   // Output:
115   // ID = 1001
116   // Name = Shashi
117   // ID = 1002
118   // Name = Shashi
119
120   // Output:
121   // ID = 1001
122   // Name = Shashi
123   // ID = 1002
124   // Name = Shashi
125
126   // Output:
127   // ID = 1001
128   // Name = Shashi
129   // ID = 1002
130   // Name = Shashi
131
132   // Output:
133   // ID = 1001
134   // Name = Shashi
135   // ID = 1002
136   // Name = Shashi
137
138   // Output:
139   // ID = 1001
140   // Name = Shashi
141   // ID = 1002
142   // Name = Shashi
143
144   // Output:
145   // ID = 1001
146   // Name = Shashi
147   // ID = 1002
148   // Name = Shashi
149
150   // Output:
151   // ID = 1001
152   // Name = Shashi
153   // ID = 1002
154   // Name = Shashi
155
156   // Output:
157   // ID = 1001
158   // Name = Shashi
159   // ID = 1002
160   // Name = Shashi
161
162   // Output:
163   // ID = 1001
164   // Name = Shashi
165   // ID = 1002
166   // Name = Shashi
167
168   // Output:
169   // ID = 1001
170   // Name = Shashi
171   // ID = 1002
172   // Name = Shashi
173
174   // Output:
175   // ID = 1001
176   // Name = Shashi
177   // ID = 1002
178   // Name = Shashi
179
180   // Output:
181   // ID = 1001
182   // Name = Shashi
183   // ID = 1002
184   // Name = Shashi
185
186   // Output:
187   // ID = 1001
188   // Name = Shashi
189   // ID = 1002
190   // Name = Shashi
191
192   // Output:
193   // ID = 1001
194   // Name = Shashi
195   // ID = 1002
196   // Name = Shashi
197
198   // Output:
199   // ID = 1001
200   // Name = Shashi
201   // ID = 1002
202   // Name = Shashi
203
204   // Output:
205   // ID = 1001
206   // Name = Shashi
207   // ID = 1002
208   // Name = Shashi
209
210   // Output:
211   // ID = 1001
212   // Name = Shashi
213   // ID = 1002
214   // Name = Shashi
215
216   // Output:
217   // ID = 1001
218   // Name = Shashi
219   // ID = 1002
220   // Name = Shashi
221
222   // Output:
223   // ID = 1001
224   // Name = Shashi
225   // ID = 1002
226   // Name = Shashi
227
228   // Output:
229   // ID = 1001
230   // Name = Shashi
231   // ID = 1002
232   // Name = Shashi
233
234   // Output:
235   // ID = 1001
236   // Name = Shashi
237   // ID = 1002
238   // Name = Shashi
239
240   // Output:
241   // ID = 1001
242   // Name = Shashi
243   // ID = 1002
244   // Name = Shashi
245
246   // Output:
247   // ID = 1001
248   // Name = Shashi
249   // ID = 1002
250   // Name = Shashi
251
252   // Output:
253   // ID = 1001
254   // Name = Shashi
255   // ID = 1002
256   // Name = Shashi
257
258   // Output:
259   // ID = 1001
260   // Name = Shashi
261   // ID = 1002
262   // Name = Shashi
263
264   // Output:
265   // ID = 1001
266   // Name = Shashi
267   // ID = 1002
268   // Name = Shashi
269
270   // Output:
271   // ID = 1001
272   // Name = Shashi
273   // ID = 1002
274   // Name = Shashi
275
276   // Output:
277   // ID = 1001
278   // Name = Shashi
279   // ID = 1002
280   // Name = Shashi
281
282   // Output:
283   // ID = 1001
284   // Name = Shashi
285   // ID = 1002
286   // Name = Shashi
287
288   // Output:
289   // ID = 1001
290   // Name = Shashi
291   // ID = 1002
292   // Name = Shashi
293
294   // Output:
295   // ID = 1001
296   // Name = Shashi
297   // ID = 1002
298   // Name = Shashi
299
299 No issues found
```

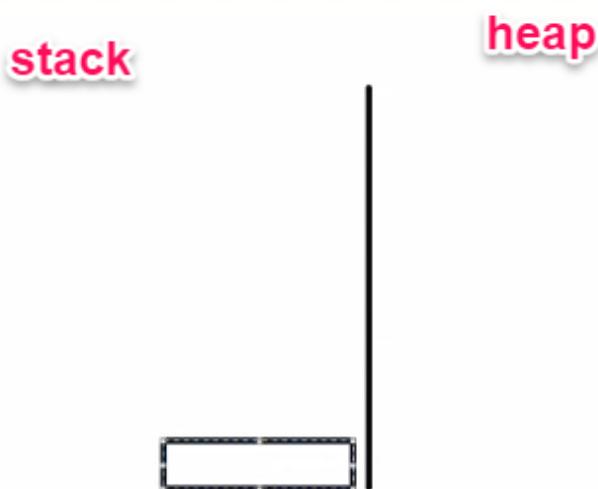
look at this

```
cmd Select C:\WINDOWS\system32\cmd.exe
ID = 1001
Name = Shashi
ID = 1002
Name = Shashi
ID = 1002
Name = Shashi
Press any key to continue . . .
```

```
ValueDemo2.cs  # X | ConsoleApp  
ConsoleApp  
ValueDemo2.cs  
C:\WINDOWS\system32\cmd.exe  
Select C:\WINDOWS\system32\cmd.exe  
5 ID = 1001  
6 Name = Shashi  
7 ID = 1002  
8 Name = Shashi  
9 ID = 1001  
10 Name = Shashi  
11 Press any key to continue . . .  
12  
with the struct  
this is the output  
10 ID = 1001  
11 Name = Shashi  
12 ID = 1002  
13 Name = Shashi  
14 Press any key to continue . . .  
with the class this is the  
output
```

why there is a change in id when i am using class

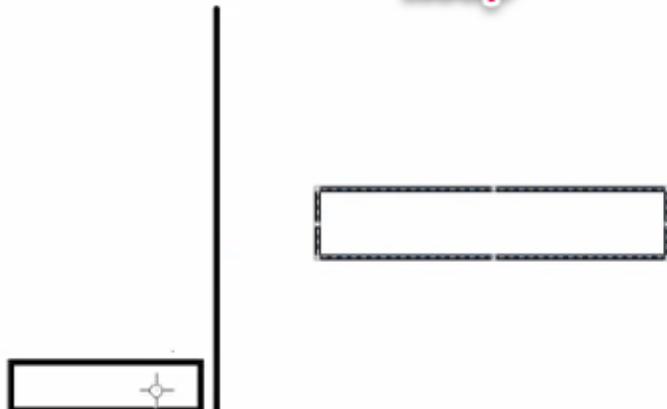
because Structure is a value type -----when you create a structure what happens internally So, i am going to my main here in main i am saying **Student student 1**, So your computers memory will be divide into 2 parts **stack** and **heap** so in the stack it will gone to create me the student1 so this is the reference what we have created ,this is not a object this just a reference



now i am saying **= new Student();** so, in the heap part it will create me a object

stack

heap



now as structure is of reference type it will hold the address of this and here in the student1, i'll id = 100 and name = shashi

Clipboard

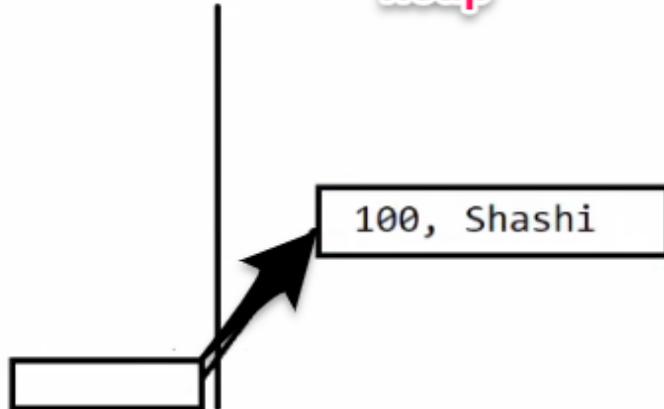
Image

Tools

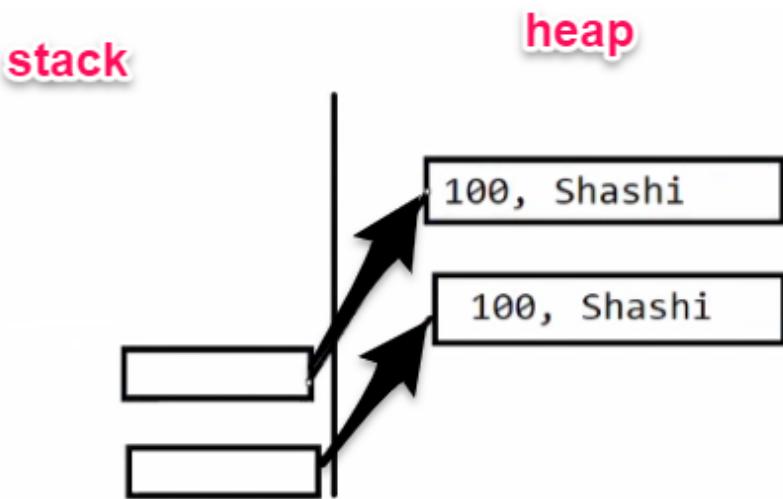
Shapes

stack

heap

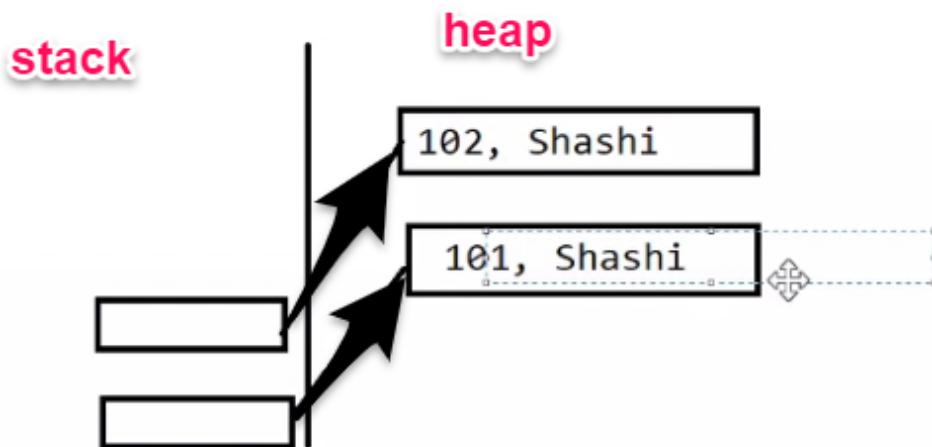


now i'll call the display method it will print me 100 and shashi
but when i say Student student2 = student1;
again in the stack we'll have student2 which have values of
student1
so again in the heap student2 is created which has 100 and shashi



now ,when i say `displaystudent(student2)` it will print me 100 and shashi

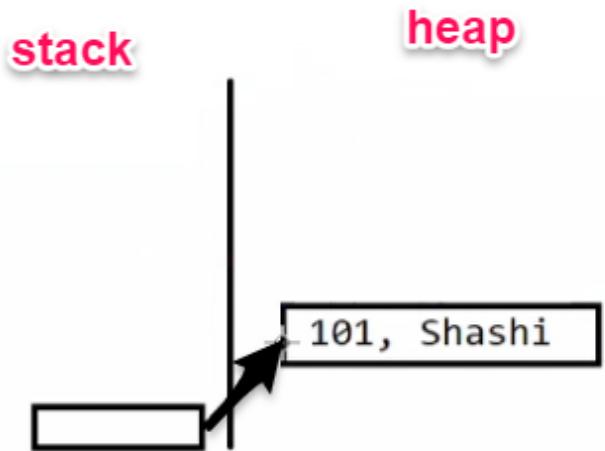
after this i am saying `student2.id = 102`



now when i print student 2 it will print me 1002 and shashi
and when i print student 1 it will print me 1001 and shashi
this is story if it is a structure

What if it is a class?

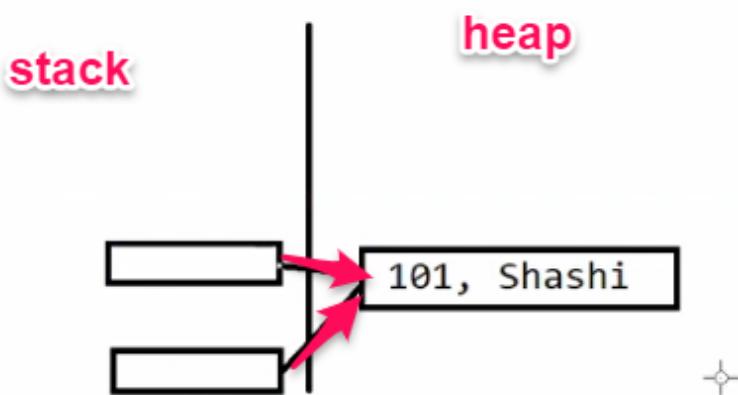
We'll have a stack and heap in stack we have a reference created and in heap we'll have object created



and when i call `displaystudent(student1)` it will print me 101 and shashi

now when i say `Student student2 = student1;` **now here is a twist**

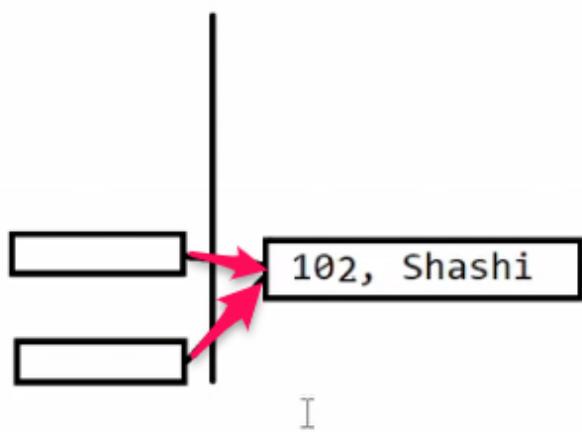
here now, the student 2 will hold the reference of student 1 meaning it will not create a new object there



Now, when i pass this [student2.id=1002](#)

stack

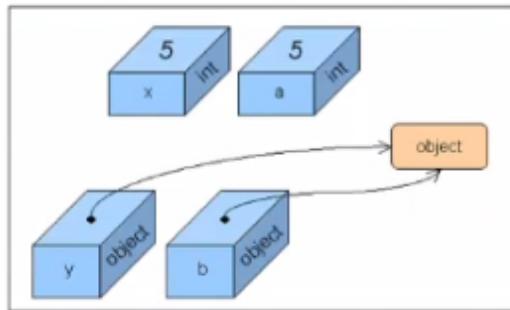
heap



so when i print student2 i'll get 1002 and shashi
and when i print student 1 i'll get 1002 and shashi
because both of them are referring the same reference
so, that's the difference between value type and reference type
**So, in the value type only the values get stored and in
reference type they store the reference and with the
references if any one of the object reference is change the
value other object reference will also get the same thing
affected**

Copying Value vs. Copying Reference

- Now, let us analyze what happens when we introduce two new variables and do some copying from the original variables.
- Assume we have the following code:
 - `int a = x; object b = y;`



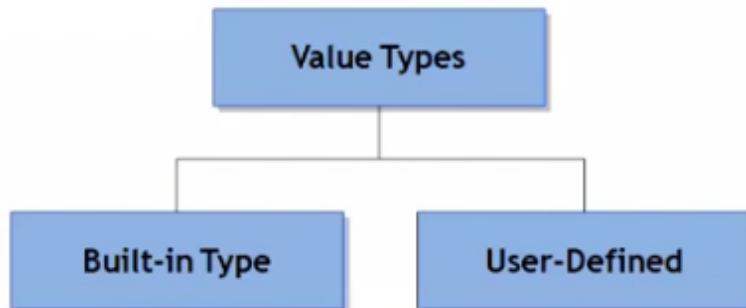
As can be observed, a has a copy of the value of x. If we modify the value of one of these variables, the other variable would remain unchanged. In the case of y and b, both variables refer to the same object. If we alter the state of the object using variable y, then the resulting changes will be observable using variable b, and vice versa.

Comparing Value and Reference Types

Value Type	Reference Type
1. Directly contain the value of a particular data type	1. Directly does not contain the data (object), rather contains the reference of the object
2. Two variables of same data type stores different values or copy of the same value	2. Two variables of same data type can store reference of same object or different object
3. Operations on one variable does not affect another	3. Operation on one variable can affect another

Data Types

Value Types



- Examples of built-in value types:
 - int
 - float

- Examples of user-defined value types:
 - enum
 - struct

Built in value type are those which are the part of system user define --where me being a developer i can create my own types just like we have created a structure now

```
4 references
struct Student
{
    public int id;
    public string name;
}
```

now when you talk about built in type we have all the types like integer , float ,double

But Remember when you talk about the built in types we mainly categorise the built in types to 2 types

1. integer

2. float

Built in Value Types

Type	Description	Range	Size
<u>sbyte</u>	Signed byte integer	- 128 to 127	1 byte
<u>byte</u>	Unsigned byte integer	0 to 255	1 byte
<u>ushort</u>	Unsigned short integer	0 to 65,535	2 byte
<u>short</u>	Signed short integer.	-32,768 to 32,767	2 byte
<u>uint</u>	Unsigned integer. Examples: 26U, 0x1AU (mandatory U suffix)	0 to 4294967295	4 byte
<u>Int</u>	Signed integer. Literals may be in decimal (default) or hexadecimal notation (with an 0x prefix). Examples: 26, 0x1A	-2147483648 to 2147483647	4 byte
<u>ulong</u>	Unsigned long integer. Examples: 26UL, 0x1AUL (mandatory UL suffix)	0 to 2 to the power 64	8 byte

Built in Value Types

Type	Description	Range	Size
<u>long</u>	Signed long integer. Examples: 26L, 0x1AL (mandatory L suffix)	(- 2 to the power 63) to (2 to the power 63) -1	8 byte
<u>float</u>	IEEE 754 single precision floating point number. Examples: 1.2F, 1E10F (mandatory F suffix)	1.5*10^-45 to 3.4*10^38	4 byte
<u>double</u>	IEEE 754 double precision floating point number. Examples: 1.2, 1E10, 1D (optional D suffix)	5.0*10^-324 to 1.7*10^308	8 byte
<u>decimal</u>	Numeric data type suitable for financial and monetary calculations, exact to the 28th decimal place. Example: 123.45M (mandatory M suffix)	1.0*10^-28 to 1.0*10^28	16 byte
<u>char</u>	Unicode character. Example: 'A' (contained within single quotes)	0 to 65,535	stored as integer between 0 to 65535
<u>bool</u>	Boolean value. The only valid literals are true and false.		True or False

Rules and Recommendations for Naming Variables

- Rules
 - Use letters, the underscore, and digits
- Recommendations
 - Avoid using all uppercase letters
 - Avoid starting with an underscore
 - Avoid using abbreviations
 - Use PascalCasing naming in multiple-word names



Integer - In integer we have numeric values you will have short, byteall this fall under numeric including char type, char is also an integer type

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays three open files: "IntTypeDemo.cs*", "PassbyValueDemo2.cs*", and "ConsoleApp*". The current file is "ConsoleApp.cs", which contains the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         static void Main()
12         {
13             int a; //declaration
14             a = 10; //initialization
15         }
16     }
17 }
18
```

A vertical yellow highlight bar is positioned to the left of the code, spanning from line 10 to line 14. The code editor has a light gray background with syntax highlighting. The status bar at the bottom right shows the text "ConsoleApp.IntTypeDemo".

```
IntTypeDemo.cs*  X  PassbyValueDemo2.cs*  ConsoleApp*
```

```
ConsoleApp
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         int a = 10; //no Error
12     }
13
14     static void Main()
15     {
16         int a; //decl
17         a = 10; //init
18     }
19 }
20 }
```

0 references

0 references

```
133 %  ✘ 0  ⚠ 1  ← →  ⌂ ⌄
```

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays three open files: "IntTypeDemo.cs*", "PassbyValueDemo2.cs*", and "ConsoleApp*". The main area is a code editor for "ConsoleApp.cs". The code is as follows:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         int a; //ERROR
12         a = 10;
13     }
14
15     static void Main()
16     {
17         int a; //decl
18         a = 10; //init
19     }
20 }
```

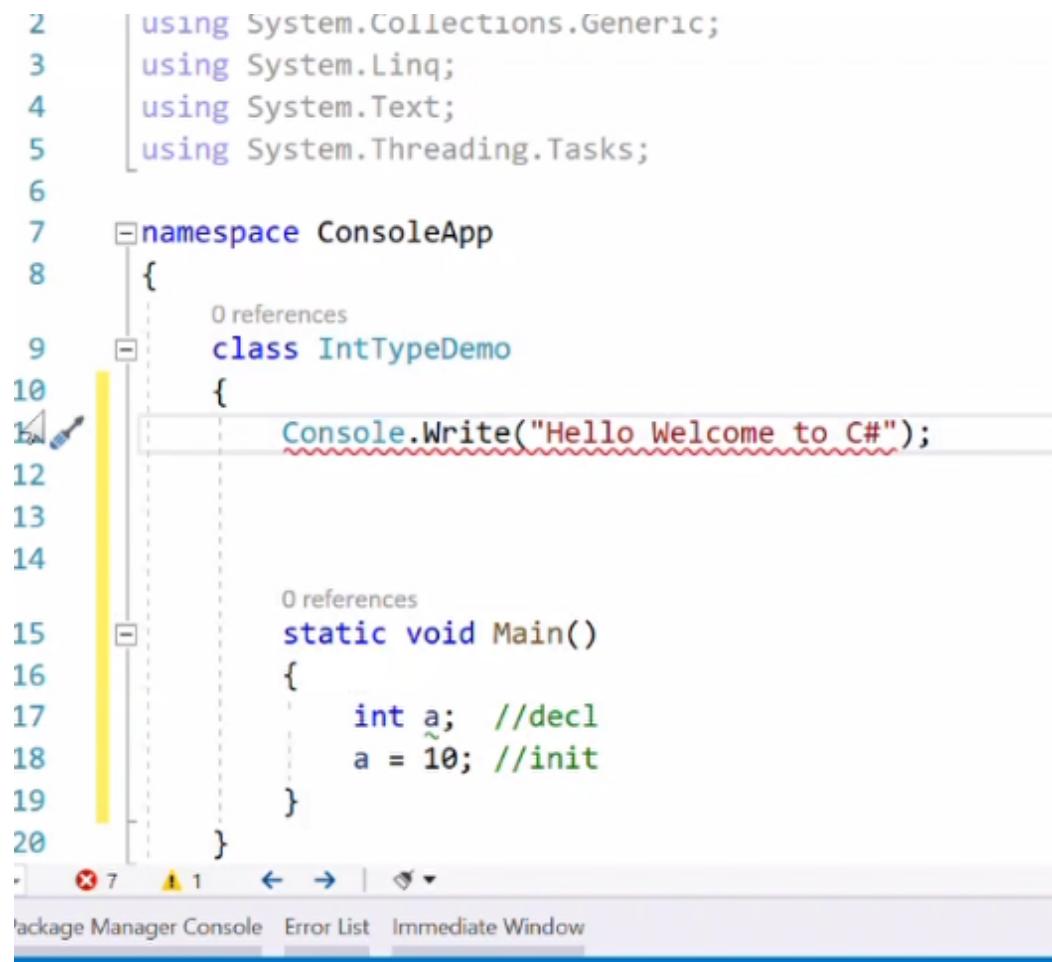
A vertical yellow bar highlights the declaration of variable 'a' at line 11. A red error icon is placed next to the assignment statement 'a = 10' at line 12. The status bar at the bottom shows "133 %", "x 3", "1", and navigation icons. Below the editor, tabs for "Output", "Package Manager Console", "Error List", and "Immediate Window" are visible. The bottom navigation bar includes "Ready" and search bars.

because this is a two different lines we have a declaration we have a initialization

**Remember In the class you
can declare the variables but
you cannot initialize them**

because initialization becomes a statement
seprately you cannot do it

If the declaration is a seprate line and initialization is a seprate line so inside the class you will get a error but if it is in the same line it is still ok



```
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         Console.WriteLine("Hello Welcome to C#");
12
13
14
15         static void Main()
16         {
17             int a; //decl
18             a = 10; //init
19         }
20     }
21 }
```

The screenshot shows a Microsoft Visual Studio IDE window. The code editor displays a C# program. Line 11 contains a syntax error: `Console.WriteLine("Hello Welcome to C#");` is highlighted with a red squiggly underline. The code editor has a vertical yellow margin bar on the left. The status bar at the bottom shows tabs for 'Package Manager Console', 'Error List', and 'Immediate Window'. The error list tab is currently selected.

Why are we getting this red line here?

Inside the class You can have only 2 things

1. Declaration of variable with the initialization if required
2. Methods

you cannot have anything else apart from this 2

```
using System.IO;
```

```
|namespace ConsoleApp
```

```
{
```

```
    0 references
```

```
    class IntTypeDemo
```

```
{
```

```
        int z = 10;
```

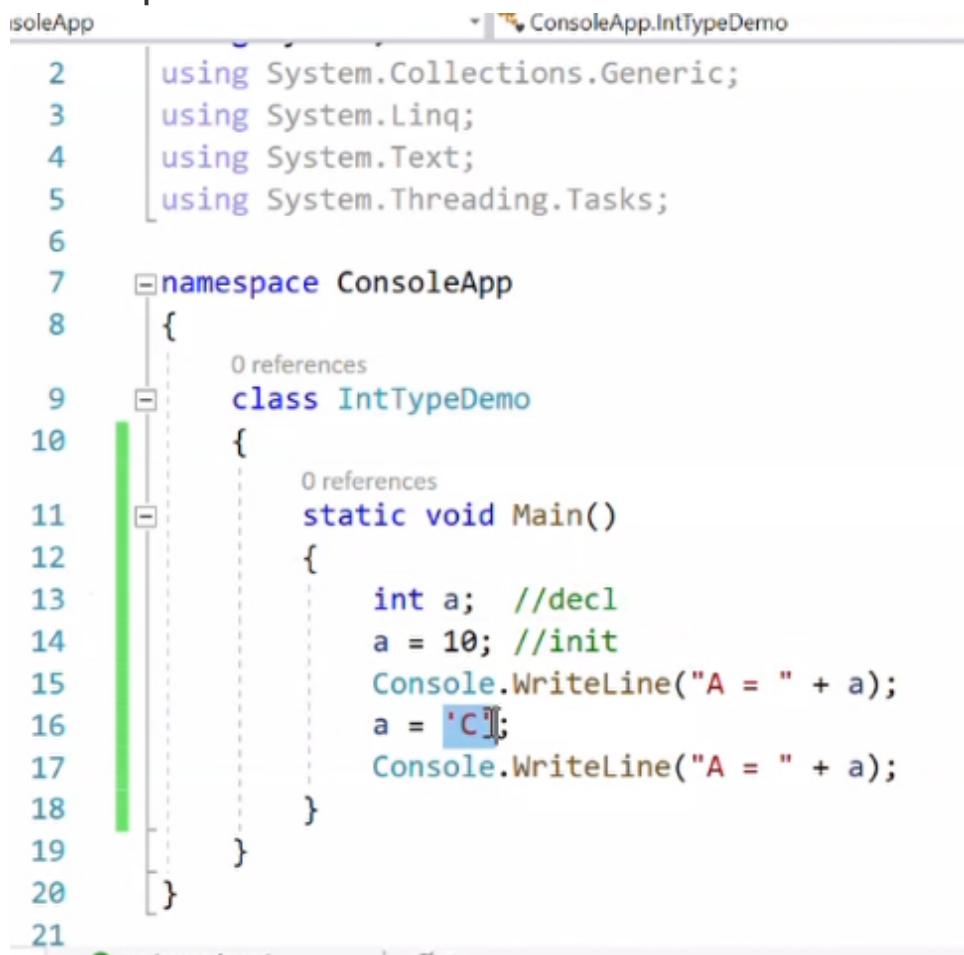
```
        0 references
```

```
        void Show()
```

```
{
```

```
}
```

Is this possible ?



```
soleApp
```

```
ConsoleApp.IntTypeDemo
```

```
2     using System.Collections.Generic;
3     using System.Linq;
4     using System.Text;
5     using System.Threading.Tasks;
6
7     namespace ConsoleApp
8     {
9         0 references
10        class IntTypeDemo
11        {
12            0 references
13            static void Main()
14            {
15                int a; //decl
16                a = 10; //init
17                Console.WriteLine("A = " + a);
18                a = 'C';
19                Console.WriteLine("A = " + a);
20            }
21        }
```

it is directly storing the value that's it is a character what i am trying to store

Remember - in types we have numeric type and character is also a numeric type

it will just store the ASCII of it

```
A = 10
A = 67
Press any key to continue . . .
```

Remember this all are of the same type ,type wise they are same
But how will i make sure that it will print me c as a character?
that's where i have to mention that boss change this to character
type don't print me number

```
1  using System.Collections.Generic;
2  using System.Linq;
3  using System.Text;
4  using System.Threading.Tasks;
5
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         static void Main()
12         {
13             int a; //decl
14             a = 10; //init
15             Console.WriteLine("A = " + a);
16             a = 'C';
17             Console.WriteLine("A = " + (char)a);
18         }
19     }
20 }
```

I am Explicitly saying please print me character

```
A = 10
A = C
Press any key to continue . . .
```

Float

What will be the output of this ?

```
namespace ConsoleApp
{
    class IntTypeDemo
    {
        static void Main()
        {
            float f = 100;
            Console.WriteLine("F = " + f);
        }
    }
}
```

100.0?

```
C:\WINDOWS\system32\cmd.exe
F = 100
Press any key to continue . .
```

Why its printing 100?

```
using System.Threading.Tasks;

namespace ConsoleApp
{
    class IntTypeDemo
    {
        static void Main()
        {
            float f = 100.98;
            Console.WriteLine("F = " + f);

            /*
            int a; //decl
            a = 10; //init
            Console.WriteLine("A = " + a);
            a = 'C';
            Console.WriteLine("A = " + (char)a);
            */
        }
    }
}
```

in float as soon as i
put .98 look at this
I got Error



```
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         static void Main()
12         {
13             float f = 100.98;
14             Console.WriteLine
15
16             /*
17             int a; //decl
18             a = 10; //init
19             Console.WriteLine("A = " + a);
20             a = 'C';
21             Console.WriteLine("A = " + (char)a);
22         }
23     }
24 }
```

look at this when i move my mouse pointer on this what is the message i get
it says its a double value whenever you put . it will treat that as a double value and double value cannot be stored in a float variable

Float is like 2BHK you cannot put double which is 3BHK in 2BHK
So i have to specify that boss this is not 3BHK this is 2BHK

so i'll write **100.98f** i explicitly have to specify **Literal**

```
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class IntTypeDemo
10     {
11         static void Main()
12         {
13             float f = 100.98f;
14             Console.Writeline
15
16             /*
17             int a; //decl
18             a = 10; //init
19             Console.WriteLine("A = " + a);
20             a = 'C';
21             Console.WriteLine("A = " + (char)a);
22         }
23     }
24 }
```