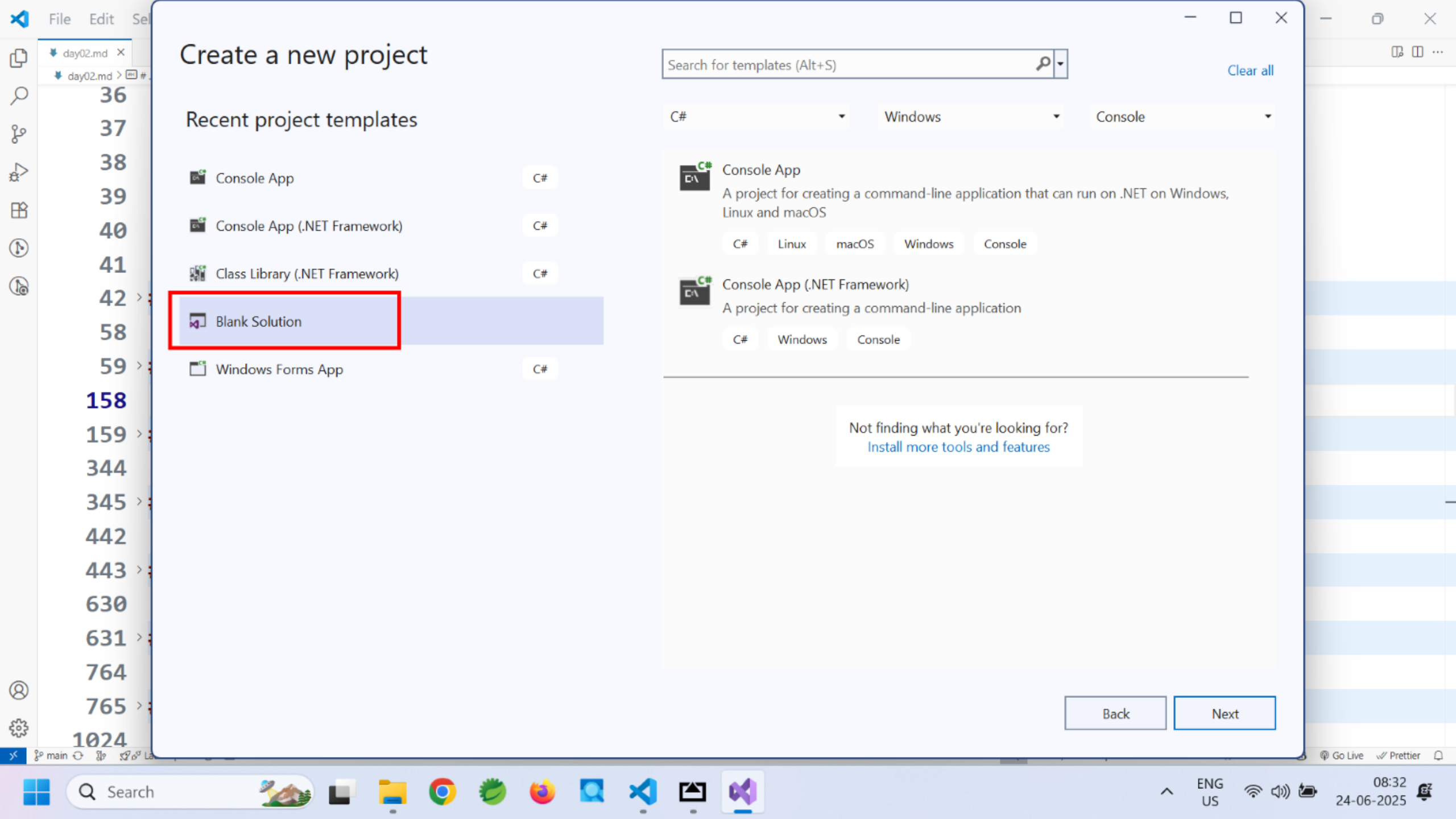
C#.NET @ Sunbeam Infotech

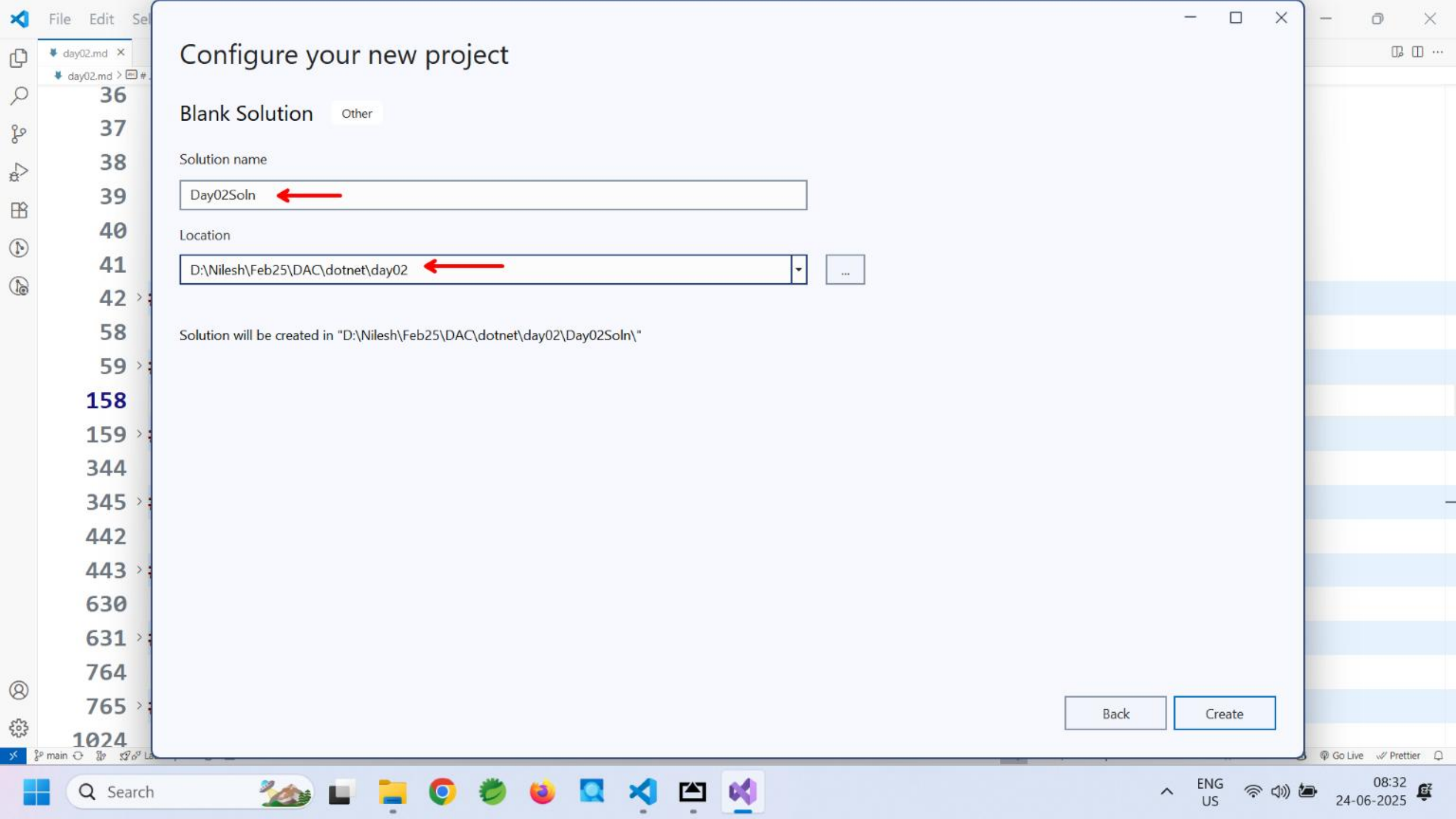
Trainer: Nilesh Ghule

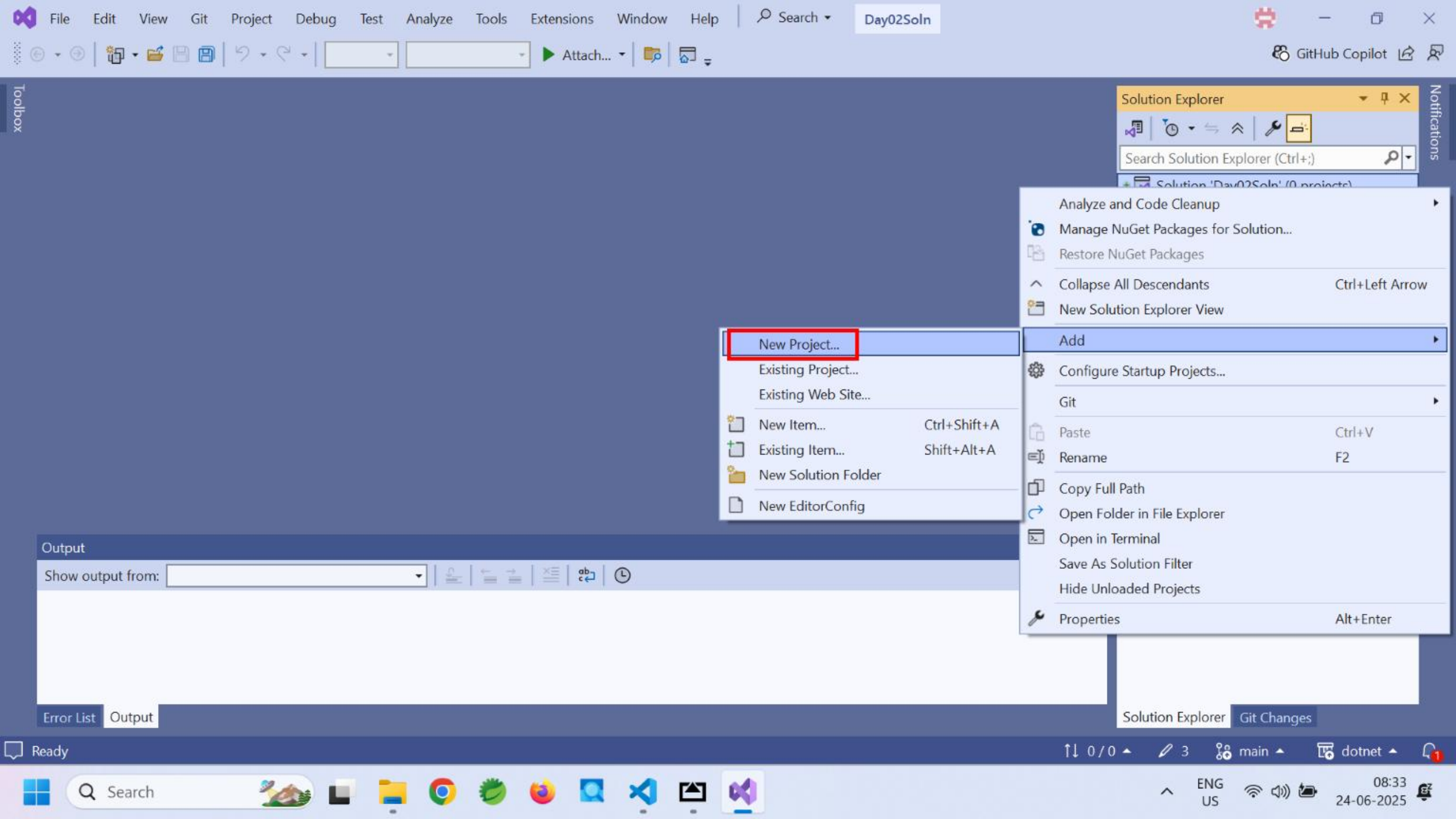
Day 02 - .Net

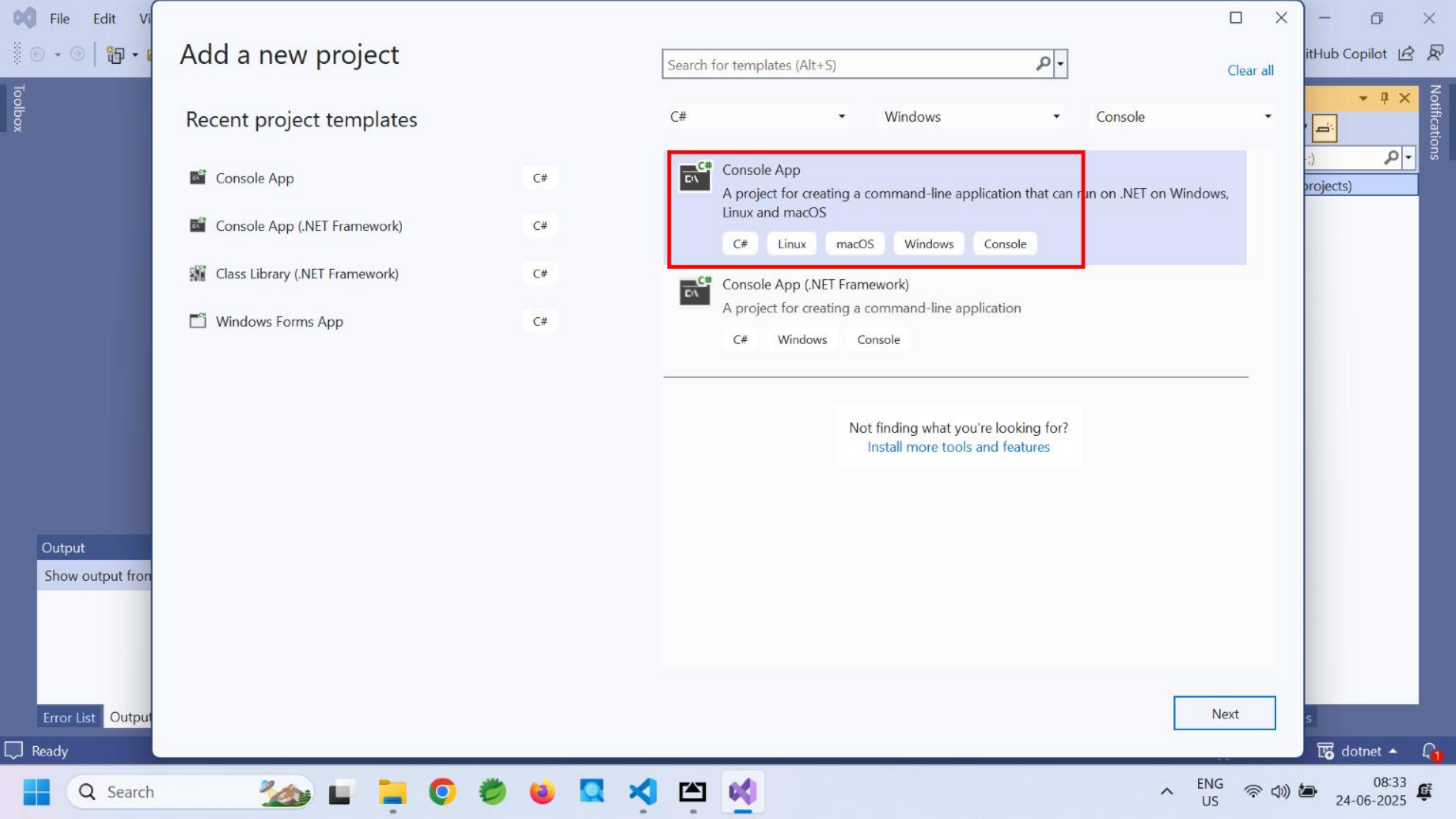
1. .NET versions
2. C# lang versions
   - latest C# 14 for .Net 10.
3. 01_HelloWorld
   - .Net core Hello
   - Top level stmts
4. 02_Namespaces
   - Logical containers
   - default namespace
   - split namespace
   - File scoped ns
   - Implicit usings for common ns.
   - Namespace alias
   - global usings

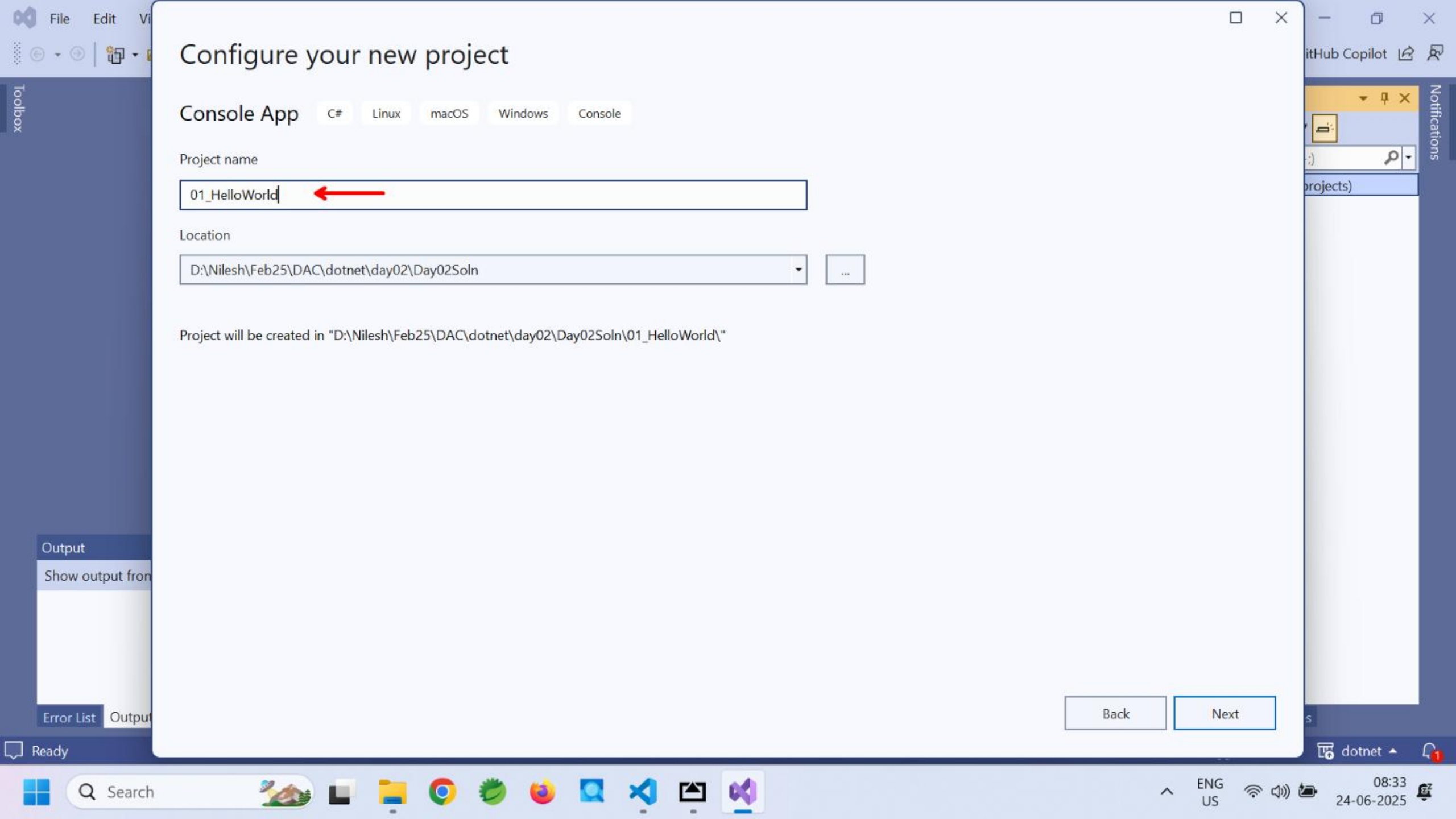
   *multiple entry points.
5. 03_Data Types.
   - Parse()
   - Convert
   - checked/ unchecked
6. 04_Struct Enums
   - Struct: props, ctor x
   - obj initializers
   - collection init
   - class vs struct
7. 05_boxing
   - int ⟺ object
   - boxing needed in non-gen collections, but not in gen col.
8. 06_methods
   - pass by val/ref
   - ref & out keyword
   - local methods
9. 07_inheritance
   - core concepts
   - base ctor call
   - new method (hiding)
   - obj slicing
   - is keyword
   - as keyword
   - virtual methods
   - new virtual
10. 08_inheritance
    - inheritance hierarchy
    - util methods - static class
    - abstract method
    - abstract class
    - sealed override
    - sealed class
    - array of objs

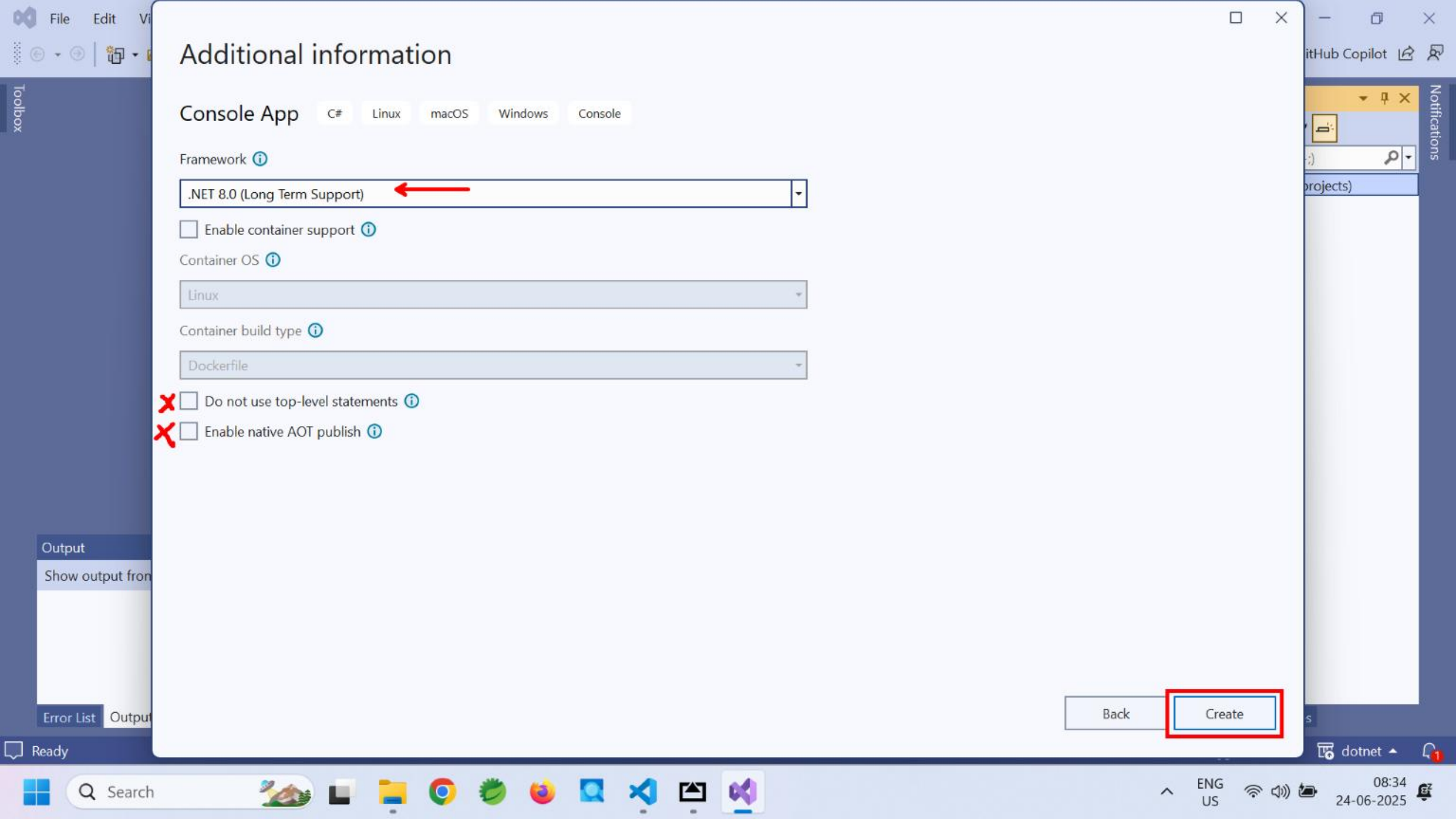

File Edit Sel
day02.md
36
37
38
39
40
41
42
58
59
158
159
344
345
442
443
630
631
764
765
1024
Create a new project
Recent project templates
Console App C#
Console App (.NET Framework) C#
Class Library (.NET Framework) C#
Blank Solution
Windows Forms App C#
Search for templates (Alt+S)
Clear all
C#
Windows
Console
Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# Linux macOS Windows Console
Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console
Not finding what you're looking for?
Install more tools and features
Back
Next
main
Go Live
Prettier
Search
ENG US
08:32
24-06-2025

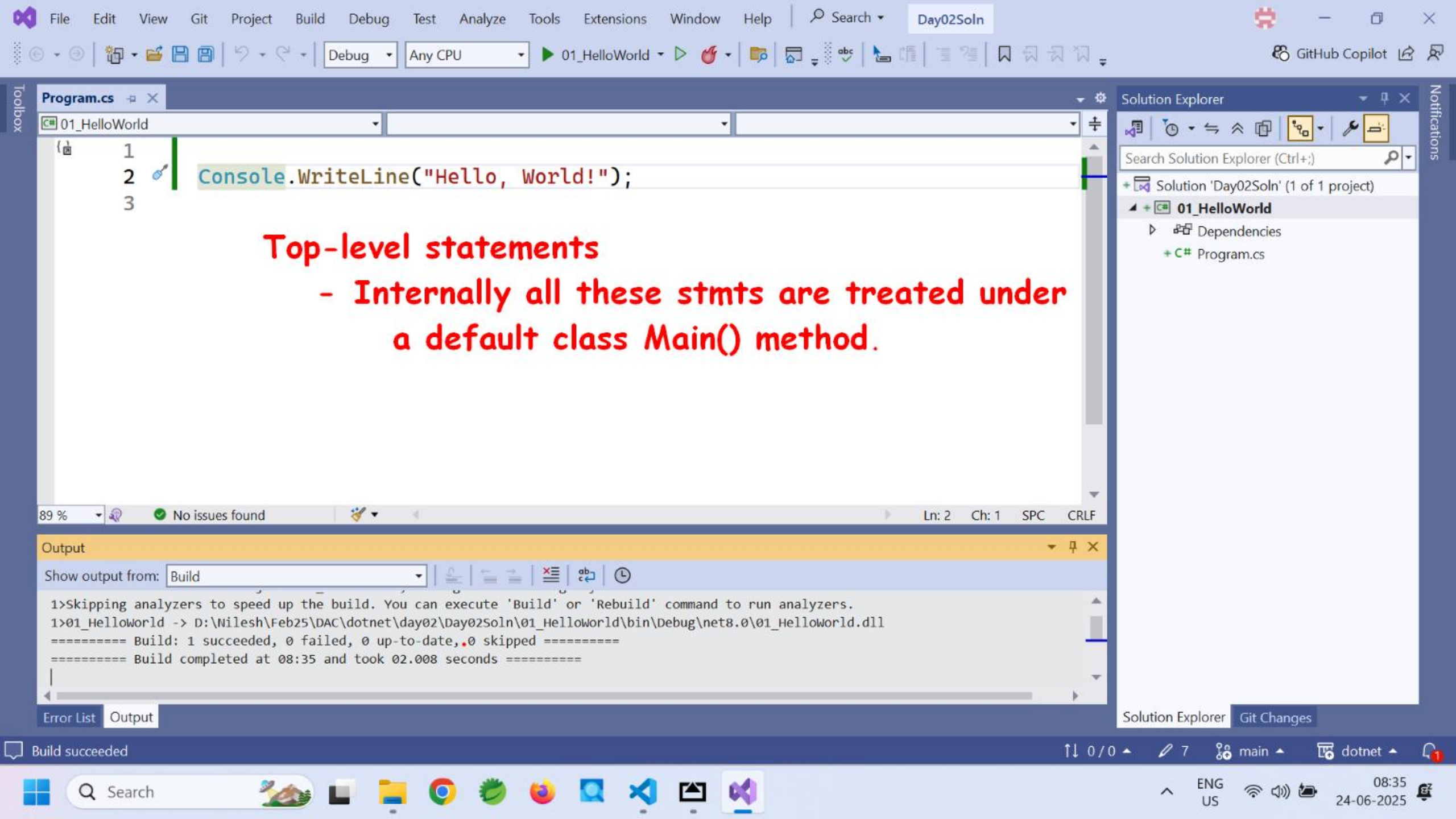

File
Edit
day02.md
Configure your new project
Blank Solution
Other
Solution name
Day02Soln
Location
D:\Nilesh\Feb25\DAC\dotnet\day02
...
Solution will be created in "D:\Nilesh\Feb25\DAC\dotnet\day02\Day02Soln\"
Back
Create
Search
ENG
US
08:32
24-06-2025

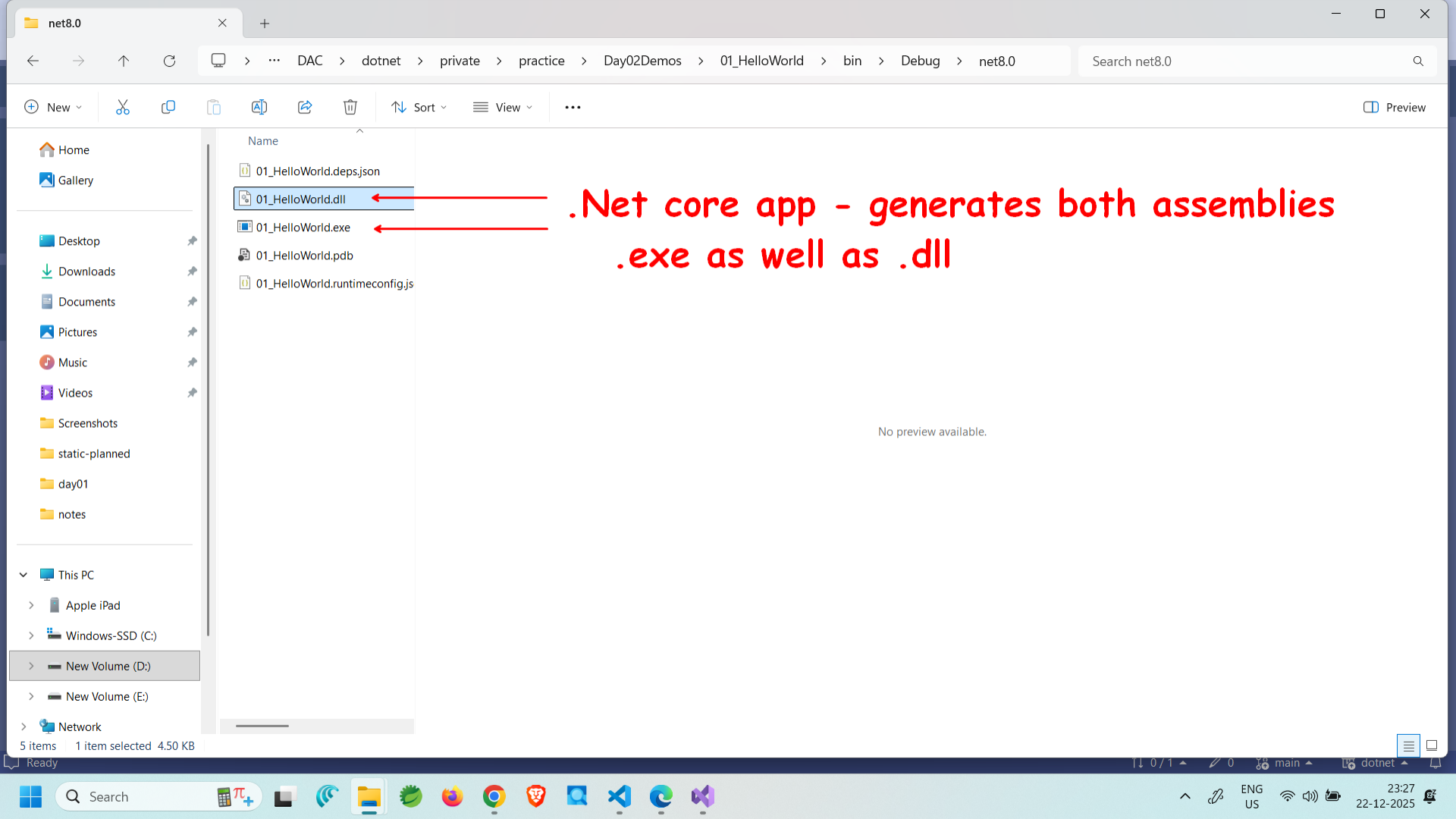

File
Edit
View
Git
Project
Debug
Test
Analyze
Tools
Extensions
Window
Help
Search
Day02Soln
Attach...
GitHub Copilot
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+;)
Notifications
Analyze and Code Cleanup
Manage NuGet Packages for Solution...
Restore NuGet Packages
Collapse All Descendants Ctrl+Left Arrow
New Solution Explorer View
Add
Configure Startup Projects...
Git
Paste Ctrl+V
Rename F2
Copy Full Path
Open Folder in File Explorer
Open in Terminal
Save As Solution Filter
Hide Unloaded Projects
Properties Alt+Enter
New Project...
Existing Project...
Existing Web Site...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Solution Folder
New EditorConfig
Output
Show output from:
Error List
Output
Solution Explorer
Git Changes
Ready
0 / 0
3
main
dotnet
Search
ENG
US
08:33
24-06-2025


File
Edit
Add a new project
Search for templates (Alt+S)
Clear all
Recent project templates
Console App
C#
Console App (.NET Framework)
C#
Class Library (.NET Framework)
C#
Windows Forms App
C#
C#
Windows
Console
Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C#
Linux
macOS
Windows
Console
Console App (.NET Framework)
A project for creating a command-line application
C#
Windows
Console
Not finding what you're looking for?
Install more tools and features
Next
Toolbox
Notifications
Output
Show output from
Error List
Output
Ready
dotnet
Search
ENG
US
08:33
24-06-2025


File
Edit
Configure your new project
Console App
C#
Linux
macOS
Windows
Console
Project name
01_HelloWorld
Location
D:\Nilesh\Feb25\DAC\dotnet\day02\Day02Soln
...
Project will be created in "D:\Nilesh\Feb25\DAC\dotnet\day02\Day02Soln\01_HelloWorld\"
Back
Next
Toolbox
Notifications
Output
Show output from
Error List
Output
Ready
dotnet
Search
ENG
US
08:33
24-06-2025


File
Edit
Additional information
Console App
C#
Linux
macOS
Windows
Console
Framework
.NET 8.0 (Long Term Support)
Enable container support
Container OS
Linux
Container build type
Dockerfile
Do not use top-level statements
Enable native AOT publish
Back
Create
Toolbox
Notifications
Output
Show output from
Error List
Output
Ready
dotnet
Search
ENG
US
08:34
24-06-2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
01_HelloWorld
GitHub Copilot
Program.cs
01_HelloWorld
1
2 Console.WriteLine("Hello, World!");
3
Top-level statements
- Internally all these stmts are treated under
a default class Main() method.
89 %
No issues found
Ln: 2 Ch: 1 SPC CRLF
Output
Show output from: Build
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>01_HelloWorld -> D:\Nilesh\Feb25\DAC\dotnet\day02\Day02Soln\01_HelloWorld\bin\Debug\net8.0\01_HelloWorld.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date,.0 skipped ==========
========== Build completed at 08:35 and took 02.008 seconds ==========
Error List
Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Day02Soln' (1 of 1 project)
01_HelloWorld
Dependencies
Program.cs
Solution Explorer
Git Changes
Build succeeded
0 / 0
7
main
dotnet
Search
ENG
US
08:35
24-06-2025


net8.0
DAC
dotnet
private
practice
Day02Demos
01_HelloWorld
bin
Debug
net8.0
Search net8.0
New
Sort
View
Preview
Home
Gallery
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
static-planned
day01
notes
This PC
Apple iPad
Windows-SSD (C:)
New Volume (D:)
New Volume (E:)
Network
Name
01_HelloWorld.deps.json
01_HelloWorld.dll
01_HelloWorld.exe
01_HelloWorld.pdb
01_HelloWorld.runtimeconfig.js
.Net core app - generates both assemblies
.exe as well as .dll
No preview available.
5 items
1 item selected 4.50 KB
Ready
main
dotnet
Search
ENG
US
23:27
22-12-2025

Add a new project

Search for templates (Alt+S)

Clear all

C# | Windows | Console

Recent project templates

- Console App — C#
- Console App (.NET Framework) — C#
- Class Library (.NET Framework) — C#
- Windows Forms App — C#

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

Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console

Not finding what you're looking for?
Install more tools and features

Next

Output
Show output from
1>Skipping an
1>01_HelloWor
========== Bu
========== Bu
Error List Output
This item does not s
dotnet
Search
ENG US
08:37
24-06-2025

Additional information

Console App

C# Linux macOS Windows Console

Framework

.NET 8.0 (Long Term Support)

Enable container support

Container OS

Linux

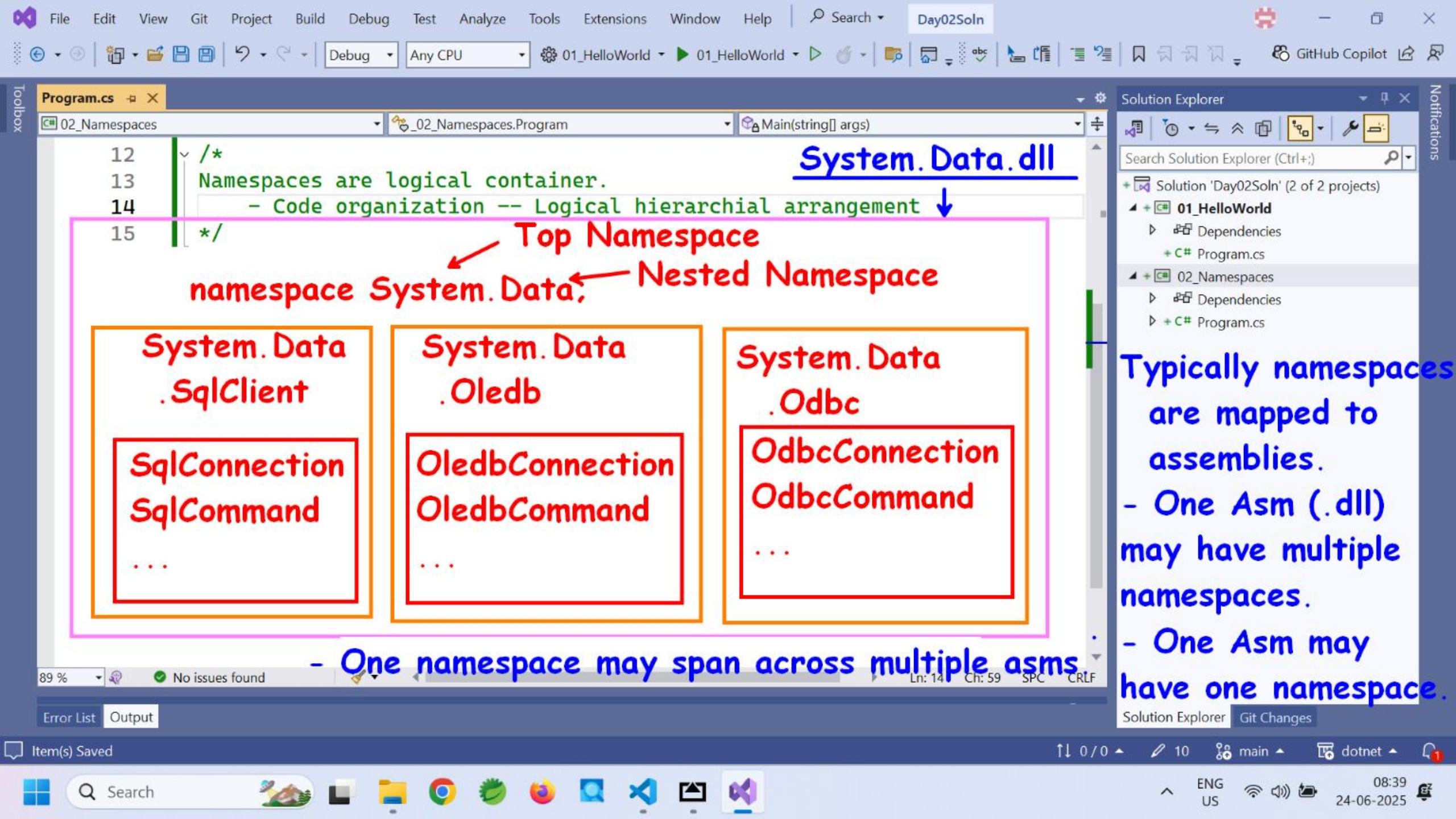
Container build type

Dockerfile

Do not use top-level statements

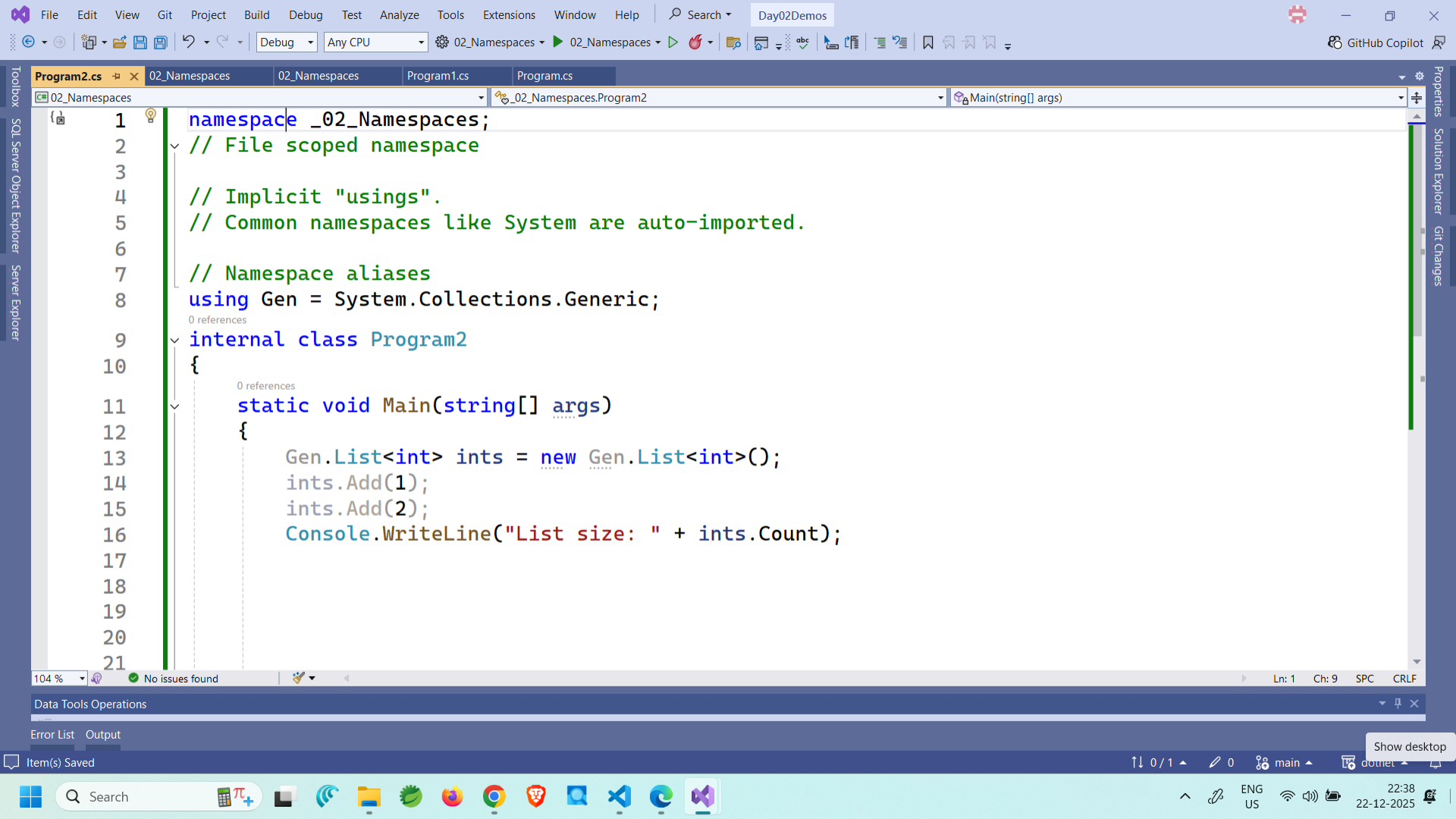
Enable native AOT publish

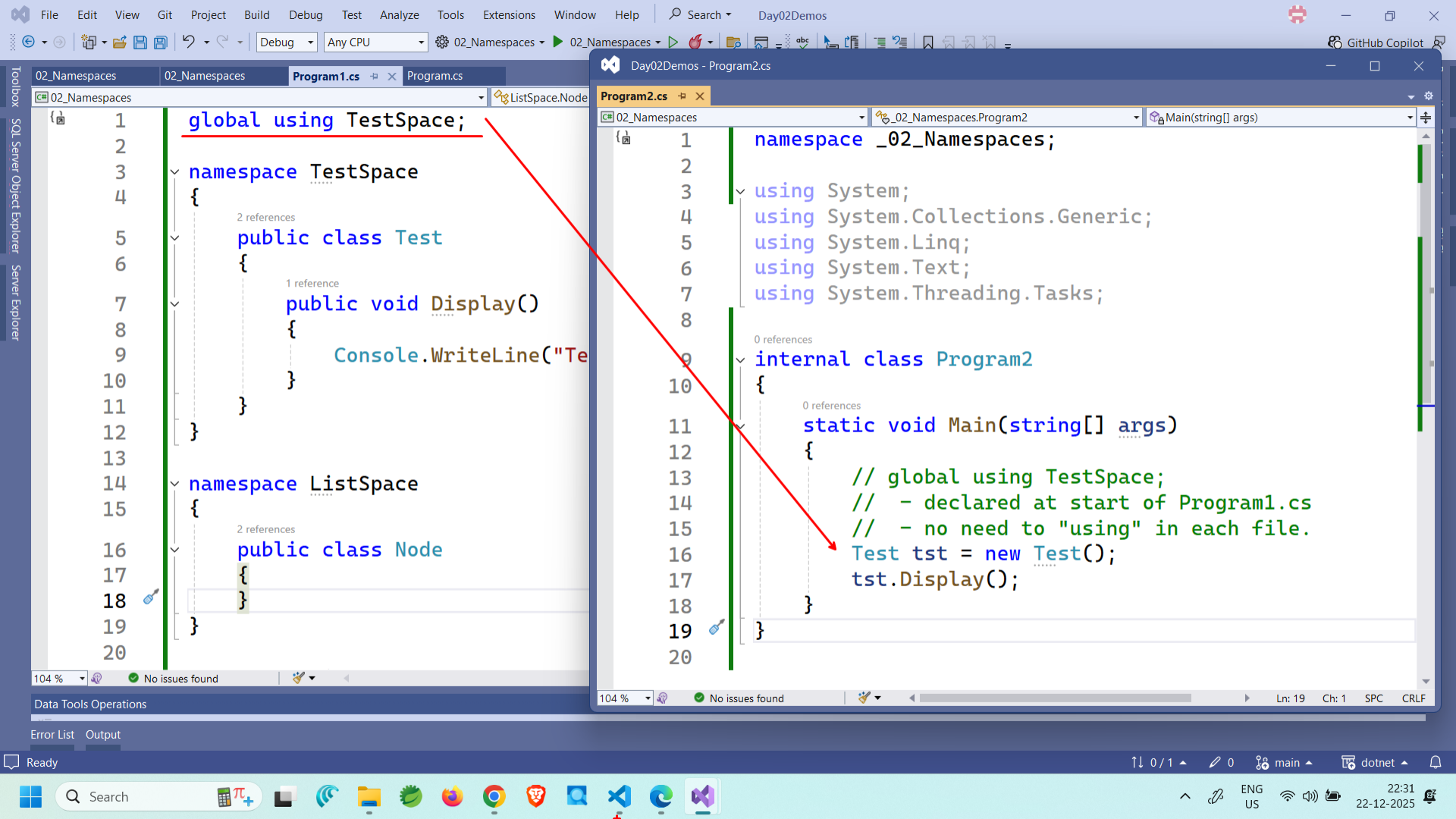
Back Create

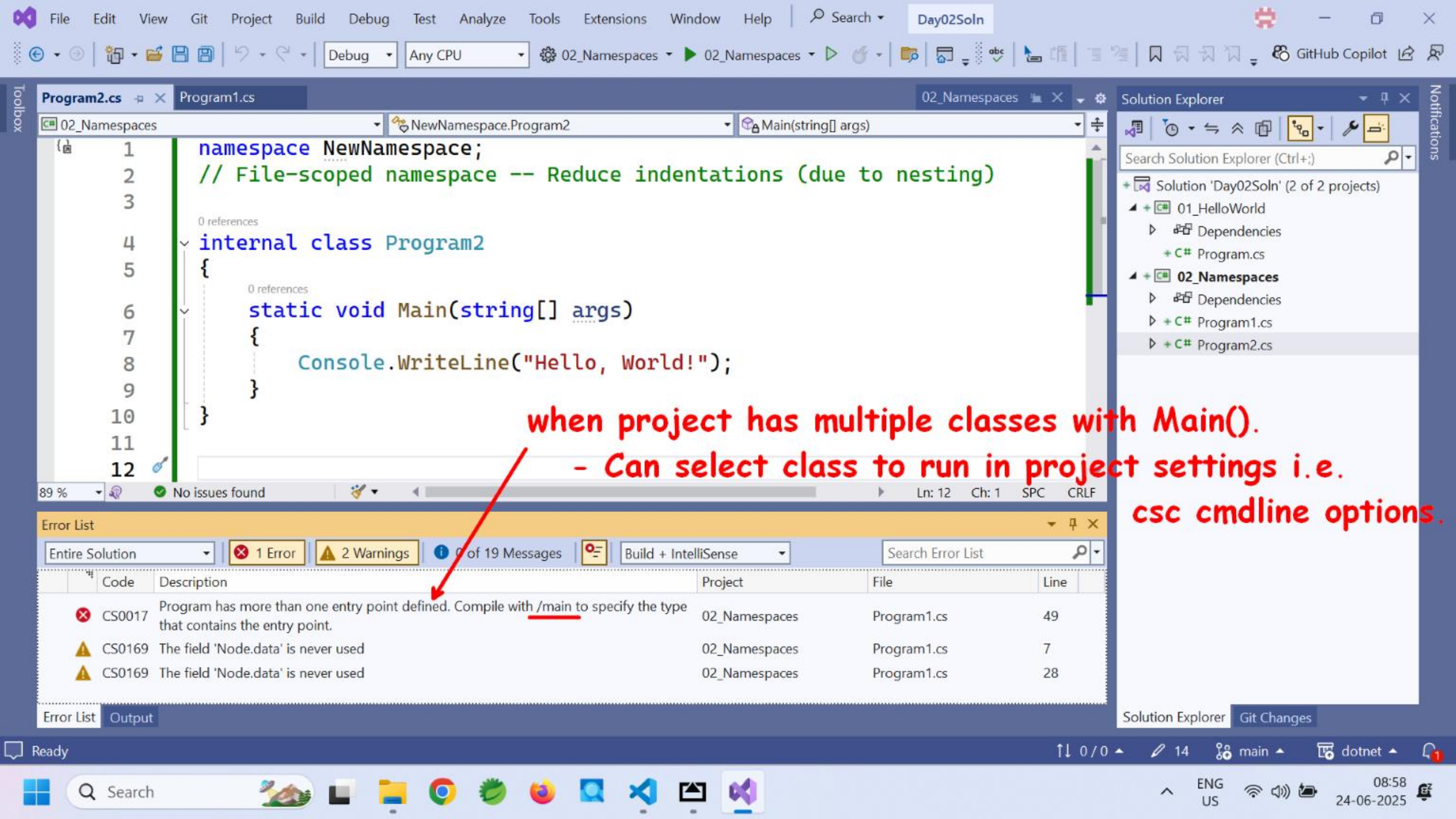

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
01_HelloWorld
01_HelloWorld
GitHub Copilot
Program.cs
02_Namespaces
_02_Namespaces.Program
Main(string[] args)
12 /*
13 Namespaces are logical container.
14 - Code organization -- Logical hierarchial arrangement
15 */
System.Data.dll
Top Namespace
Nested Namespace
namespace System.Data;
System.Data
.SqlClient
SqlConnection
SqlCommand
...
System.Data
.Oledb
OledbConnection
OledbCommand
...
System.Data
.Odbc
OdbcConnection
OdbcCommand
...
- One namespace may span across multiple asms
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Day02Soln' (2 of 2 projects)
01_HelloWorld
Dependencies
Program.cs
02_Namespaces
Dependencies
Program.cs
Typically namespaces are mapped to assemblies.
- One Asm (.dll) may have multiple namespaces.
- One Asm may have one namespace.
89 %
No issues found
Ln: 14 Ch: 59 SPC CRLF
Error List
Output
Solution Explorer
Git Changes
Item(s) Saved
0 / 0
10
main
dotnet
Search
ENG
US
08:39
24-06-2025

02_Namespace - Program1.cs

```
    }
}

/*
Namespaces are logical container.
    - Code organization -- Logical hierarchial arrangement
    - Avoids name conflicts
        - e.g. ListSpace = Node class + List class
        and TreeSpace = Node class + List class

By default for each project, one namespace is created.
    - It has same name as of project name.
    - Programmer can create additional namespaces in same file
        or across files.

Whole namespace can be written at once.
    - or can be split in multiple parts in same file (e.g ListSpace ...)
    - or can be split in multiple parts in multiple files/assemblies.
*/
```

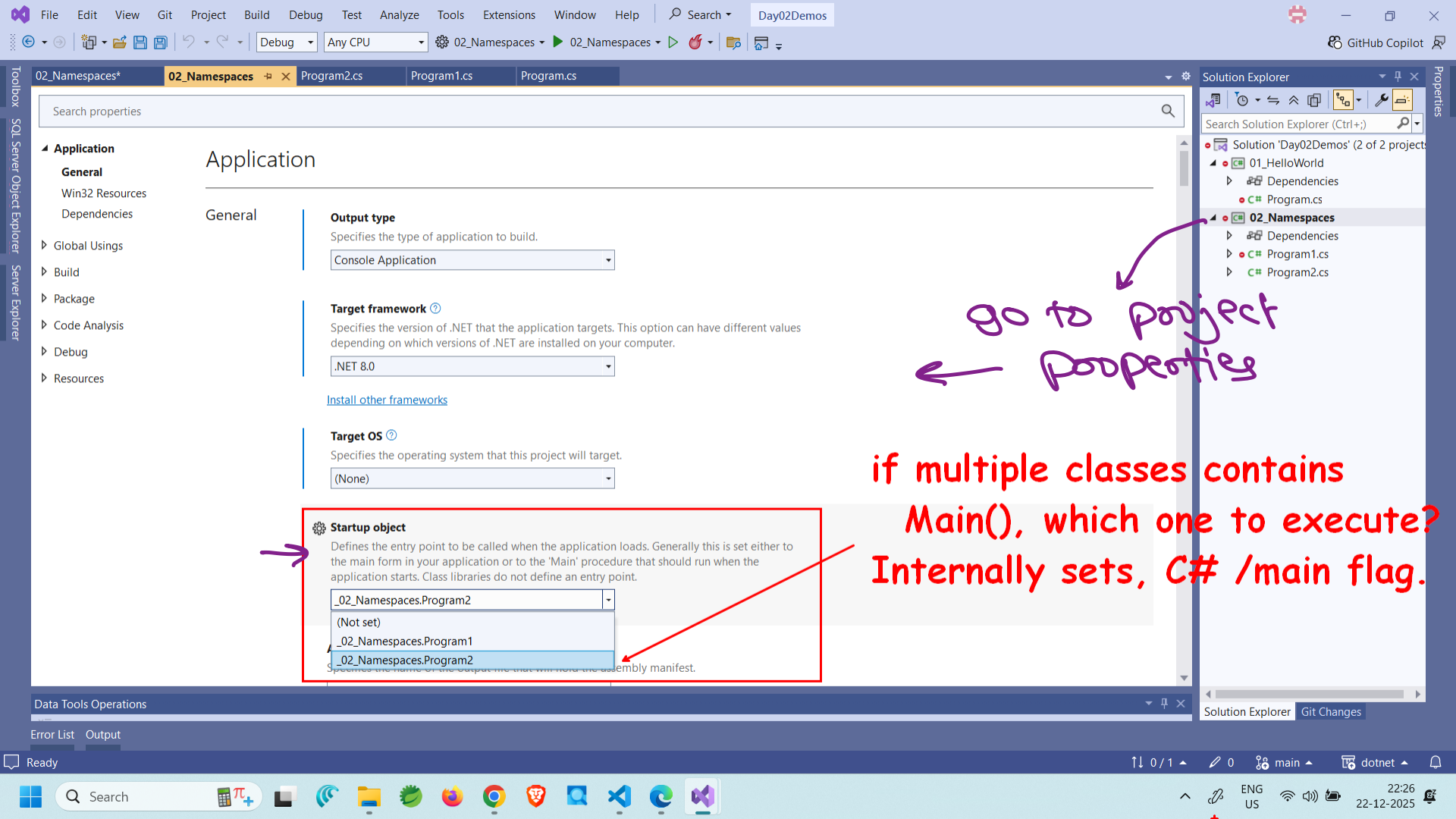

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Demos
Debug
Any CPU
02_Namespaces
02_Namespaces
GitHub Copilot
Program2.cs
02_Namespaces
02_Namespaces
Program1.cs
Program.cs
02_Namespaces
_02_Namespaces.Program2
Main(string[] args)
namespace _02_Namespaces;
// File scoped namespace
// Implicit "usings".
// Common namespaces like System are auto-imported.
// Namespace aliases
using Gen = System.Collections.Generic;
0 references
internal class Program2
{
0 references
static void Main(string[] args)
{
Gen.List<int> ints = new Gen.List<int>();
ints.Add(1);
ints.Add(2);
Console.WriteLine("List size: " + ints.Count);
104 %
No issues found
Ln: 1
Ch: 9
SPC
CRLF
Data Tools Operations
Error List
Output
Item(s) Saved
0 / 1
0
main
Show desktop
Search
ENG
US
22:38
22-12-2025

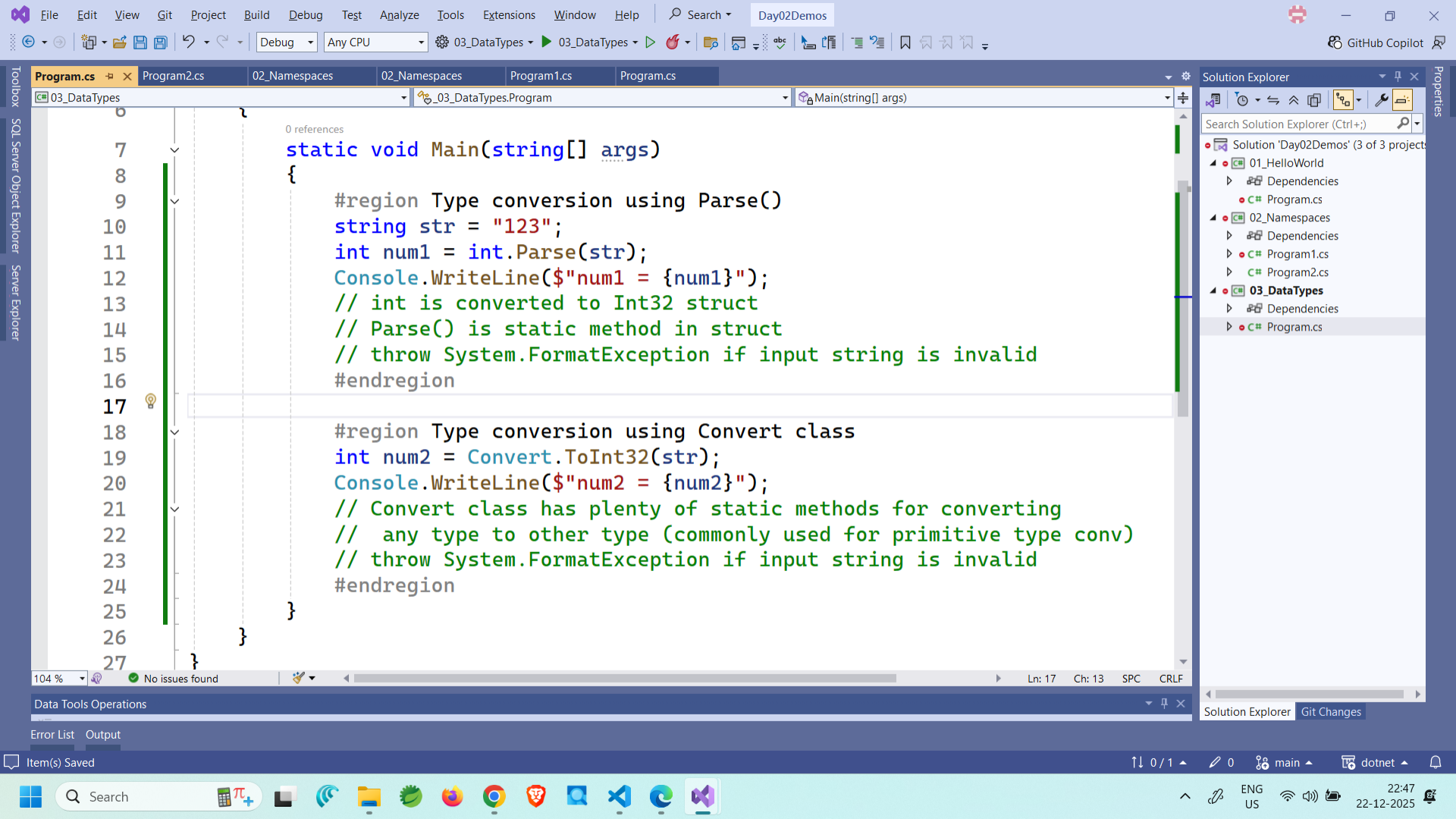

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Demos
Debug
Any CPU
02_Namespaces
GitHub Copilot
Toolbox
SQL Server Object Explorer
Server Explorer
02_Namespaces
02_Namespaces
Program1.cs
Program.cs
ListSpace.Node
global using TestSpace;
namespace TestSpace
{
2 references
public class Test
{
1 reference
public void Display()
{
Console.WriteLine("Te
}
}
}
namespace ListSpace
{
2 references
public class Node
{
}
}
104 %
No issues found
Data Tools Operations
Error List
Output
Day02Demos - Program2.cs
Program2.cs
02_Namespaces
_02_Namespaces.Program2
Main(string[] args)
namespace _02_Namespaces;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
0 references
internal class Program2
{
0 references
static void Main(string[] args)
{
// global using TestSpace;
// - declared at start of Program1.cs
// - no need to "using" in each file.
Test tst = new Test();
tst.Display();
}
}
104 %
No issues found
Ln: 19
Ch: 1
SPC
CRLF
Ready
0 / 1
0
main
dotnet
Search
ENG
US
22:31
22-12-2025

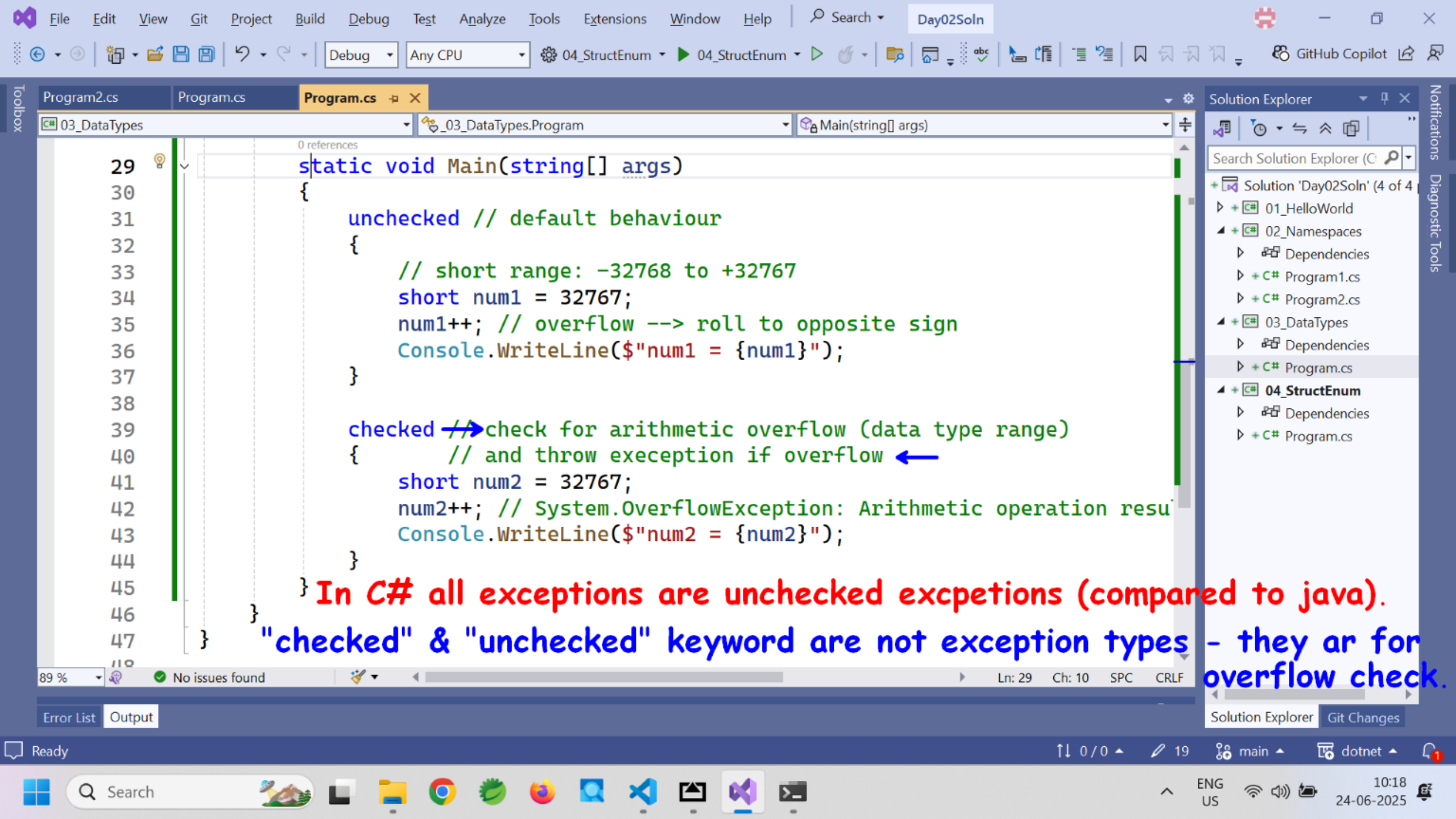


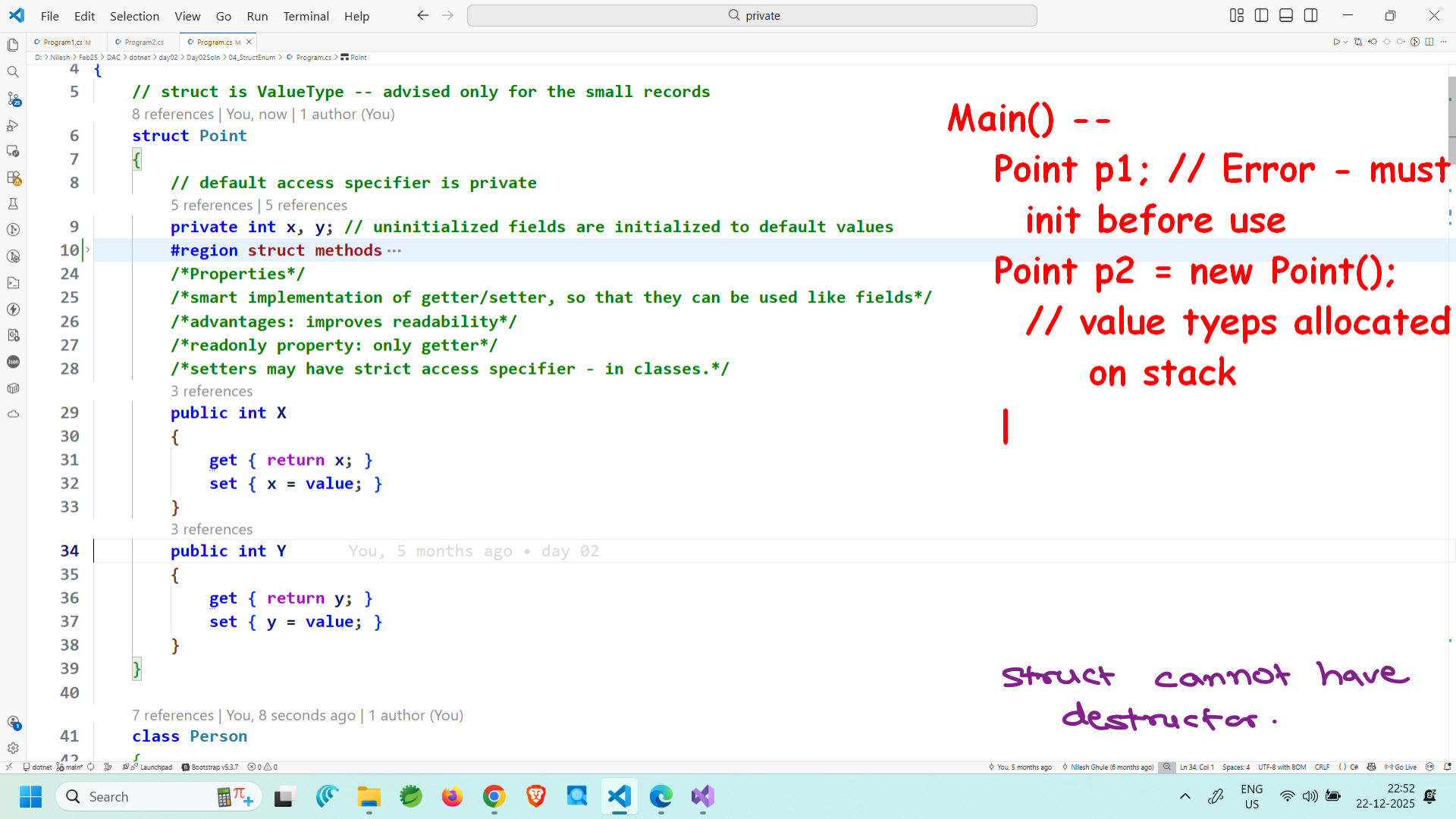
when project has multiple classes with Main().

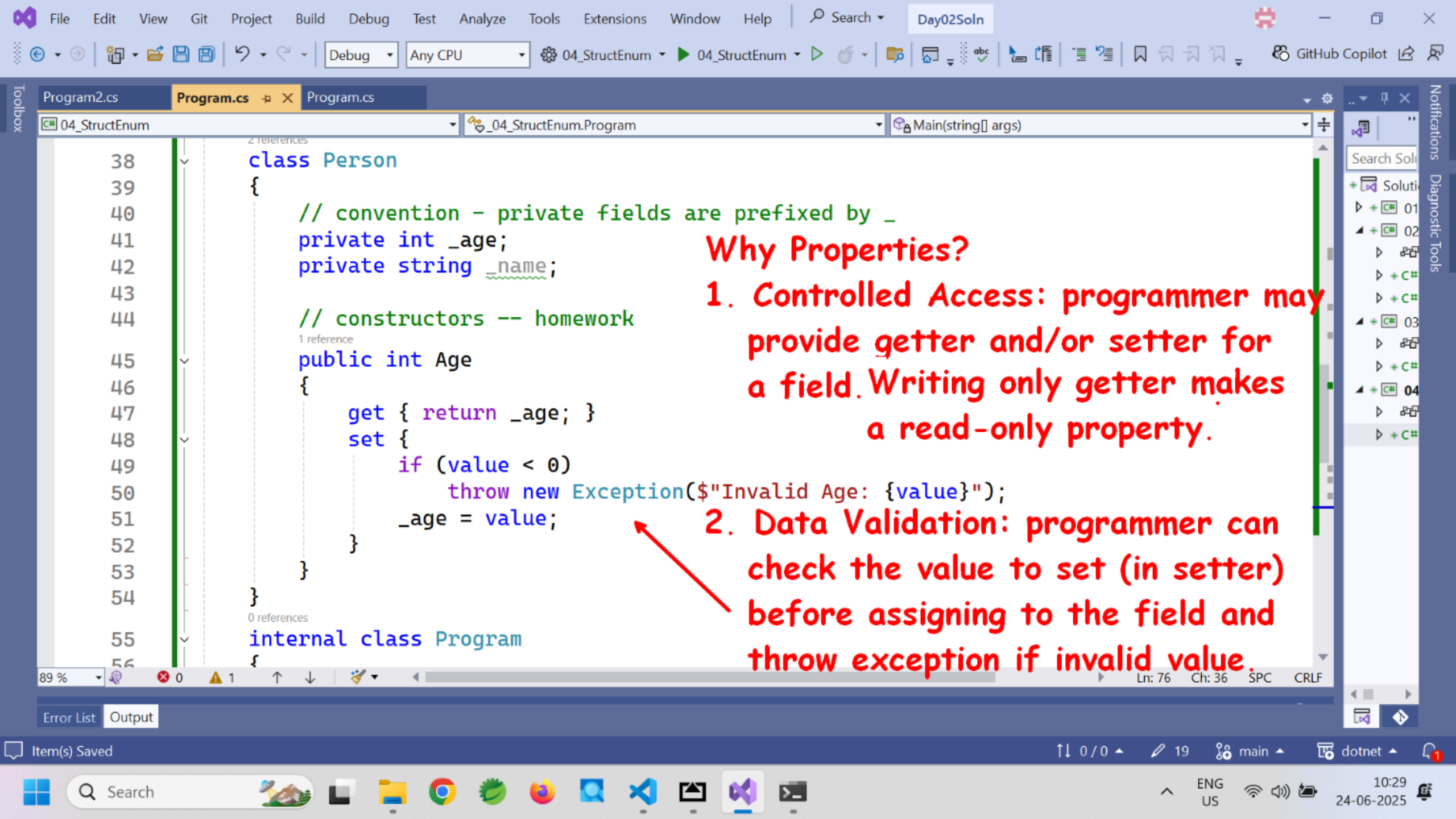
- Can select class to run in project settings i.e. csc cmdline options.

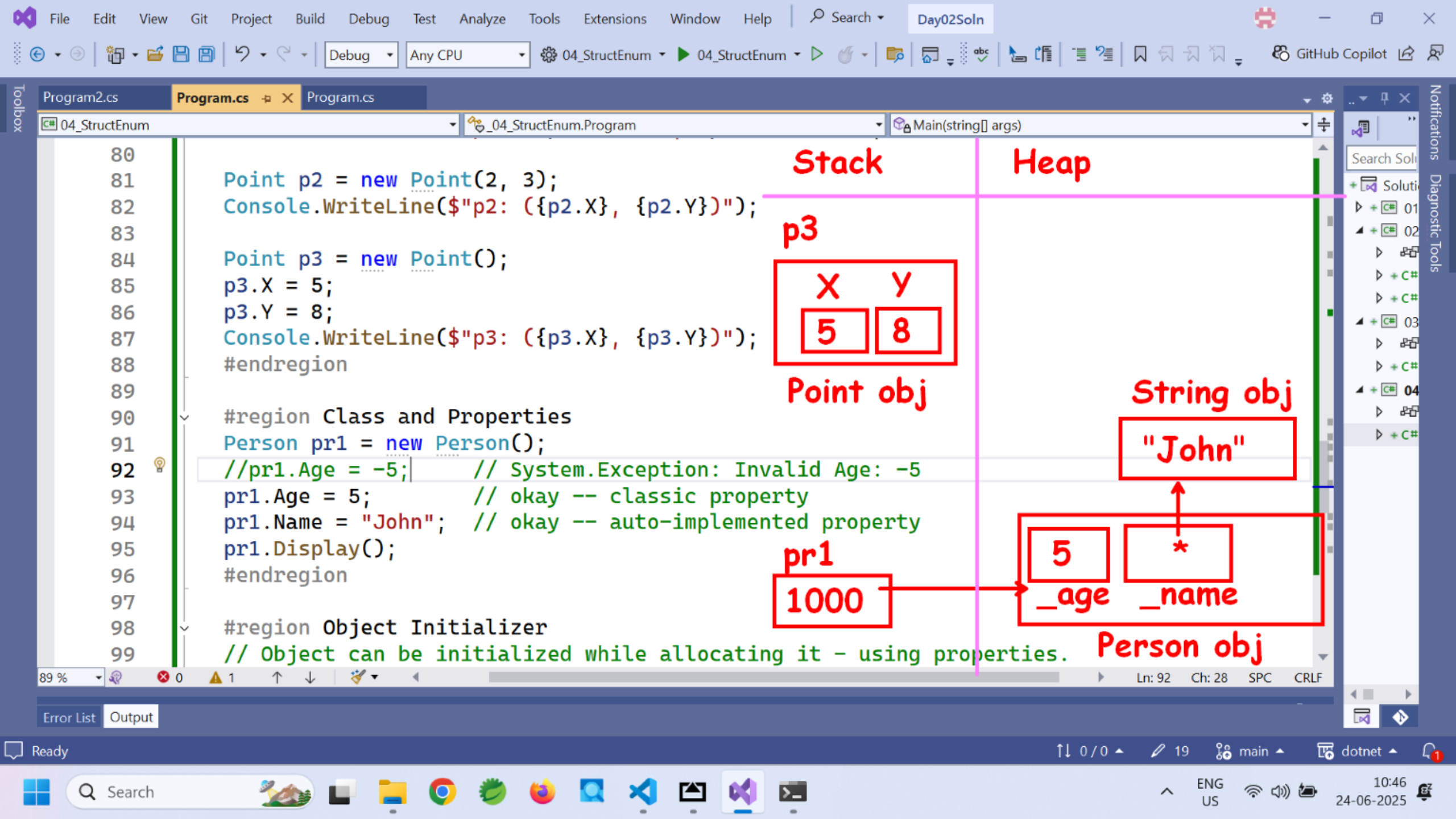

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Demos
Debug
Any CPU
02_Namespaces
GitHub Copilot
02_Namespaces*
02_Namespaces
Program2.cs
Program1.cs
Program.cs
Search properties
Application
General
Win32 Resources
Dependencies
Global Usings
Build
Package
Code Analysis
Debug
Resources
Application
General
Output type
Specifies the type of application to build.
Console Application
Target framework
Specifies the version of .NET that the application targets. This option can have different values depending on which versions of .NET are installed on your computer.
.NET 8.0
Install other frameworks
Target OS
Specifies the operating system that this project will target.
(None)
Startup object
Defines the entry point to be called when the application loads. Generally this is set either to the main form in your application or to the 'Main' procedure that should run when the application starts. Class libraries do not define an entry point.
_02_Namespaces.Program2
(Not set)
_02_Namespaces.Program1
_02_Namespaces.Program2
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Day02Demos' (2 of 2 projects)
01_HelloWorld
Dependencies
Program.cs
02_Namespaces
Dependencies
Program1.cs
Program2.cs
go to project properties
if multiple classes contains Main(), which one to execute?
Internally sets, C# /main flag.
Data Tools Operations
Error List
Output
Solution Explorer
Git Changes
Ready
main
dotnet
Search
ENG US
22:26
22-12-2025

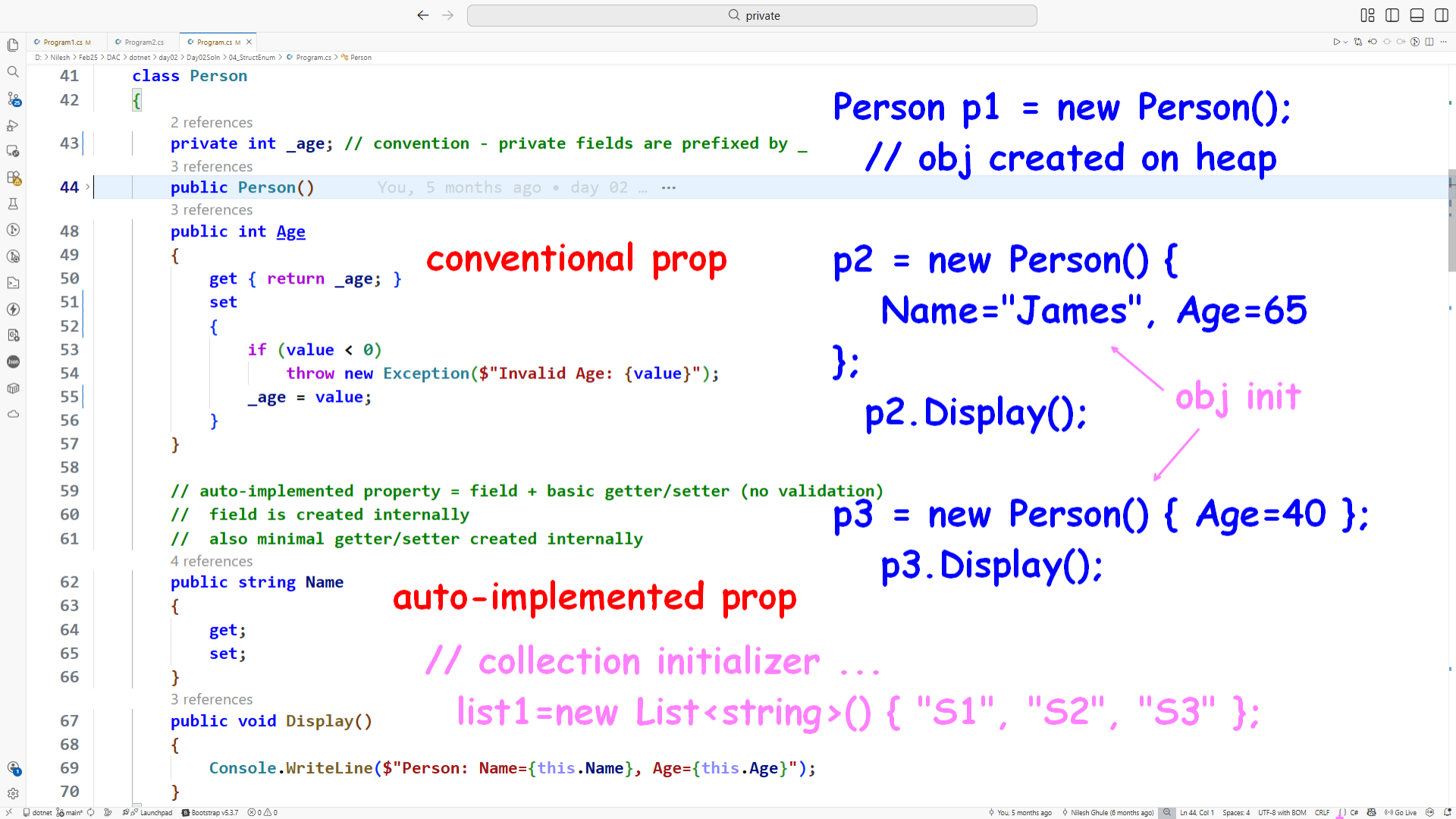

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Demos
Debug
Any CPU
03_DataTypes
GitHub Copilot
Program.cs
Program2.cs
02_Namespaces
02_Namespaces
Program1.cs
Program.cs
03_DataTypes
_03_DataTypes.Program
Main(string[] args)
0 references
static void Main(string[] args)
{
#region Type conversion using Parse()
string str = "123";
int num1 = int.Parse(str);
Console.WriteLine($"num1 = {num1}");
// int is converted to Int32 struct
// Parse() is static method in struct
// throw System.FormatException if input string is invalid
#endregion
#region Type conversion using Convert class
int num2 = Convert.ToInt32(str);
Console.WriteLine($"num2 = {num2}");
// Convert class has plenty of static methods for converting
// any type to other type (commonly used for primitive type conv)
// throw System.FormatException if input string is invalid
#endregion
}
}
}
104 %
No issues found
Ln: 17
Ch: 13
SPC
CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Day02Demos' (3 of 3 projects)
01_HelloWorld
Dependencies
Program.cs
02_Namespaces
Dependencies
Program1.cs
Program2.cs
03_DataTypes
Dependencies
Program.cs
Data Tools Operations
Error List
Output
Item(s) Saved
main
dotnet
Search
ENG
US
22:47
22-12-2025

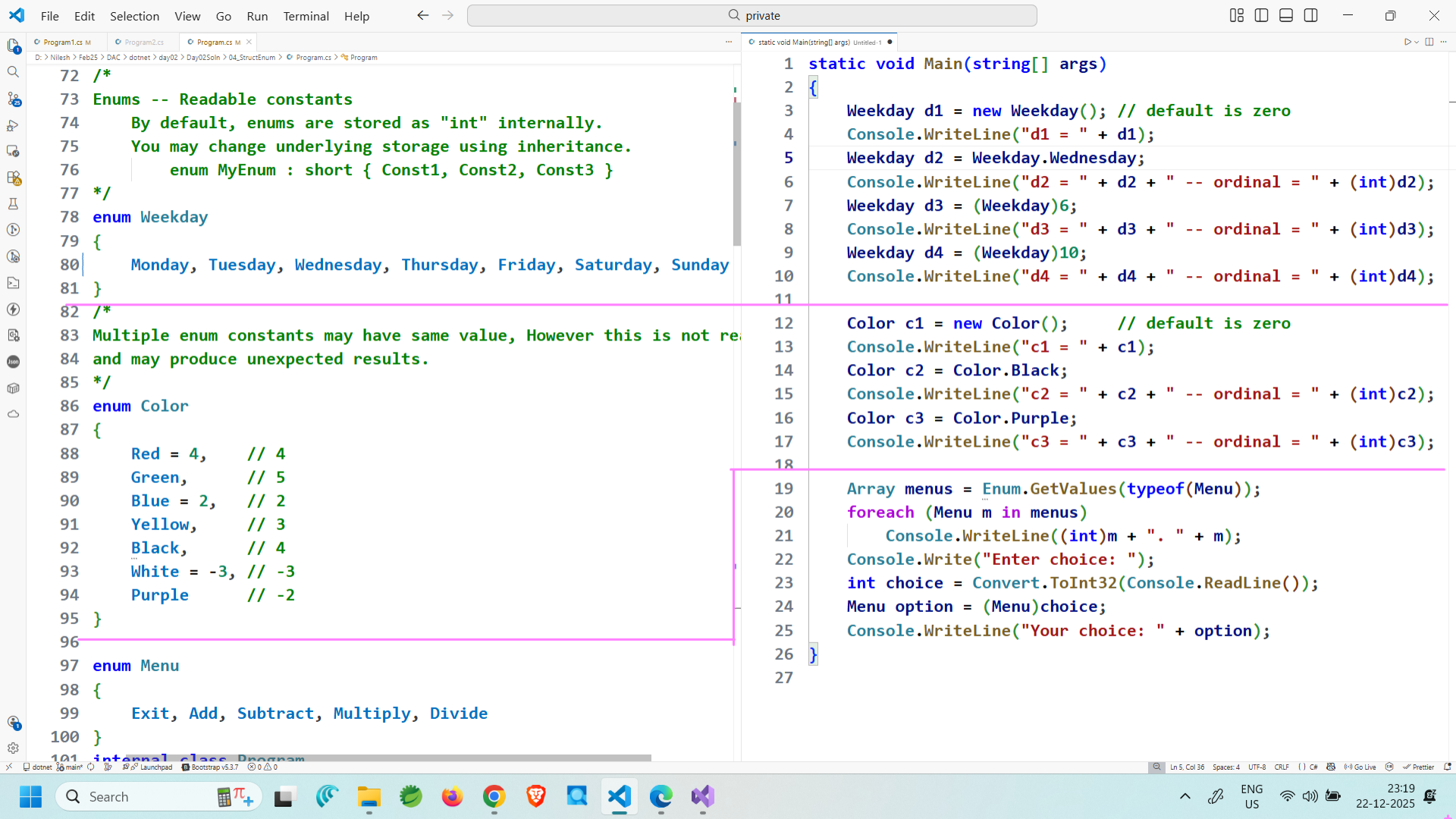

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
04_StructEnum
04_StructEnum
GitHub Copilot
Program2.cs
Program.cs
Program.cs
03_DataTypes
_03_DataTypes.Program
Main(string[] args)
0 references
static void Main(string[] args)
{
unchecked // default behaviour
{
// short range: -32768 to +32767
short num1 = 32767;
num1++; // overflow --> roll to opposite sign
Console.WriteLine($"num1 = {num1}");
}
checked //check for arithmetic overflow (data type range)
{ // and throw execption if overflow
short num2 = 32767;
num2++; // System.OverflowException: Arithmetic operation resu
Console.WriteLine($"num2 = {num2}");
}
}
}
}
In C# all exceptions are unchecked excpetions (compared to java).
"checked" & "unchecked" keyword are not exception types - they ar for overflow check.
Solution Explorer
Search Solution Explorer (C
Solution 'Day02Soln' (4 of 4
01_HelloWorld
02_Namespaces
Dependencies
Program1.cs
Program2.cs
03_DataTypes
Dependencies
Program.cs
04_StructEnum
Dependencies
Program.cs
89 %
No issues found
Ln: 29
Ch: 10
SPC
CRLF
Error List
Output
Solution Explorer
Git Changes
Ready
0 / 0
19
main
dotnet
Search
ENG
US
10:18
24-06-2025

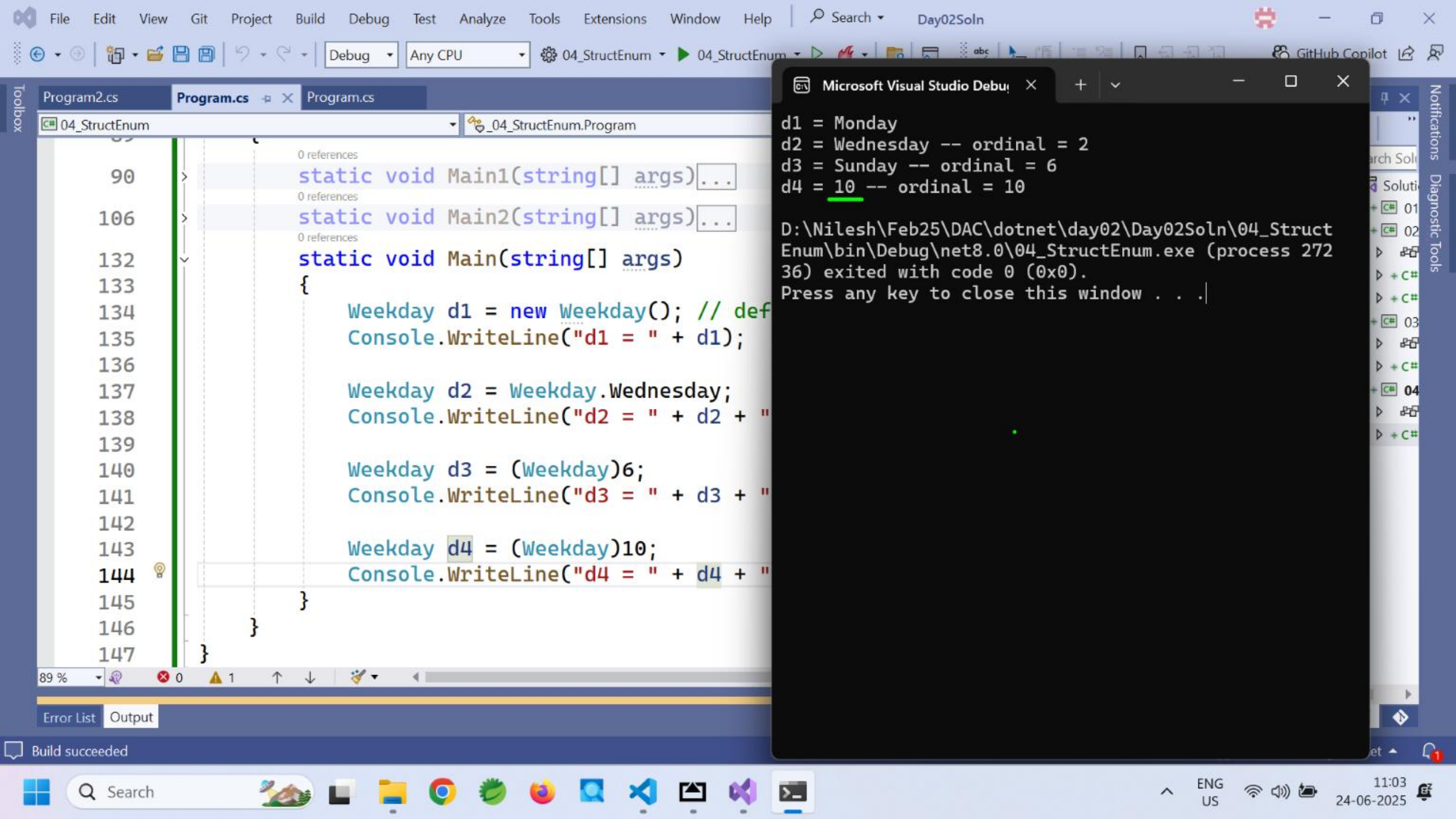

File Edit Selection View Go Run Terminal Help
private
Program1.cs M
Program2.cs
Program.cs M
D: > Nilesh > Feb25 > DAC > dotnet > day02 > Day02Soln > 04_StructEnum > Program.cs > Point
4 {
5 // struct is ValueType -- advised only for the small records
8 references | You, now | 1 author (You)
6 struct Point
7 {
8 // default access specifier is private
5 references | 5 references
9 private int x, y; // uninitialized fields are initialized to default values
10 #region struct methods ···
24 /*Properties*/
25 /*smart implementation of getter/setter, so that they can be used like fields*/
26 /*advantages: improves readability*/
27 /*readonly property: only getter*/
28 /*setters may have strict access specifier - in classes.*/
3 references
29 public int X
30 {
31 get { return x; }
32 set { x = value; }
33 }
3 references
34 public int Y You, 5 months ago • day 02
35 {
36 get { return y; }
37 set { y = value; }
38 }
39 }
40
7 references | You, 8 seconds ago | 1 author (You)
41 class Person
Main() --
Point p1; // Error - must init before use
Point p2 = new Point();
// value tyeps allocated on stack
struct cannot have destructor.
dotnet main* Launchpad Bootstrap v5.3.7 0 0
You, 5 months ago Nilesh Ghule (6 months ago) Ln 34, Col 1 Spaces: 4 UTF-8 with BOM CRLF C# Go Live
Search
ENG US
22:52 22-12-2025

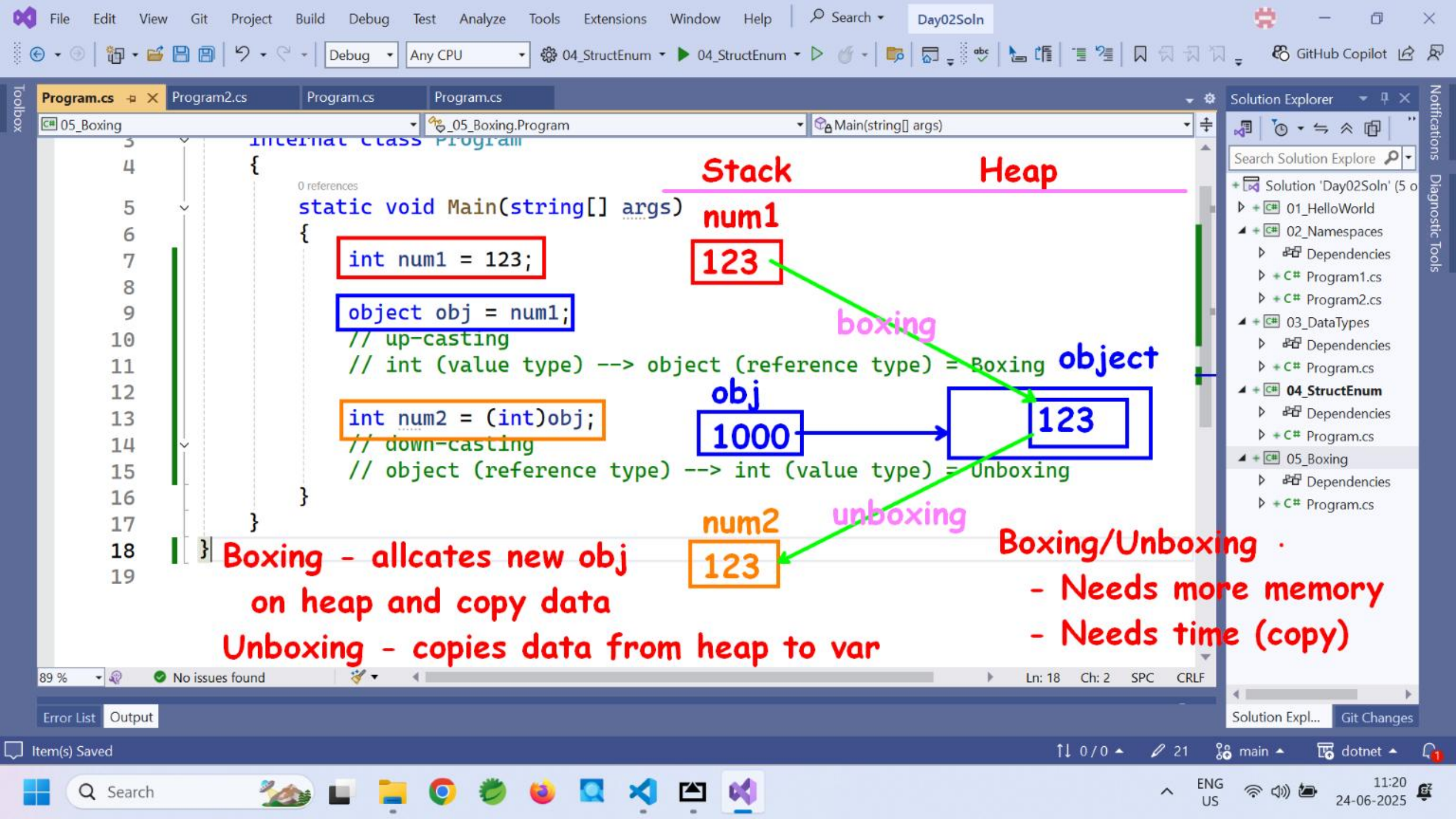

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
04_StructEnum
GitHub Copilot
Program2.cs
Program.cs
Program.cs
04_StructEnum
_04_StructEnum.Program
Main(string[] args)
2 references
class Person
{
// convention - private fields are prefixed by _
private int _age;
private string _name;
// constructors -- homework
1 reference
public int Age
{
get { return _age; }
set {
if (value < 0)
throw new Exception($"Invalid Age: {value}");
_age = value;
}
}
}
0 references
internal class Program
{
Why Properties?
1. Controlled Access: programmer may provide getter and/or setter for a field. Writing only getter makes a read-only property.
2. Data Validation: programmer can check the value to set (in setter) before assigning to the field and throw exception if invalid value.
89 %
0
1
Ln: 76
Ch: 36
SPC
CRLF
Error List
Output
Item(s) Saved
0 / 0
19
main
dotnet
Search
ENG
US
10:29
24-06-2025

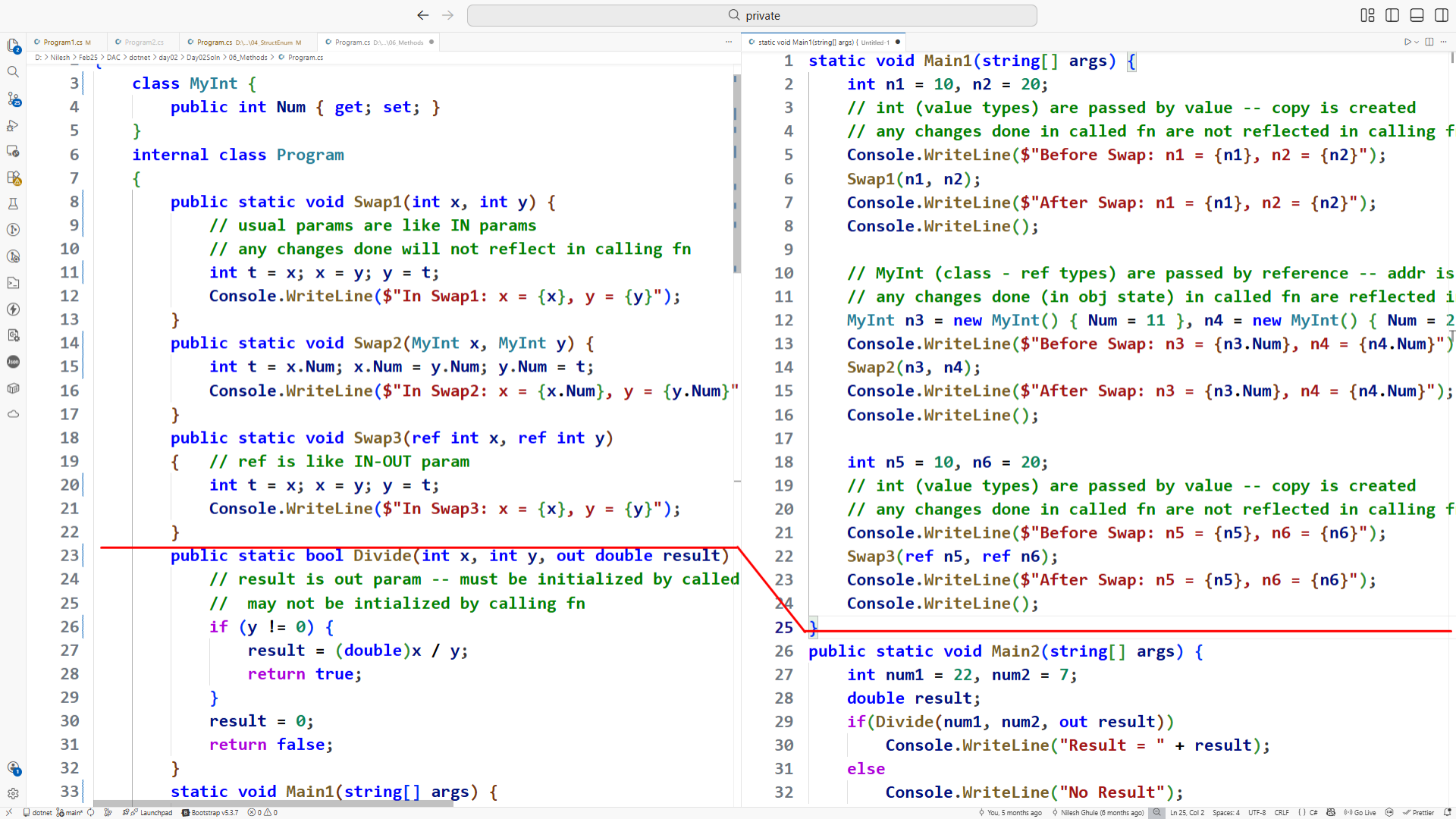

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
04_StructEnum
GitHub Copilot
Program2.cs
Program.cs
Program.cs
04_StructEnum
_04_StructEnum.Program
Main(string[] args)
80
81 Point p2 = new Point(2, 3);
82 Console.WriteLine($"p2: ({p2.X}, {p2.Y})");
83
84 Point p3 = new Point();
85 p3.X = 5;
86 p3.Y = 8;
87 Console.WriteLine($"p3: ({p3.X}, {p3.Y})");
88 #endregion
89
90 #region Class and Properties
91 Person pr1 = new Person();
92 //pr1.Age = -5; // System.Exception: Invalid Age: -5
93 pr1.Age = 5; // okay -- classic property
94 pr1.Name = "John"; // okay -- auto-implemented property
95 pr1.Display();
96 #endregion
97
98 #region Object Initializer
99 // Object can be initialized while allocating it - using properties.
Stack
Heap
p3
X
Y
5
8
Point obj
String obj
"John"
pr1
1000
5
_age
*
_name
Person obj
89 %
0
1
Ln: 92
Ch: 28
SPC
CRLF
Error List
Output
Ready
0 / 0
19
main
dotnet
Search
ENG
US
10:46
24-06-2025


private
Program1.cs M
Program2.cs
Program.cs M
D: > Nilesh > Feb25 > DAC > dotnet > day02 > Day02Soln > 04_StructEnum > Program.cs > Person
class Person
{
2 references
private int _age; // convention - private fields are prefixed by _
3 references
public Person()
You, 5 months ago • day 02 …
3 references
public int Age
{
get { return _age; }
set
{
if (value < 0)
throw new Exception($"Invalid Age: {value}");
_age = value;
}
}
// auto-implemented property = field + basic getter/setter (no validation)
// field is created internally
// also minimal getter/setter created internally
4 references
public string Name
{
get;
set;
}
3 references
public void Display()
{
Console.WriteLine($"Person: Name={this.Name}, Age={this.Age}");
}
conventional prop
auto-implemented prop
Person p1 = new Person();
// obj created on heap
p2 = new Person() {
Name="James", Age=65
};
p2.Display();
obj init
p3 = new Person() { Age=40 };
p3.Display();
// collection initializer ...
list1=new List<string>() { "S1", "S2", "S3" };


File Edit Selection View Go Run Terminal Help
private
Program1.cs M
Program2.cs
Program.cs M
D: > Nilesh > Feb25 > DAC > dotnet > day02 > Day02Soln > 04_StructEnum > Program.cs > Program
/*
Enums -- Readable constants
By default, enums are stored as "int" internally.
You may change underlying storage using inheritance.
enum MyEnum : short { Const1, Const2, Const3 }
*/
enum Weekday
{
Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
}
/*
Multiple enum constants may have same value, However this is not re
and may produce unexpected results.
*/
enum Color
{
Red = 4, // 4
Green, // 5
Blue = 2, // 2
Yellow, // 3
Black, // 4
White = -3, // -3
Purple // -2
}
enum Menu
{
Exit, Add, Subtract, Multiply, Divide
}
internal class Program
static void Main(string[] args) Untitled-1
static void Main(string[] args)
{
Weekday d1 = new Weekday(); // default is zero
Console.WriteLine("d1 = " + d1);
Weekday d2 = Weekday.Wednesday;
Console.WriteLine("d2 = " + d2 + " -- ordinal = " + (int)d2);
Weekday d3 = (Weekday)6;
Console.WriteLine("d3 = " + d3 + " -- ordinal = " + (int)d3);
Weekday d4 = (Weekday)10;
Console.WriteLine("d4 = " + d4 + " -- ordinal = " + (int)d4);
Color c1 = new Color(); // default is zero
Console.WriteLine("c1 = " + c1);
Color c2 = Color.Black;
Console.WriteLine("c2 = " + c2 + " -- ordinal = " + (int)c2);
Color c3 = Color.Purple;
Console.WriteLine("c3 = " + c3 + " -- ordinal = " + (int)c3);
Array menus = Enum.GetValues(typeof(Menu));
foreach (Menu m in menus)
Console.WriteLine((int)m + ". " + m);
Console.Write("Enter choice: ");
int choice = Convert.ToInt32(Console.ReadLine());
Menu option = (Menu)choice;
Console.WriteLine("Your choice: " + option);
}
Search
ENG US
23:19 22-12-2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
04_StructEnum
GitHub Copilot
Program2.cs
Program.cs
Program.cs
04_StructEnum
_04_StructEnum.Program
0 references
static void Main1(string[] args)...
0 references
static void Main2(string[] args)...
0 references
static void Main(string[] args)
{
Weekday d1 = new Weekday(); // def
Console.WriteLine("d1 = " + d1);
Weekday d2 = Weekday.Wednesday;
Console.WriteLine("d2 = " + d2 + "
Weekday d3 = (Weekday)6;
Console.WriteLine("d3 = " + d3 + "
Weekday d4 = (Weekday)10;
Console.WriteLine("d4 = " + d4 + "
}
}
}
89 %
0
1
Error List
Output
Build succeeded
Microsoft Visual Studio Debu
d1 = Monday
d2 = Wednesday -- ordinal = 2
d3 = Sunday -- ordinal = 6
d4 = 10 -- ordinal = 10
D:\Nilesh\Feb25\DAC\dotnet\day02\Day02Soln\04_StructEnum\bin\Debug\net8.0\04_StructEnum.exe (process 27236) exited with code 0 (0x0).
Press any key to close this window . . .
Notifications
Diagnostic Tools
Search
ENG
US
11:03
24-06-2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Day02Soln
Debug
Any CPU
04_StructEnum
GitHub Copilot
Program.cs
Program2.cs
Program.cs
Program.cs
05_Boxing
_05_Boxing.Program
Main(string[] args)
internal class Program
{
0 references
static void Main(string[] args)
{
int num1 = 123;
object obj = num1;
// up-casting
// int (value type) --> object (reference type) = Boxing
int num2 = (int)obj;
// down-casting
// object (reference type) --> int (value type) = Unboxing
}
}
}
Stack
Heap
num1
123
boxing
object
obj
1000
123
unboxing
num2
123
Boxing - allcates new obj
on heap and copy data
Unboxing - copies data from heap to var
Boxing/Unboxing
- Needs more memory
- Needs time (copy)
Solution Explorer
Solution 'Day02Soln' (5 o
01_HelloWorld
02_Namespaces
Dependencies
Program1.cs
Program2.cs
03_DataTypes
Dependencies
Program.cs
04_StructEnum
Dependencies
Program.cs
05_Boxing
Dependencies
Program.cs
89 %
No issues found
Ln: 18
Ch: 2
SPC
CRLF
Error List
Output
Item(s) Saved
main
dotnet
Search
ENG
US
11:20
24-06-2025


private
Program1.cs M
Program2.cs
Program.cs D:\...\04_StructEnum M
Program.cs D:\...\06_Methods
D: > Nilesh > Feb25 > DAC > dotnet > day02 > Day02Soln > 06_Methods > Program.cs
class MyInt {
public int Num { get; set; }
}
internal class Program
{
public static void Swap1(int x, int y) {
// usual params are like IN params
// any changes done will not reflect in calling fn
int t = x; x = y; y = t;
Console.WriteLine($"In Swap1: x = {x}, y = {y}");
}
public static void Swap2(MyInt x, MyInt y) {
int t = x.Num; x.Num = y.Num; y.Num = t;
Console.WriteLine($"In Swap2: x = {x.Num}, y = {y.Num}"
}
public static void Swap3(ref int x, ref int y)
{ // ref is like IN-OUT param
int t = x; x = y; y = t;
Console.WriteLine($"In Swap3: x = {x}, y = {y}");
}
public static bool Divide(int x, int y, out double result)
// result is out param -- must be initialized by called
// may not be intialized by calling fn
if (y != 0) {
result = (double)x / y;
return true;
}
result = 0;
return false;
}
static void Main1(string[] args) {
static void Main1(string[] args) {
int n1 = 10, n2 = 20;
// int (value types) are passed by value -- copy is created
// any changes done in called fn are not reflected in calling f
Console.WriteLine($"Before Swap: n1 = {n1}, n2 = {n2}");
Swap1(n1, n2);
Console.WriteLine($"After Swap: n1 = {n1}, n2 = {n2}");
Console.WriteLine();
// MyInt (class - ref types) are passed by reference -- addr is
// any changes done (in obj state) in called fn are reflected i
MyInt n3 = new MyInt() { Num = 11 }, n4 = new MyInt() { Num = 2
Console.WriteLine($"Before Swap: n3 = {n3.Num}, n4 = {n4.Num}")
Swap2(n3, n4);
Console.WriteLine($"After Swap: n3 = {n3.Num}, n4 = {n4.Num}");
Console.WriteLine();
int n5 = 10, n6 = 20;
// int (value types) are passed by value -- copy is created
// any changes done in called fn are not reflected in calling f
Console.WriteLine($"Before Swap: n5 = {n5}, n6 = {n6}");
Swap3(ref n5, ref n6);
Console.WriteLine($"After Swap: n5 = {n5}, n6 = {n6}");
Console.WriteLine();
}
public static void Main2(string[] args) {
int num1 = 22, num2 = 7;
double result;
if(Divide(num1, num2, out result))
Console.WriteLine("Result = " + result);
else
Console.WriteLine("No Result");

```
67
68          static void Main(string[] args)
69          {
70              string greet = "Hello";
71              // local methods (method inside another method)
72              void PrintInfo(string name, int age, string addr="anywhere", string email="unknown")
73              {
74                  // local methods can capture the variables of the outer method - Closure
75                  // static local method cannot capture the variables of outer method
76                  Console.WriteLine($"{greet} -> Name: {name}, Age: {age}, Addr: {addr}, Email: {email}");
77              }
78              // call the method -- using positional args (seq of args compulsory)
79              PrintInfo("James Bond", 65, "London", "james@bond.com");
80              // call the method -- using named args (seq is not compulsory)
81              PrintInfo(age: 65, name: "James Bond", email: "james@bond.com", addr: "London");
82              // call the method -- passing optional params is not compusory
83              PrintInfo("James Bond", 65);
84          }
85      }
86  }
87
```

Object Slicing (Java/C#) -

When derived class object is assigned to base class ref, we can only access base class members through that ref.

```
class Person {
    public string name;
    // ...
}
class Student:Person {
    public int marks;
    // ...
}
```

```
Person p = new Student();
cw(p.name); // okay
cw(p.marks); // error
```

Exception - The overridden virtual members are accessed based on type of object (not the ref).

```
Student s = (Student) p;
cw(s.marks); // okay
```

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>