**C# Learning journey at Makeen Bootcamp**

🡺Learning topics

1. different between front-end developer and backend-developer

Front-end developer (Client side) is responsible about the user-interface and how the user is going to use it.

For example:

Login form 🡺 username:

🡺Password: 🡺 request to the server(back-end) to do the complete process.

**Front end needs:**

-**Framework** 🡺.Net, NodeJS, React-Native (Mobile App).

-**Programming languages**🡺 to design front-end (HTML, CSS, Bootstrap) but without validation and interaction there is no life with that, so it will be more advance to add validation using JavaScript to make it easy for the server in advance, if you don’t do validation and connect directly to database to perform any operation needs to make request every time and more request means will attack the server and make the server slow while response.

**Back-end developer** (Server-side) is responsible about the user needs and the purpose of the website, application and its more in doing the main functions.

Need to have basics of programming languages, Advanced Programming, problem solving, Database Fundamental, Backend Framework, DBMS MSQL Server.

Programming languages used in .NET (Visual basic, F#, C#, J# (now is disconnected and not used any more)

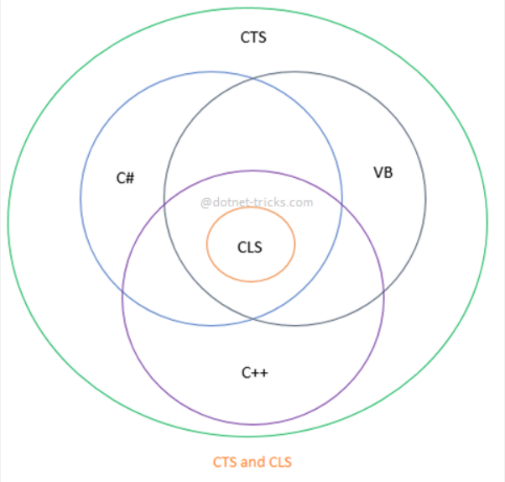
.NET: (one platform that is multifunctional means it producing product as web project, Desktop project and mobile Application), .NET is known for its high-performance and scalability, making it an ideal technology

.NET is a popular technology stack for web development, it is widely used by Full Stack Developers because it is a Cross-platform development, large developer community, Wide Range of usage (Web Development - Desktop Applications - Mobile Applications - game Development - Augmented Reality), High level of Security.

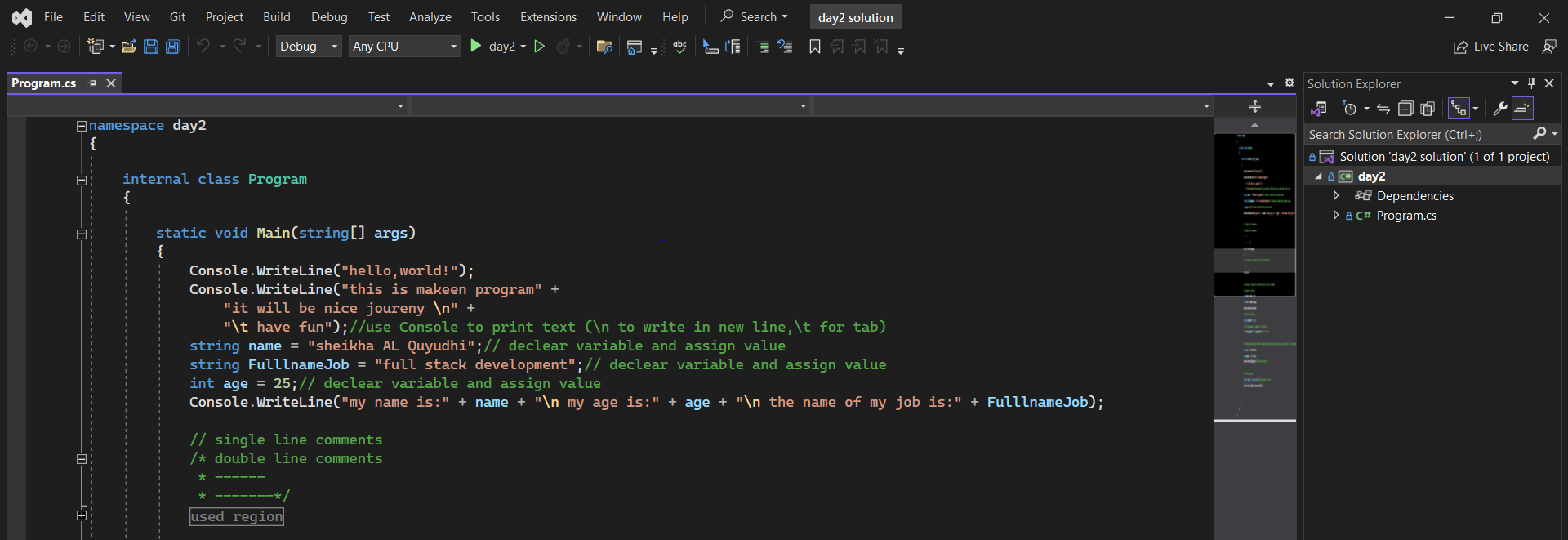
C#: is programming language that provides many benefits for developer whether you are building desktop applications, web applications or mobile app ,C# can help you writing efficient , maintainable and reliable code.

Common Type System :

The CTS defines a set of rules and guidelines for how types are defined, used, and represented in the .NET runtime environment. It provides a standard way to represent and manipulate data types, ensuring that code written in different programming languages can interoperate seamlessly within the .NET framework.



Create Blank Solution and practice codes.



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

- use (Console.WrileLine)to print text (\n to write in new line, \t for tab)

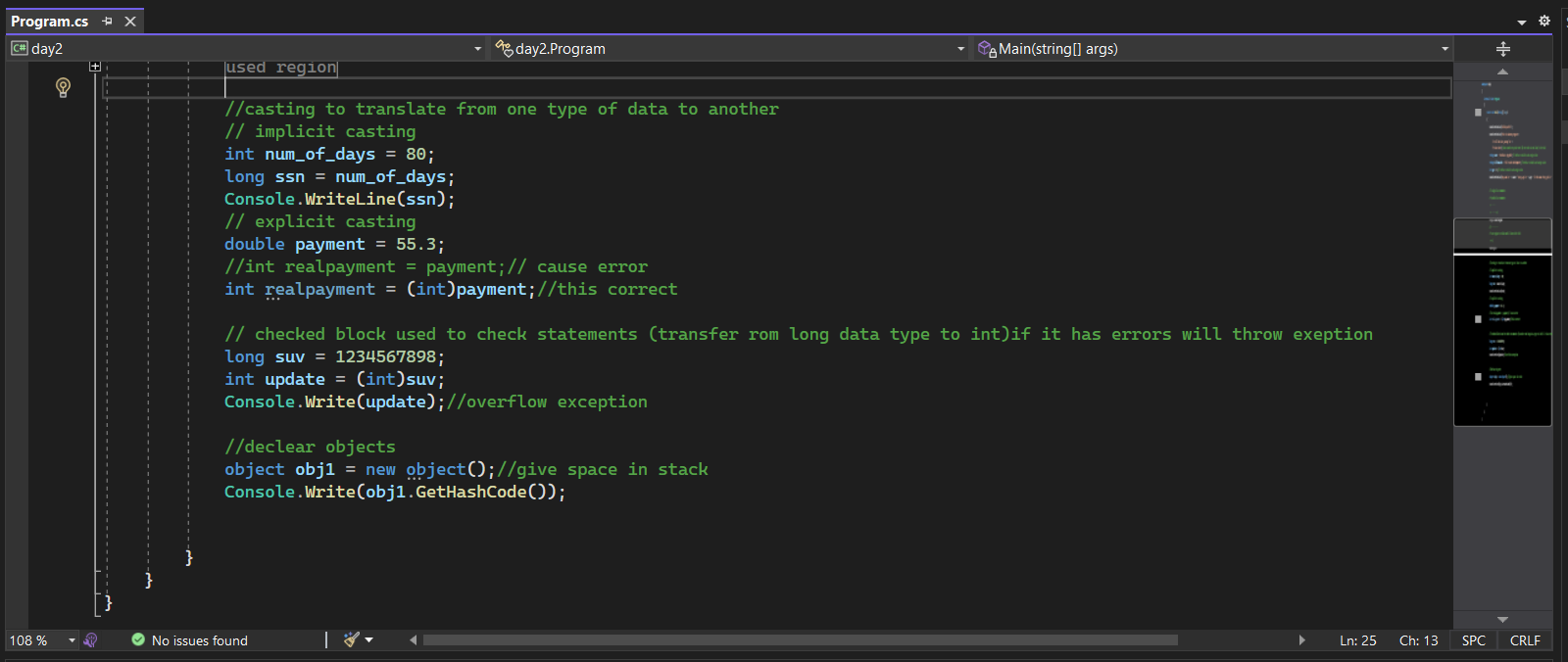
-use (Console.ReadLine) to read input from user.

- declare variable and assign value.

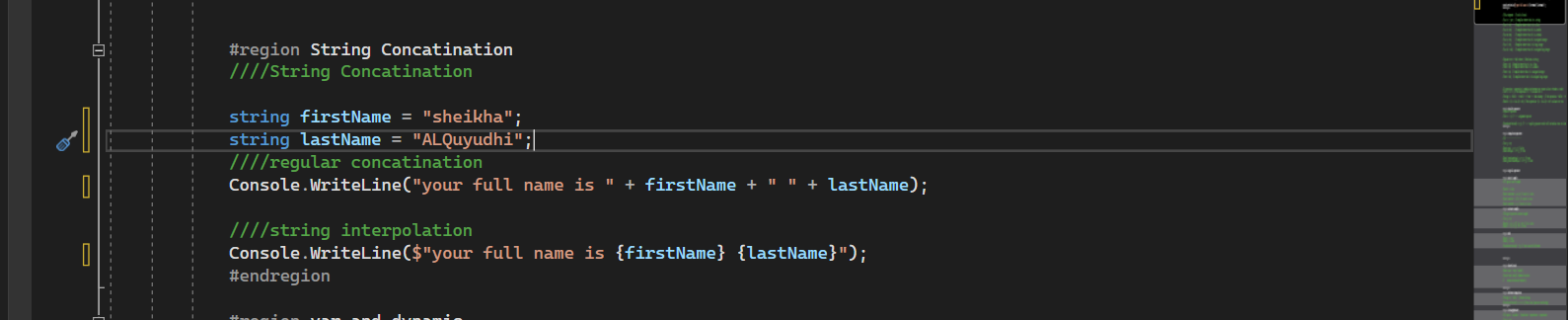
- // single line comments

- /\* \*/ multiline comments

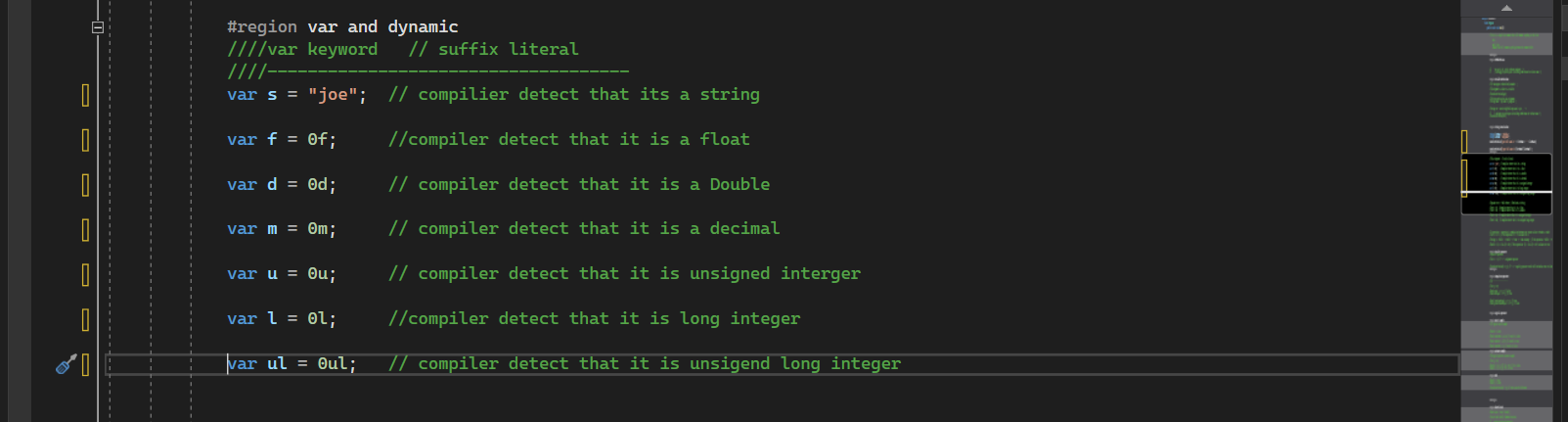
Example of using Casting (Implicit , Explicit),declear object.



Example for String Concatenation.



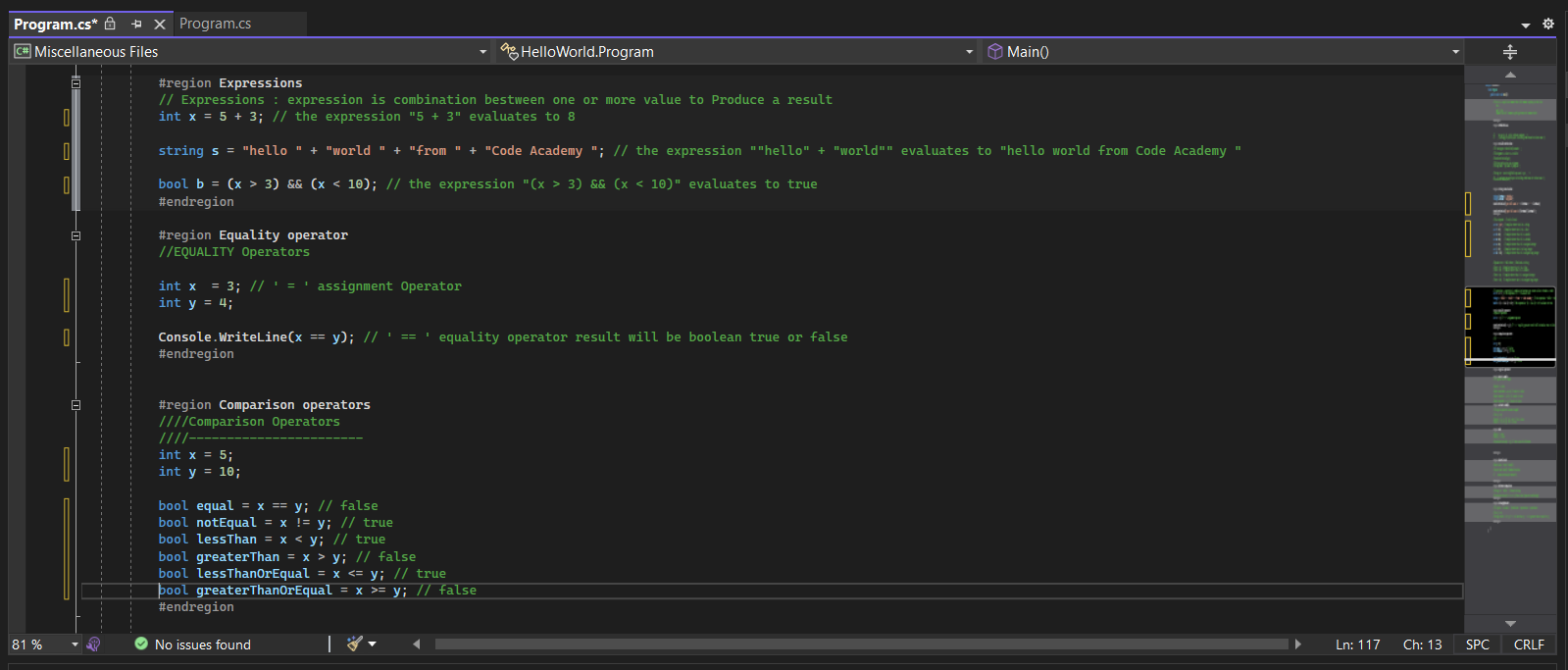
Var& Dynamic example.



Example for (Expressions, equality ,comparison operator)

Comparison Operators comparison operators are used to compare two values and determine their relationship to each other. These operators return a Boolean value of true or false depending on whether the comparison is true or false.

Equality Operators ‘ == ’: In your Program you will need to detect if something is valid under certain Condition or not by saying that if User value is equal to something you will do certain action and we will dive deep in this in upcoming lectures but for now let’s Learn Basics with below example .

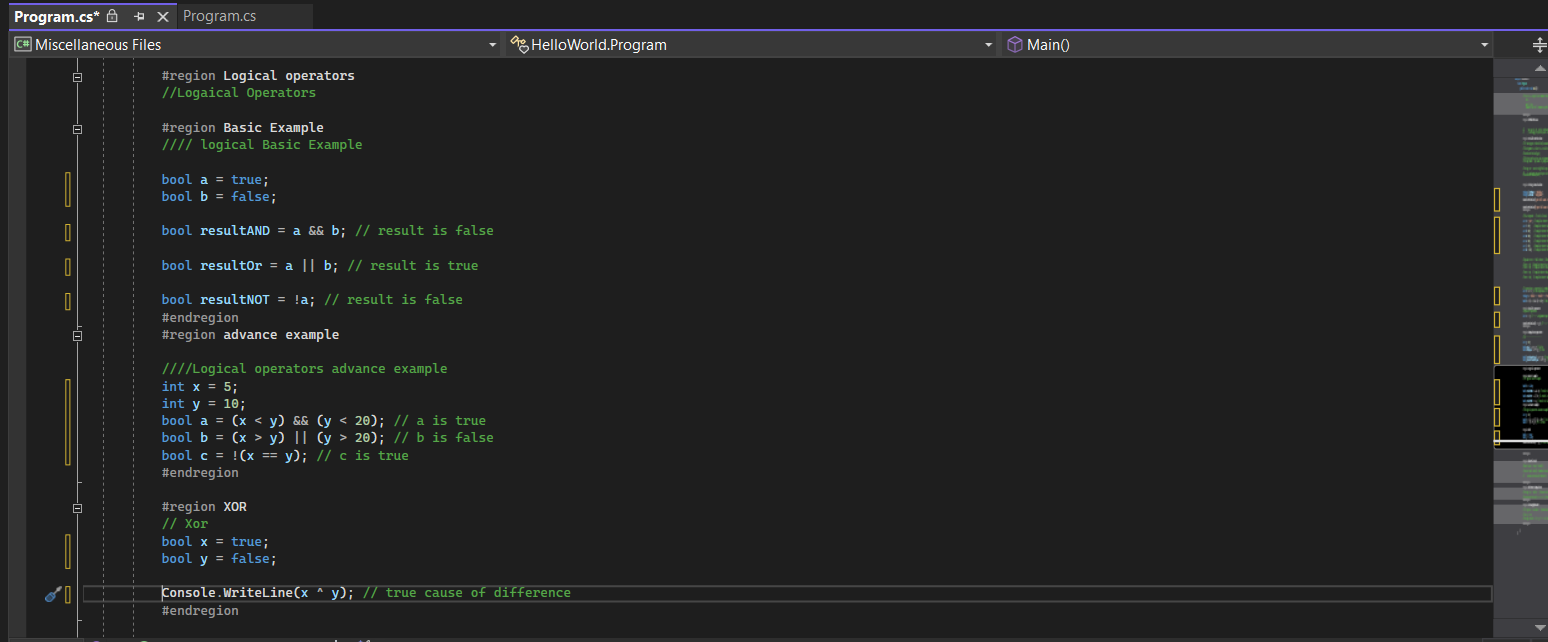


Logical Operators :are used to combine multiple Boolean expressions and determine their overall truth value. There are three logical operators in C#:

1. && (logical AND): This operator returns true if both of its operands are true, and false otherwise.

2. || (logical OR): This operator returns true if at least one of its operands is true, and false otherwise.

3. ! (logical NOT): This operator negates the Boolean value of its operand. If the operand is true, it returns false, and if the operand is false, it returns true.

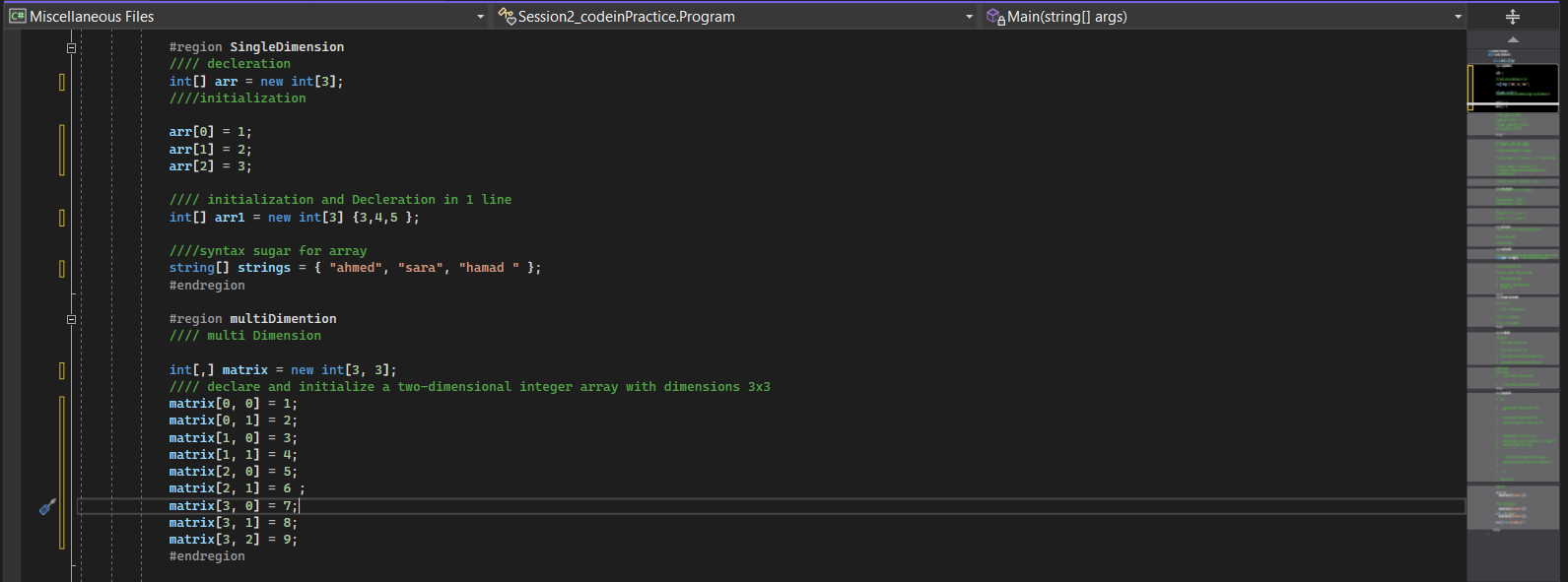


Arrays:

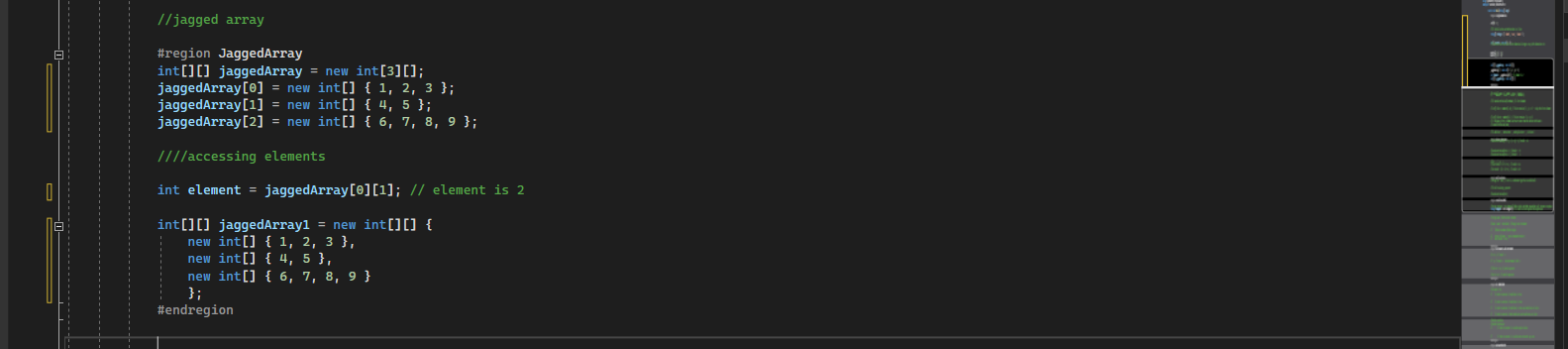
an array is a collection of elements of the same data type, stored in contiguous memory locations. Arrays are a fundamental data structure that allow for efficient storage and access to collections of data. Array is a reference type.

1-single Dimensional Array

2-Multi-Dimensional Array



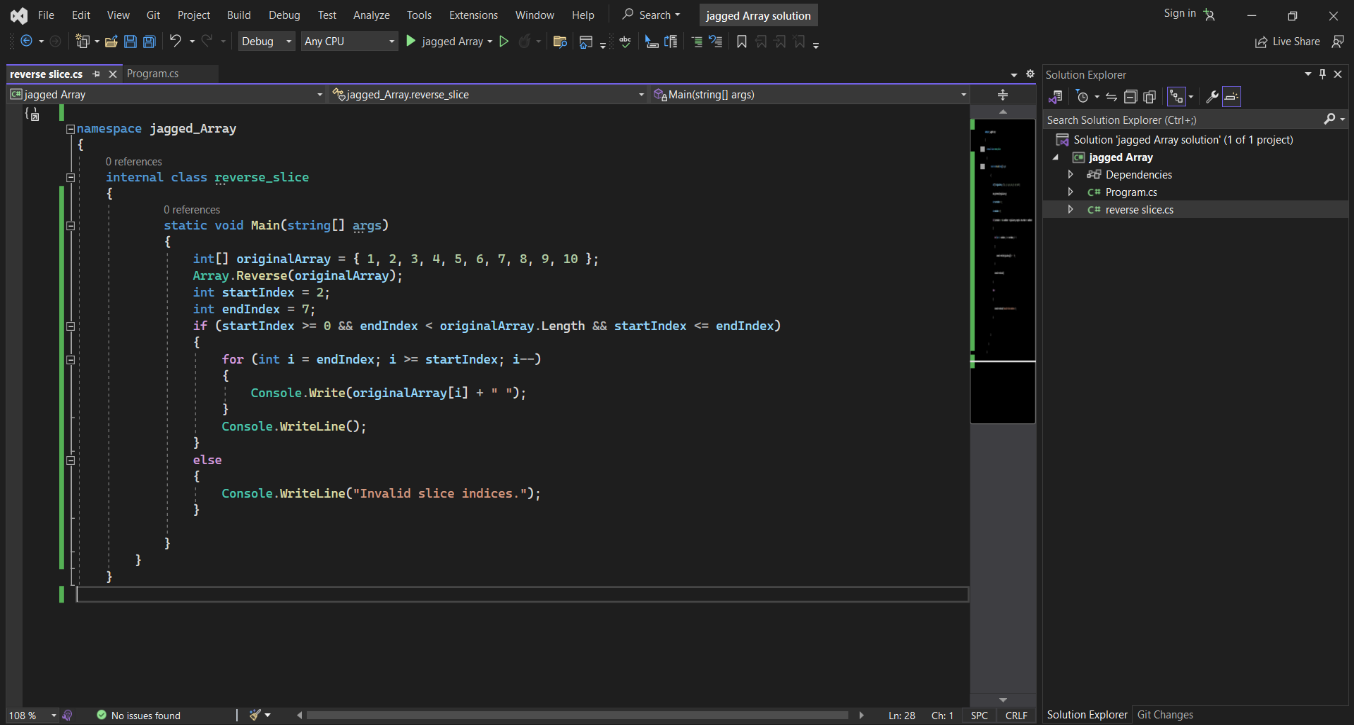
3-Jagged Array: A jagged array in C# is an array of arrays, where each element of the main array is an array itself. Unlike a multidimensional array where all rows have the same number of columns, in a jagged array, each row (array) can have a different length.



1. **reverse slicing**

In C#, reverse slicing is not directly supported as it is in some other programming languages like Python. However, you can achieve a similar result by manipulating arrays or collections manually.

For instance, suppose you have an array of integers and you want to slice it in reverse order. Here's an example demonstrating how you can achieve this using Array.

**Example**:

Conditioning

Conditions🡺If we need to create a Grading system for Students to make some operations and generate his final Grade. If statements are one of the most popular in programming its syntax is easy and can-do powerful things, in other way The if statement is the most basic form of conditioning in C#. It evaluates a condition and executes a block of code if the condition is true:

Condition: if, else if 🡺 if you want your code to validate more than one condition so if, else if is here for you, if else if works to validate that if the first expression generate False, so there another case that we can pass it through,

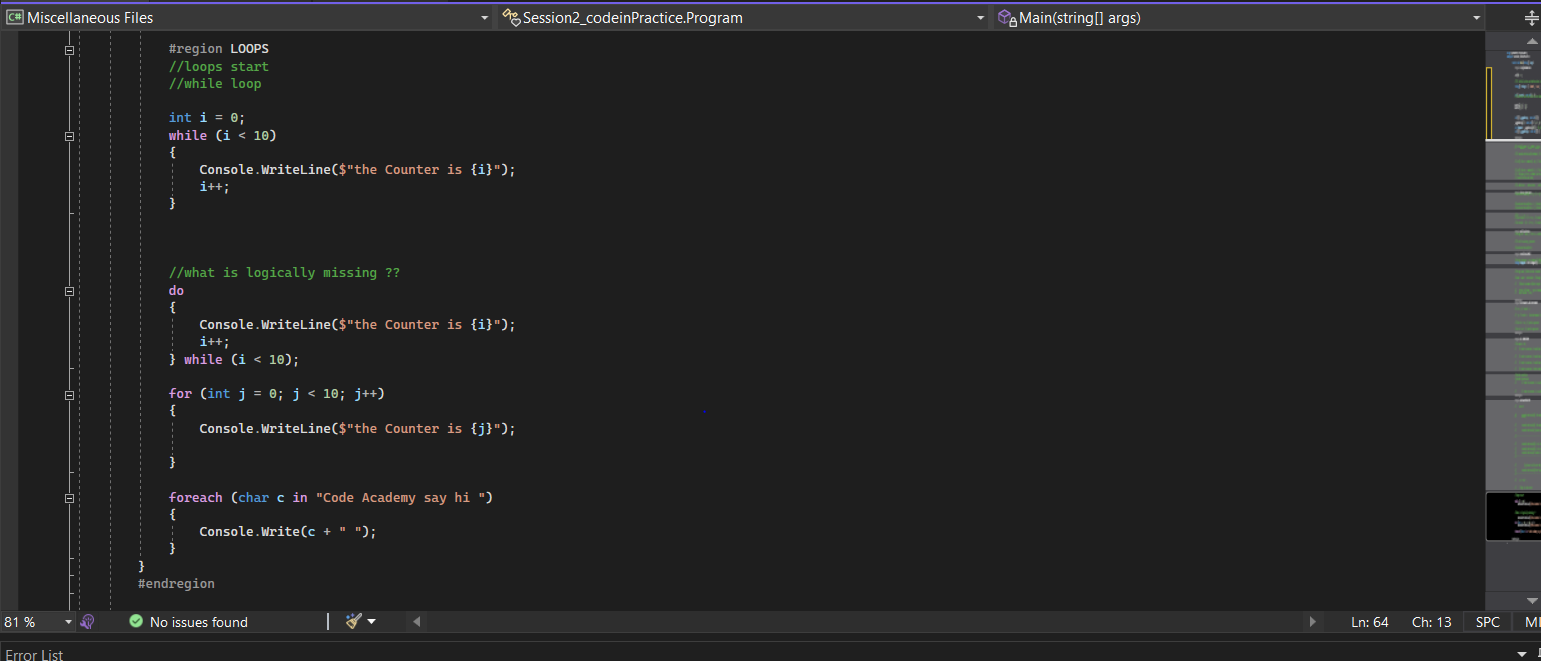
Condition Switch🡺 is Conditional expression like if but here we are deal with cases one of the most popular examples the Menu that we deal with in the ATM bank machine. Menu accept from you some option to proceed the transaction where you want to make a withdraw, debit, Transfer. Switch should have break to exit the condition

Break keyword🡺 is used to exit the loop as switch internally is work like iteration it check if we didn’t satisfied the Condition (false), it will go for the next condition, but when it found the True Value it will turn all the next condition to true so we need Break, to exit the switch after getting our value true.

Iterations & collecting all together

In programming, "iteration" refers to the process of repeatedly executing a block of code or statement a certain number of times or until a condition is met.

1. While Loop: This loop executes the code block repeatedly as long as the condition is true.
2. Do-while: This is similar to the while loop, but it executes the code block at least once before checking the condition.
3. for loop: This is a common loop used to iterate a fixed number of times. It has three parts: initialization, condition, and increment/decrement.
4. Foreach: This loop is used to iterate over the elements of an array, collection, or other enumerable type.



In C#, errors can be categorized into three main types:

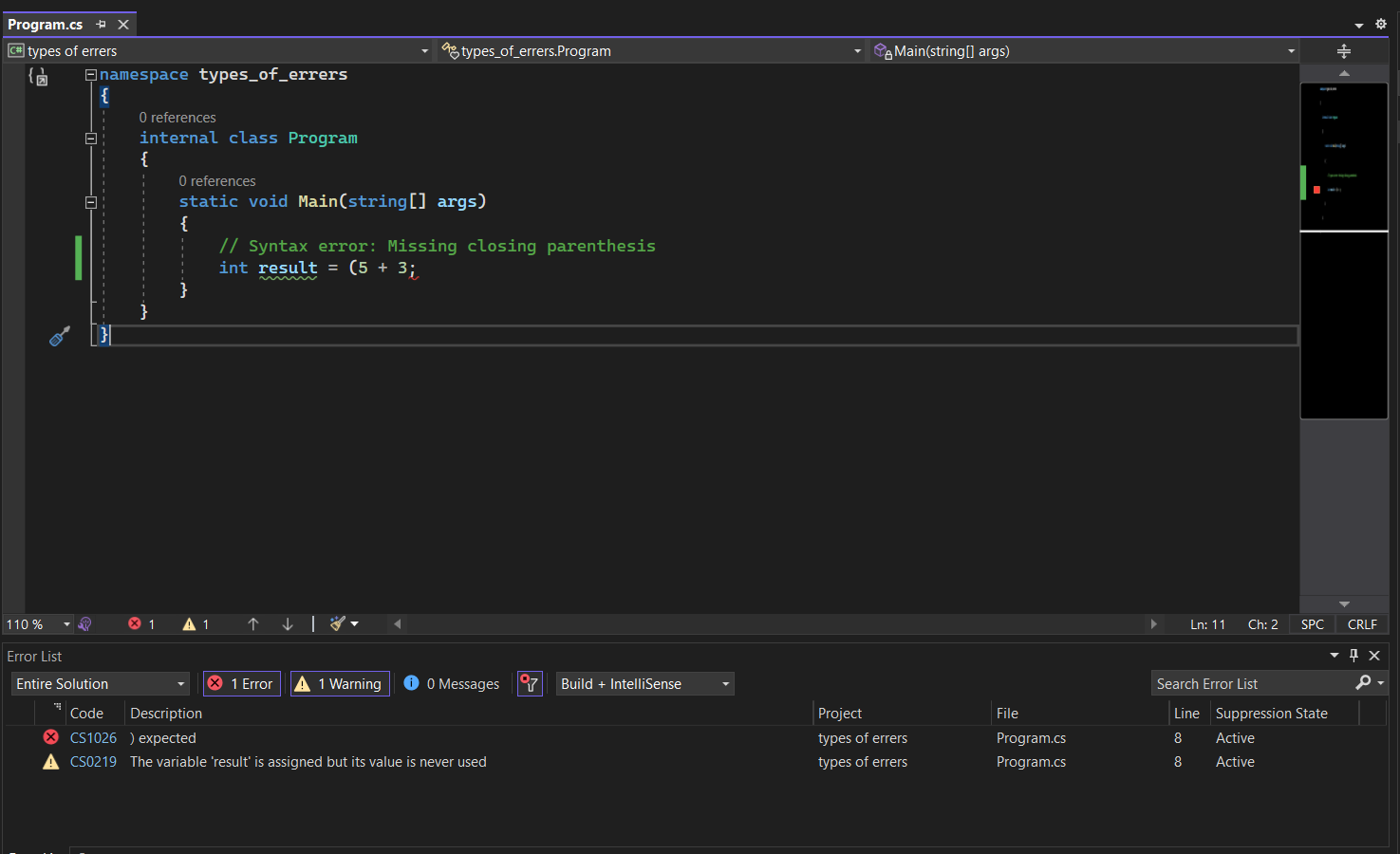
syntax errors, runtime errors, logical errors and warning errors.

while syntax errors prevent code from compiling, runtime errors occur during program execution, and logical errors produce unexpected behavior without necessarily causing the program to crash. Identifying and debugging these different types of errors require different approaches such as code inspection, using debugging tools, or handling exceptions appropriately.

**1. Syntax Errors:**

Syntax errors occur when the code violates the rules of the programming language. These errors prevent the code from being compiled or parsed correctly. Common syntax errors include missing semicolons, mismatched parentheses, misspelled keywords, etc.

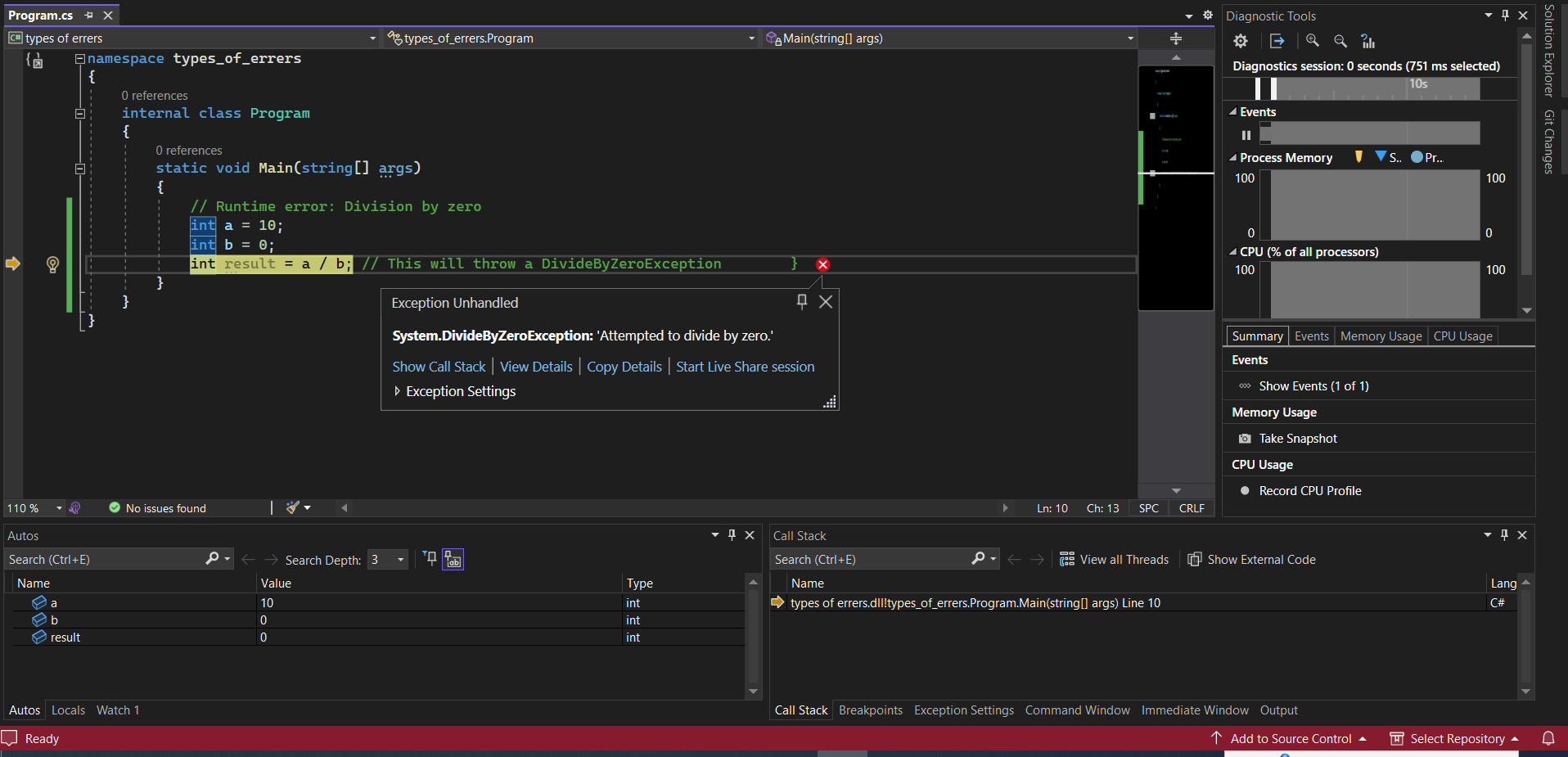
Example:



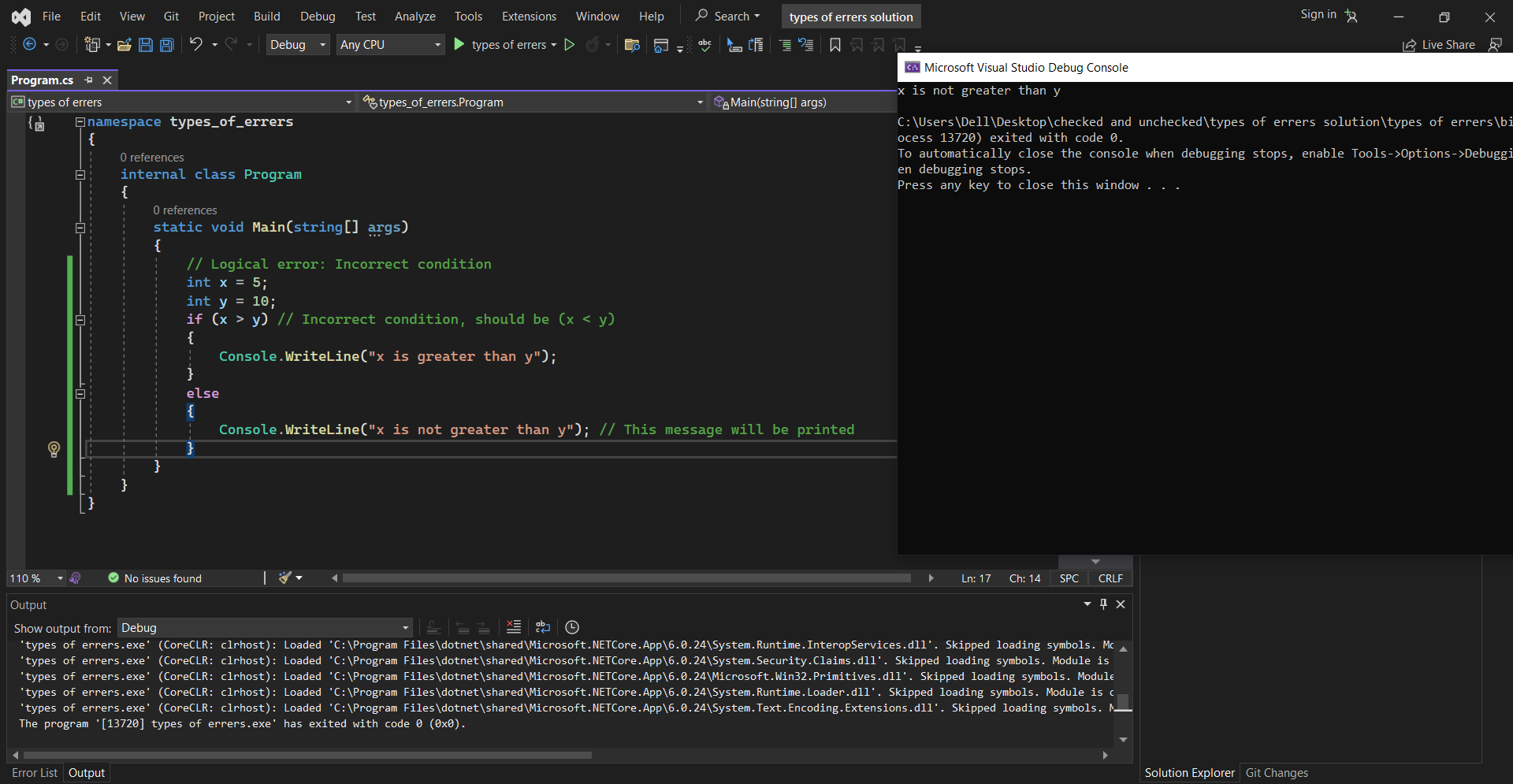
**2. Runtime Errors:**

Runtime errors occur during the execution of a program. They're also known as exceptions and may happen due to issues like invalid user input, division by zero, accessing null references, or attempting to perform unsupported operations.

Example :



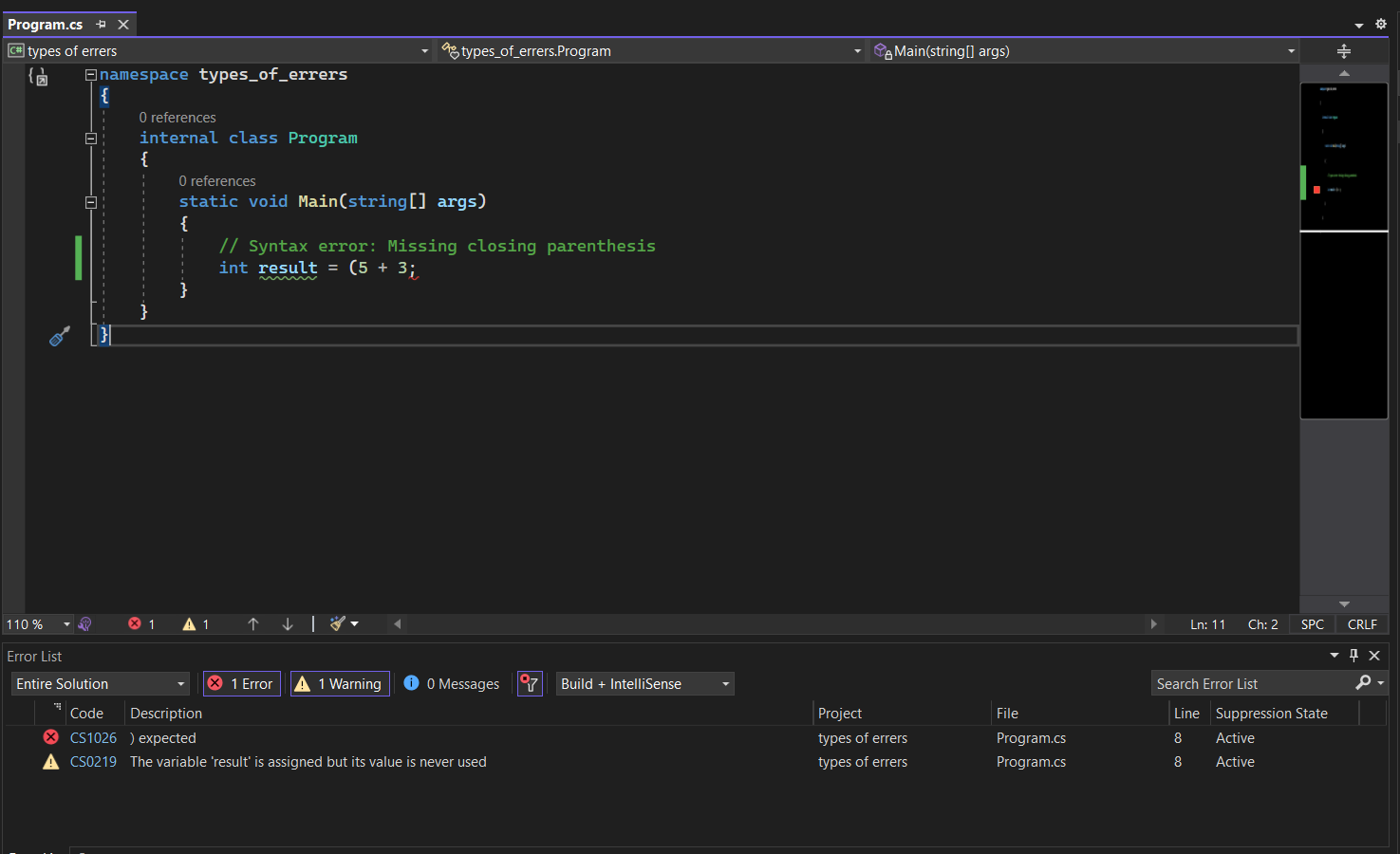
**3. Logical Errors:**

Logical errors occur when the code does not produce the expected output due to incorrect implementation or flawed logic. These errors do not cause the program to crash or throw exceptions but result in undesired behavior or incorrect results.

**4.Warning error:**

warning messages are diagnostic messages generated by the compiler to alert developers about potential issues or suspicious code constructs that might cause problems during runtime or produce unexpected behavior. Unlike errors, warnings don't prevent the compilation of code; instead, they highlight areas where improvements or corrections might be needed to ensure better code quality or prevent possible bugs.

Example:



-The variable ‘result’ is assigned but it is never used.