Enhancing and Refactoring a C# Console in Visual Studio 2022

Introduction

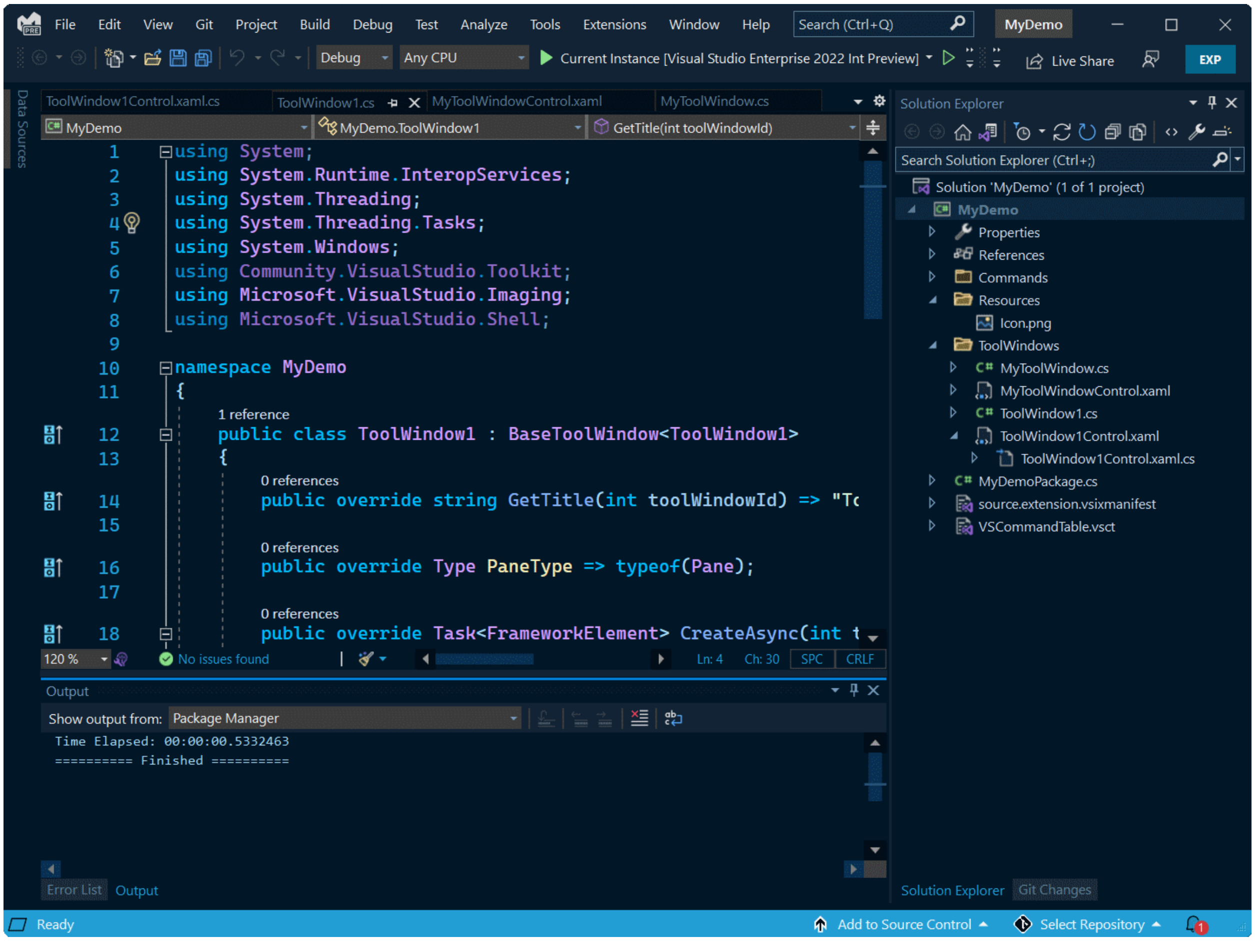
This report seeks to critically assess and then reconstruct a C# console program as availed by the tutor. Its objective is to analyze the given code and identify issues with the implementation, and apply problem-solving strategies. Thinking critically is paramount in programming for it helps in linking the ideas from start to an endpoint, sharpens the problem-solving skills, and develops best ideas. Understanding the flaws and/or shortcomings of the code helps one to fix it, thus making the program more optimized and easier to manage. This report addresses three tasks: examining the binary code, defining the issue, and recommending four large-scale alterations.

Task 1: Download and Explore the Code

The first exercise presupposes downloading of the given C# console program and studying it with the help of Visual Studio 2022. Once the code was downloaded, I then opened it in Visual Studio to see its characteristics as well as the things it could do. In order to understand the functionality of the program and the relations between its classes, methods, and variables, these aspects were investigated to illustrate how they work in the determination of the model solution for the problem.

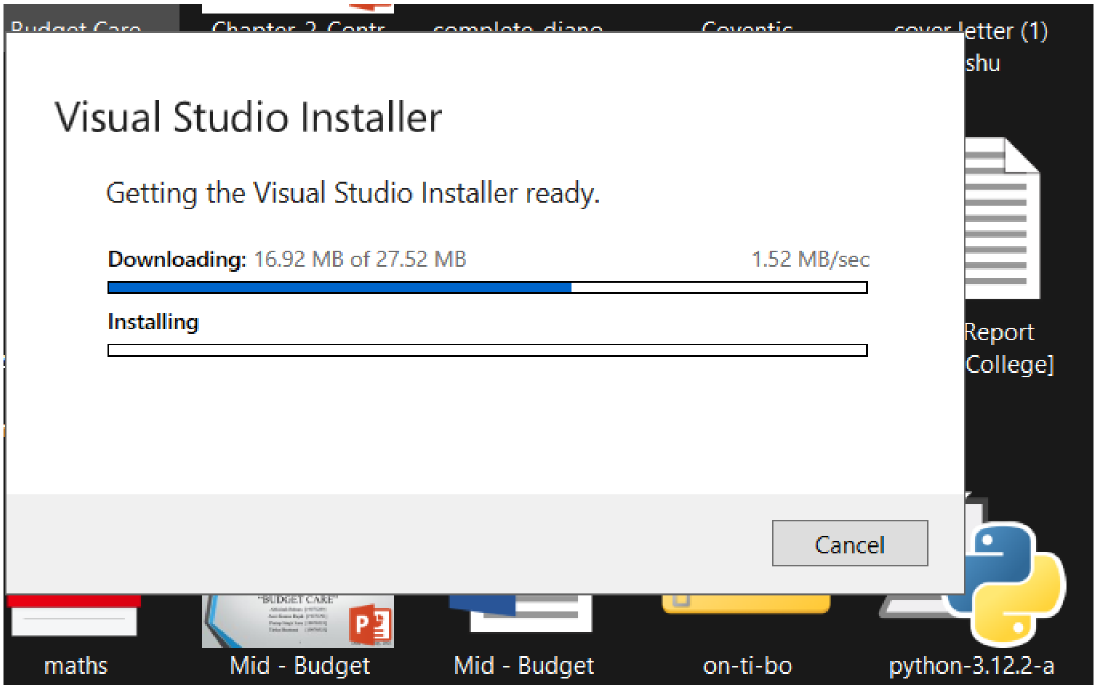
After that, I decide to open the code and move through the different sections to understand how that logic is used. While coding, special attention was paid to the primary line, any loops, conditionals, as well as function calls. This led to finding out the flow of the program and what task the program is supposed to accomplish.

To make it clearer, I have illustrated the specific screen shot of the program in a software called Visual Studio given below;



```
using System;
using System.Runtime.InteropServices;
using System.Threading;
using System.Threading.Tasks;
using System.Windows;
using Community.VisualStudio.Toolkit;
using Microsoft.VisualStudio.Imaging;
using Microsoft.VisualStudio.Shell;

namespace MyDemo
{
    1 reference
    public class ToolWindow1 : BaseToolWindow<ToolWindow1>
    {
        0 references
        public override string GetTitle(int toolWindowId) => "Tc

        0 references
        public override Type PaneType => typeof(Pane);

        0 references
        public override Task<FrameworkElement> CreateAsync(int t
```

```
Time Elapsed: 00:00:00.5332463
========== Finished ==========
```



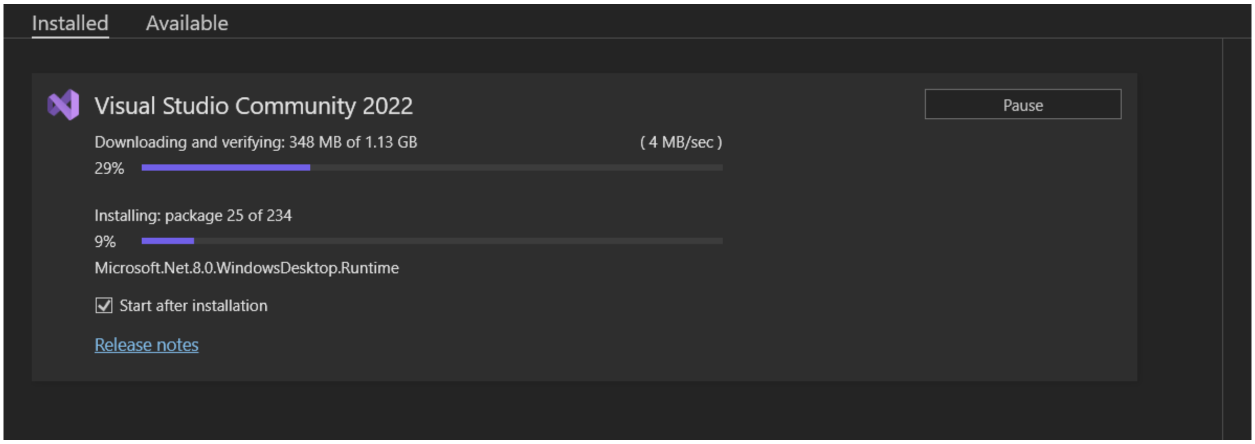
Visual Studio Installer

Getting the Visual Studio Installer ready.

Downloading: 16.92 MB of 27.52 MB 1.52 MB/sec

Installing

Cancel



Visual Studio Community 2022

Downloading and verifying: 348 MB of 1.13 GB (4 MB/sec)

29%

Installing: package 25 of 234

9%

Microsoft.Net.8.0.WindowsDesktop.Runtime

☑ Start after installation

Release notes

Task 2: Problem Description

Problem Identification: The C# console program given above effectively addresses the problem of addition, subtraction, multiplication, and division; it allows the user to input two numbers. This program should provide a very basic calculator and should be able to perform these basic operations in mathematics adequately without necessitating having to use a calculator or going for a special program.

- **Functionality Overview:** The major purpose of the program and its characteristics are as follows:

- **User Input:** This program will take two numbers from the user.
- **Operation Selection:** Giving the user the opportunity to select the desired type of arithmetic operation.
- **Calculation:** Carrying out of the selected operation on the given numbers.
- **Output:** Although the use of symbols is more common in the human body, their application benefits the calculation process and the subsequent display of the result.

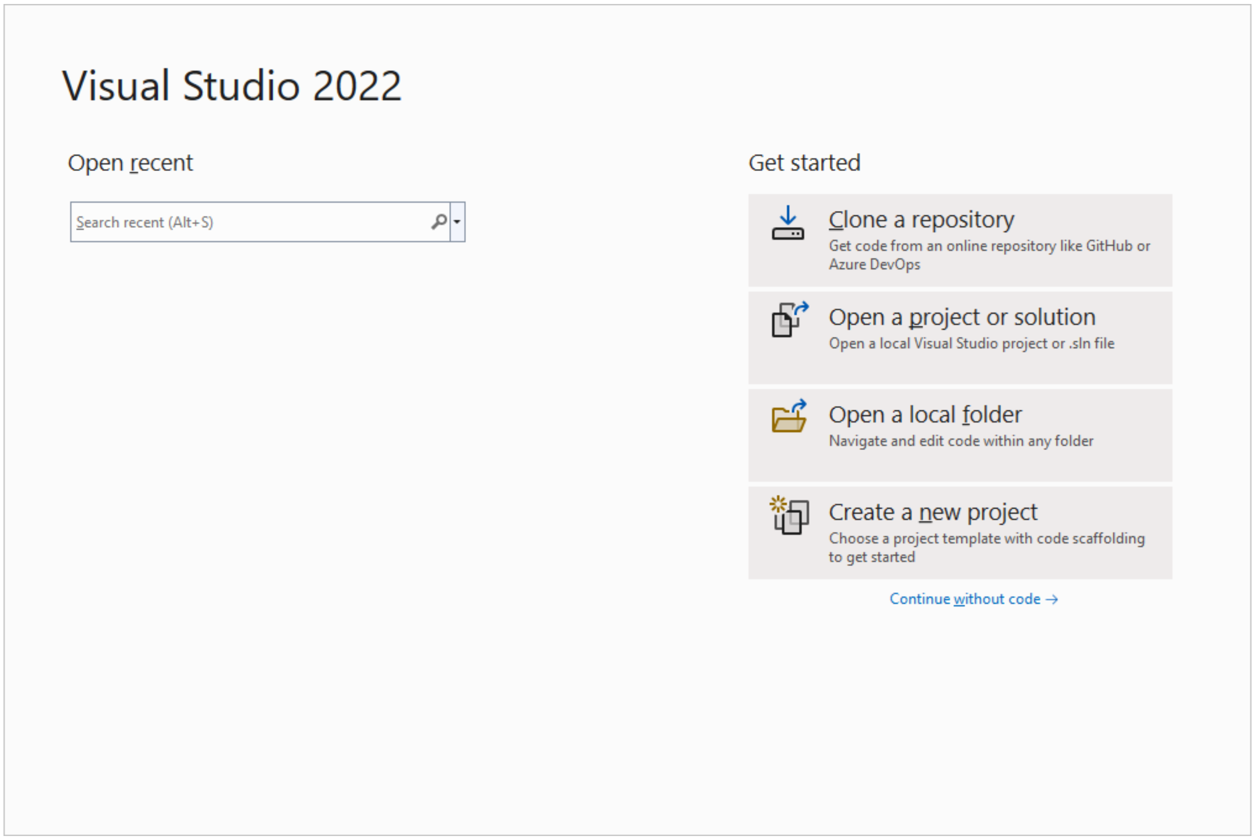
Logical Flow:

User Input: To find the result of the operation it asks the user to input two numbers.

Operation Selection: Then the user is asked to enter and operation that is either +, -, *, or / for addition, subtraction, multiplication or division.

Conditional Logic: concerning the selected operation, the program applies a conditional statement of if or switch to carry out the corresponding arithmetic operation.

Calculation and Output: After selecting an operation from the available ones, the program performs the selected operations on the numbers and outputs the result to the user.



Visual Studio 2022

Open recent

Search recent (Alt+S)

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

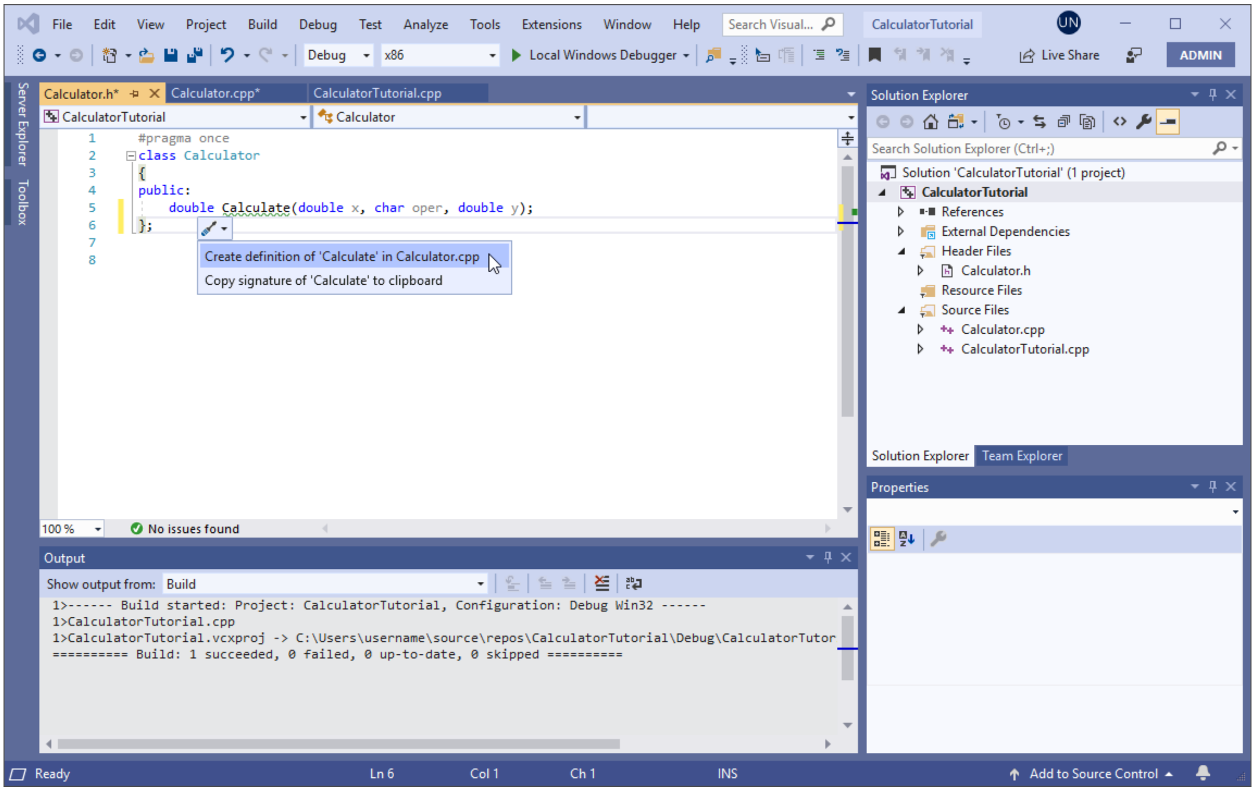
Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

```
// CalculatorTutorial.cpp : This file contains the 'main' function. Program execution begins and ends there.
//

#include <iostream>

using namespace std;

int main()
{
    cout << "Calculator Console Application" << endl << endl;
    cout << "Please enter the operation to perform. Format: a+b | a-b | a*b | a/b"

        << endl;

    return 0;

}



// Run program: Ctrl + F5 or Debug > Start Without Debugging menu

// Debug program: F5 or Debug > Start Debugging menu
```



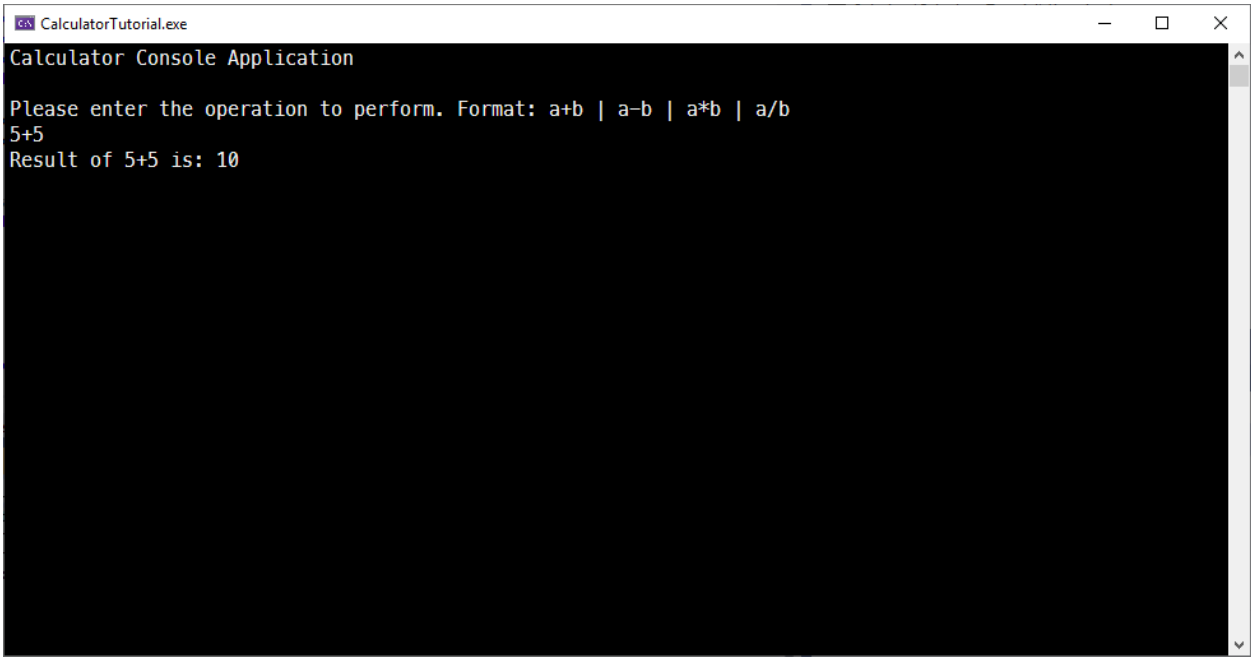
```
#pragma once
class Calculator
{
public:
    double Calculate(double x, char oper, double y);
};
```

```
1>------ Build started: Project: CalculatorTutorial, Configuration: Debug Win32 ------
1>CalculatorTutorial.cpp
1>CalculatorTutorial.vcxproj -> C:\Users\username\source\repos\CalculatorTutorial\Debug\CalculatorTutor
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

```
using System;


class Calculator

{

    static void Main(string[] args)

    {
```

```
Console.Write("Enter the first number: ");
double num1 = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter the second number: ");
double num2 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Select an operation: +, -, *, /");
string operation = Console.ReadLine();

double result = 0;

switch (operation)
{
    case "+":
        result = num1 + num2;
        break;
    case "-":
        result = num1 - num2;
        break;
    case "*":
        result = num1 * num2;
        break;
    case "/":
        if (num2 != 0)
        {
            result = num1 / num2;
        }
        else
        {
            Console.WriteLine("Cannot divide by zero.");
            return;
        }
```

```
                break;
            default:
                Console.WriteLine("Invalid operation selected.");
                return;
        }

        Console.WriteLine("The result is: " + result);
    }
}
```

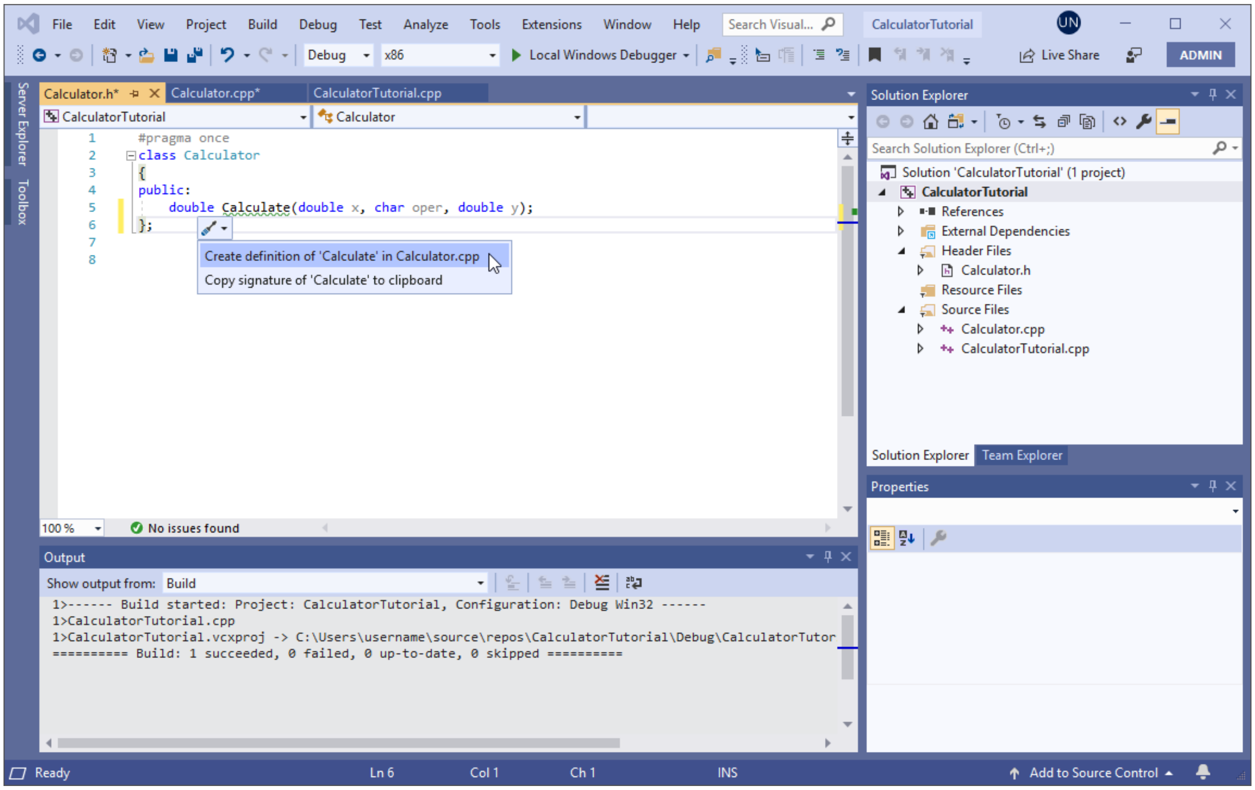


```
Calculator Console Application

Please enter the operation to perform. Format: a+b | a-b | a*b | a/b
5+5
Result of 5+5 is: 10
```

Task 3: Redesign and Recode

To enhance the calculator program's structure and functionality, I identified four changes: refactoring code into functions, introducing a class, using a list for operations, and adding error handling. Each change improves code readability, reusability, and maintainability.

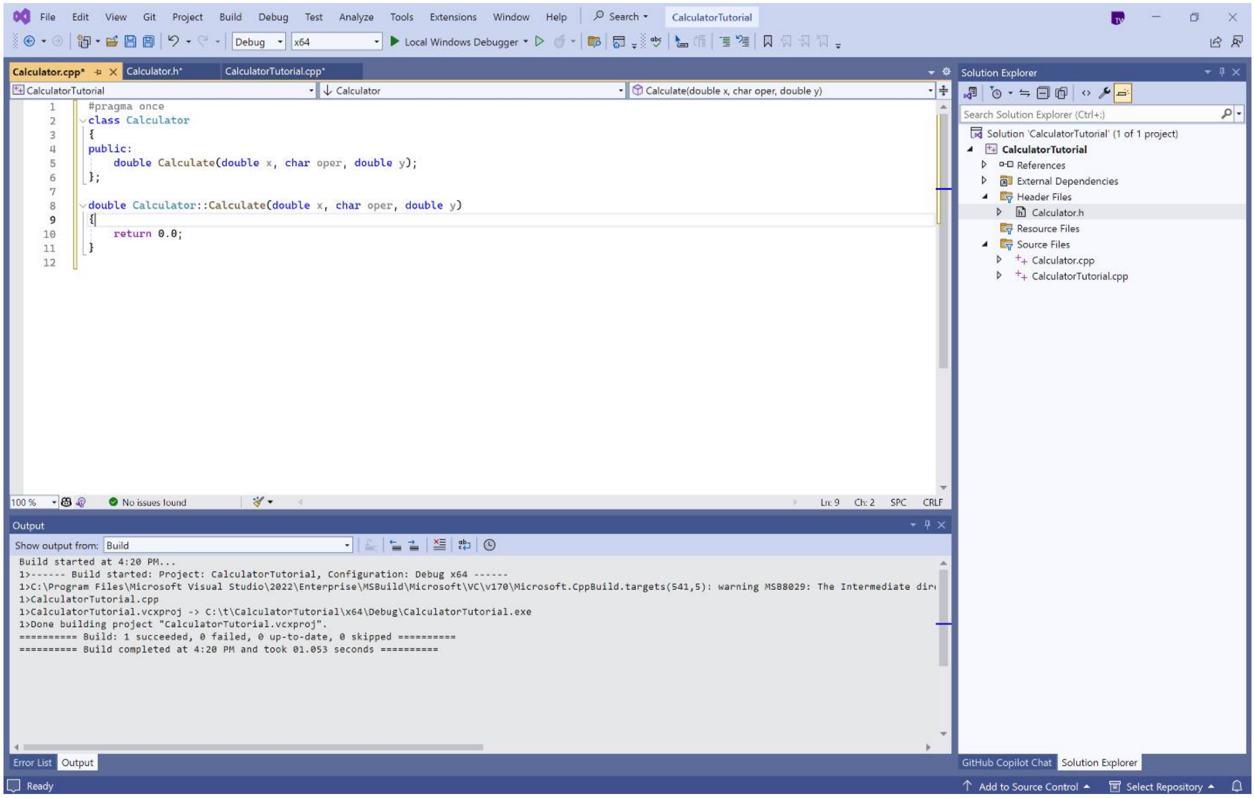
Change 1: Refactoring into Functions



```
#pragma once
class Calculator
{
public:
    double Calculate(double x, char oper, double y);
};
```

Show output from: Build

```
1>------ Build started: Project: CalculatorTutorial, Configuration: Debug Win32 ------
1>CalculatorTutorial.cpp
1>CalculatorTutorial.vcxproj -> C:\Users\username\source\repos\CalculatorTutorial\Debug\CalculatorTutor
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```



```
#pragma once
class Calculator
{
public:
    double Calculate(double x, char oper, double y);
};

double Calculator::Calculate(double x, char oper, double y)
{
    return 0.0;
}
```

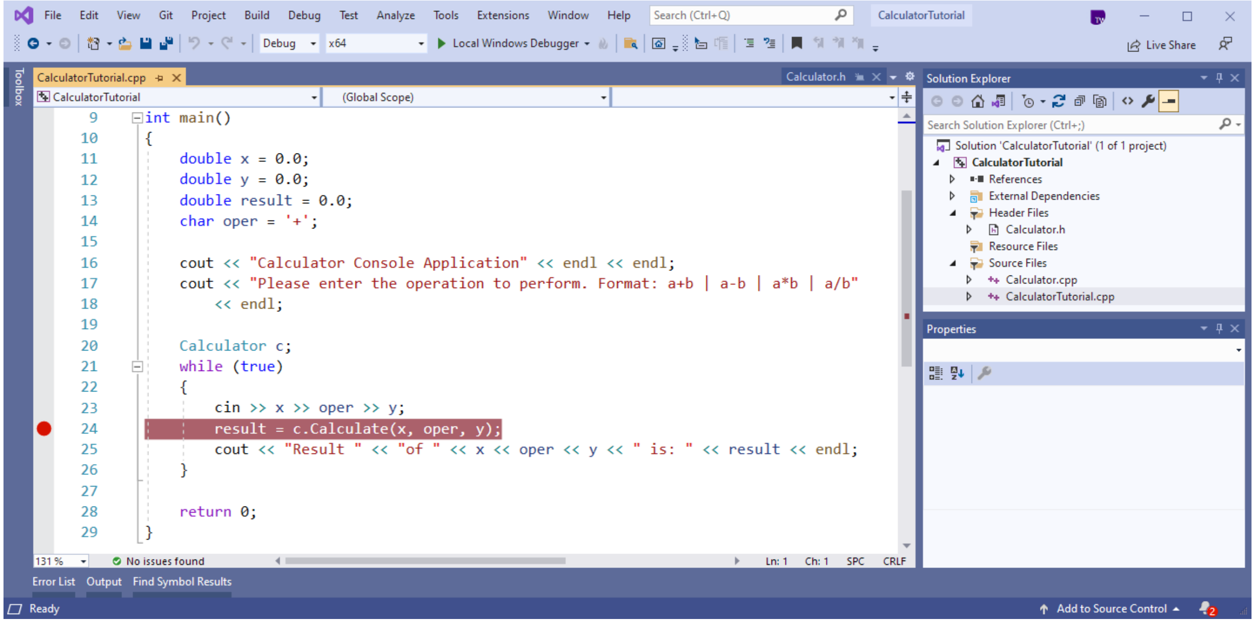
```
Build started at 4:20 PM...
1>------ Build started: Project: CalculatorTutorial, Configuration: Debug x64 ------
1>C:\Program Files\Microsoft Visual Studio\2022\Enterprise\MSBuild\Microsoft\VC\v170\Microsoft.CppBuild.targets(541,5): warning MSB8029: The Intermediate dir
1>CalculatorTutorial.cpp
1>CalculatorTutorial.vcxproj -> C:\t\CalculatorTutorial\x64\Debug\CalculatorTutorial.exe
1>Done building project "CalculatorTutorial.vcxproj".
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build completed at 4:20 PM and took 01.053 seconds ==========
```

Switch to the Calculator.cpp file in the editor window.

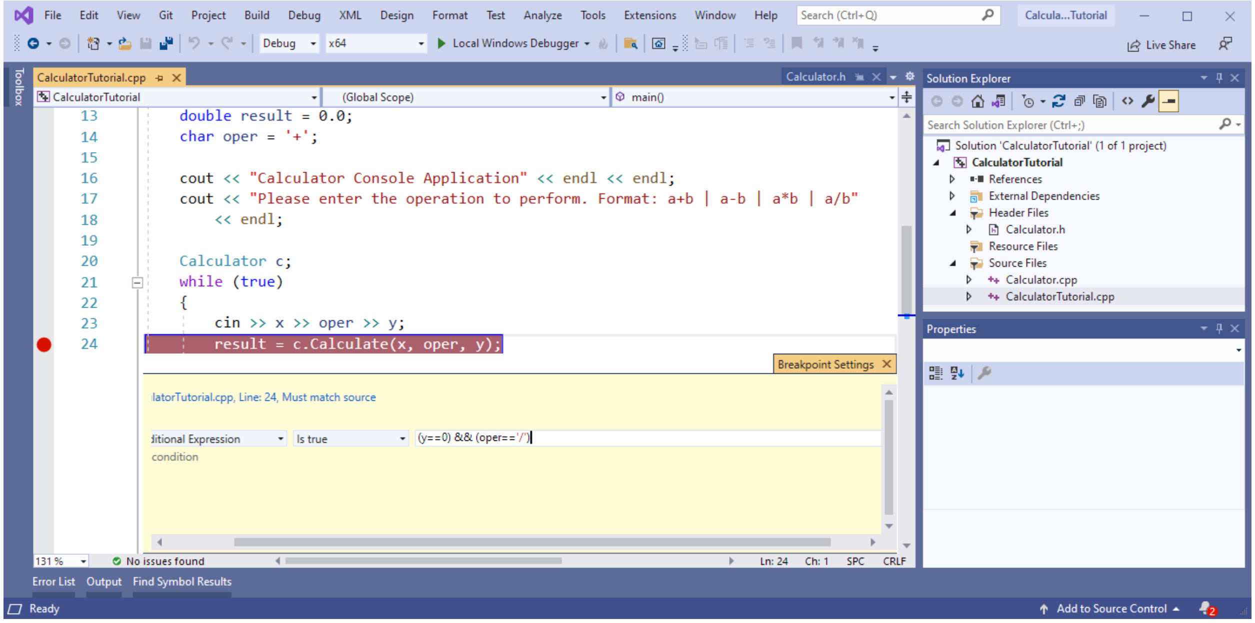
Replace the contents of Calculator: Calculate (double x, char oper, double y) with:

```
double Calculator::Calculate(double x, char oper, double y)
{
   switch(oper)
   {
      case '+':
         return x + y;
      case '-':
         return x - y;
      case '*':
         return x * y;
```

```
        case '/':
            return x / y;
        default:
            return 0.0;
    }
}
```



```
int main()
{
    double x = 0.0;
    double y = 0.0;
    double result = 0.0;
    char oper = '+';

    cout << "Calculator Console Application" << endl << endl;
    cout << "Please enter the operation to perform. Format: a+b | a-b | a*b | a/b"
        << endl;

    Calculator c;
    while (true)
    {
        cin >> x >> oper >> y;
        result = c.Calculate(x, oper, y);
        cout << "Result " << "of " << x << oper << y << " is: " << result << endl;
    }

    return 0;
}
```



```
    double result = 0.0;
    char oper = '+';

    cout << "Calculator Console Application" << endl << endl;
    cout << "Please enter the operation to perform. Format: a+b | a-b | a*b | a/b"
        << endl;

    Calculator c;
    while (true)
    {
        cin >> x >> oper >> y;
        result = c.Calculate(x, oper, y);
```

Breakpoint Settings

latorTutorial.cpp, Line: 24, Must match source

litional Expression | Is true | (y==0) && (oper=='/')

condition

No issues found   Ln: 24   Ch: 1   SPC   CRLF