

# Software Reuse

Software Design Architecture Lab # 11

Muhammad Rehan Baig

# Software Reuse

- Code reuse is the use of existing software to build new software. It is one of the holy grails of modern software development. APIs provide a mechanism to enable code reuse.
- In the early years of software development, it was common for a company to have to write all of the code for any application they produced.
- If the program needed to read GIF images or parse a text file, the company would have to write all that code in-house. Nowadays, with the proliferation of good commercial and open source libraries

# Software Reuse

- software development has become much more modular, with the use of distinct components that form the building blocks of an application and talk together via their published APIs.
- The benefit of this approach is that you don't need to understand every detail of every software component, in the same way that for the earlier house building analogy you can delegate many details to professional contractors.
- It also allows you to concentrate on your core business logic instead of having to spend time **reinventing the wheel**.

# Software Reuse Examples

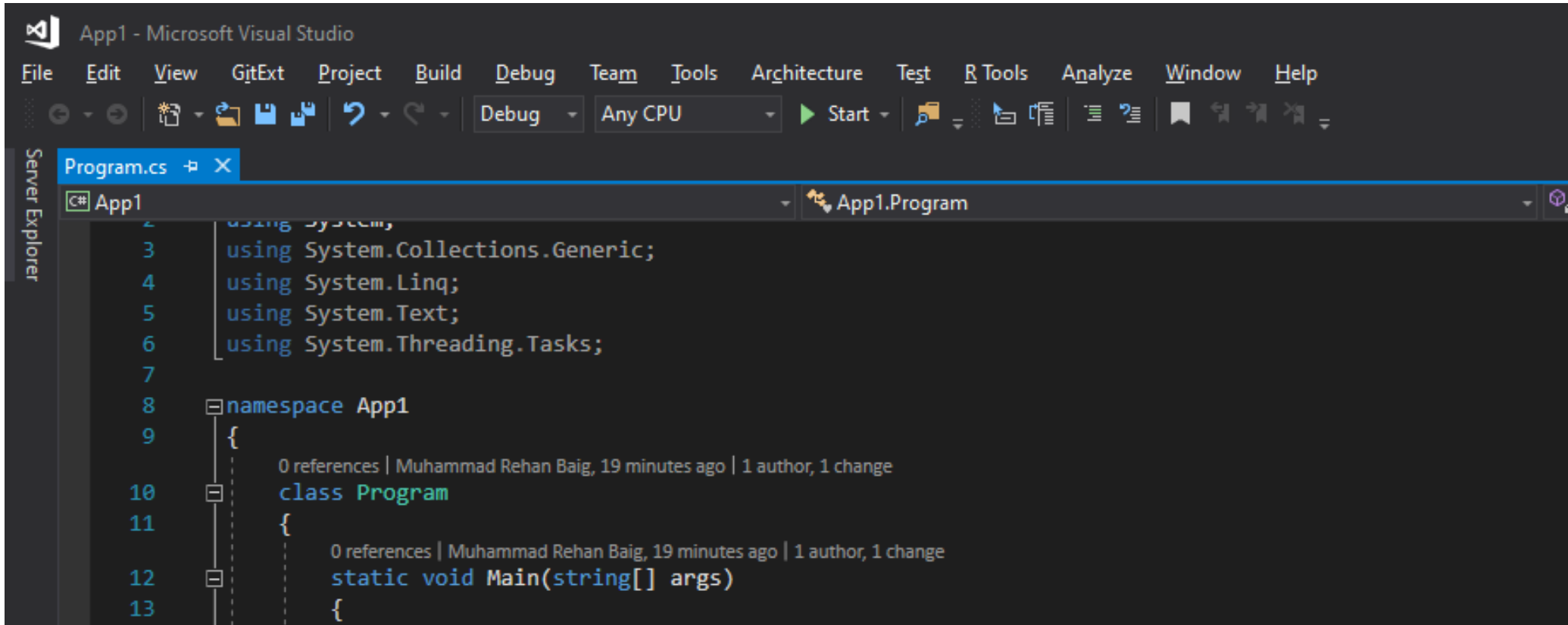
- Api(Application Programming Interfaces).  
Google Apis(Maps, Cloud, Firebase, Drive ... etc)
- Shared Libraries
- Plugins
- Extensions: eg: (NewtonSoft Json) we have used in previous lectures task

# Creating Reusable Software Example

- In our example scenario we are building shared library project that performs basic calculation.
- We will create two projects.
- 1<sup>st</sup> project contains calculation class that can be used for accounting in the project.
- We will use this project in our 2<sup>nd</sup> project as DLL shared library for calculation purpose.

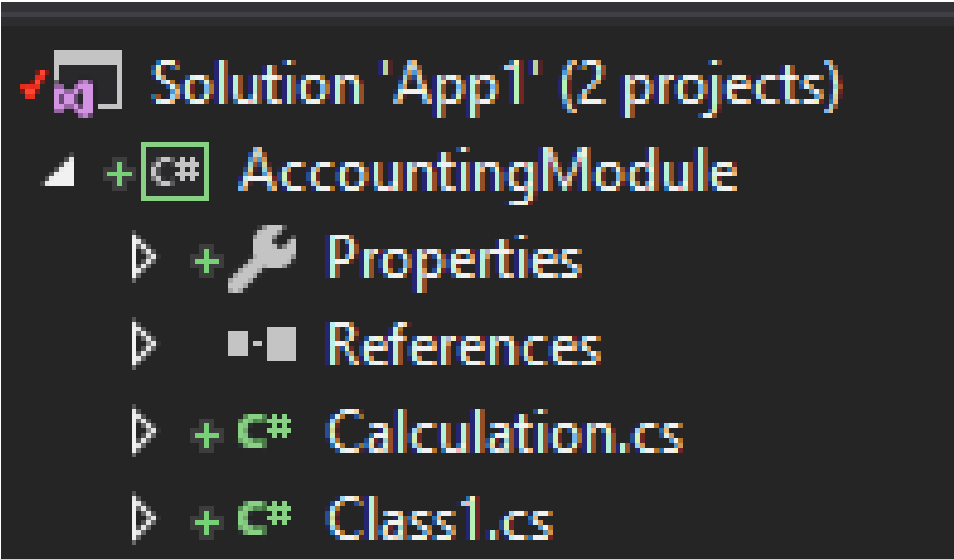
Example Step by Step

# Step1 – Creating Project 1



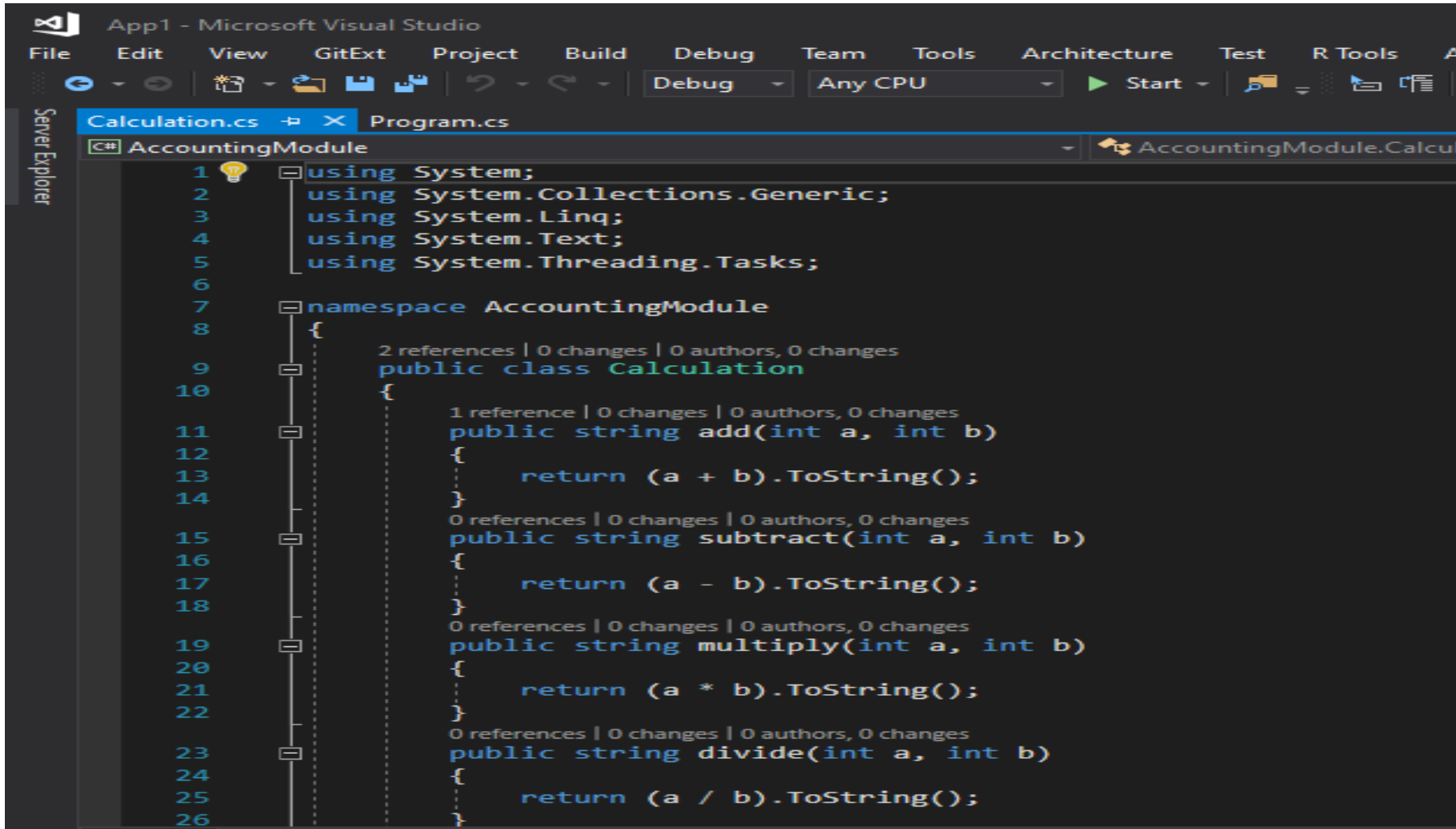
```
App1 - Microsoft Visual Studio
File Edit View GitExt Project Build Debug Team Tools Architecture Test R Tools Analyze Window Help
Debug Any CPU Start
Server Explorer
Program.cs
App1
App1.Program
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace App1
9 {
10     class Program
11     {
12         static void Main(string[] args)
13         {
```

# Step2 – Adding Class library Accounting as a Project in 1<sup>st</sup> Project for making Calculation DLL



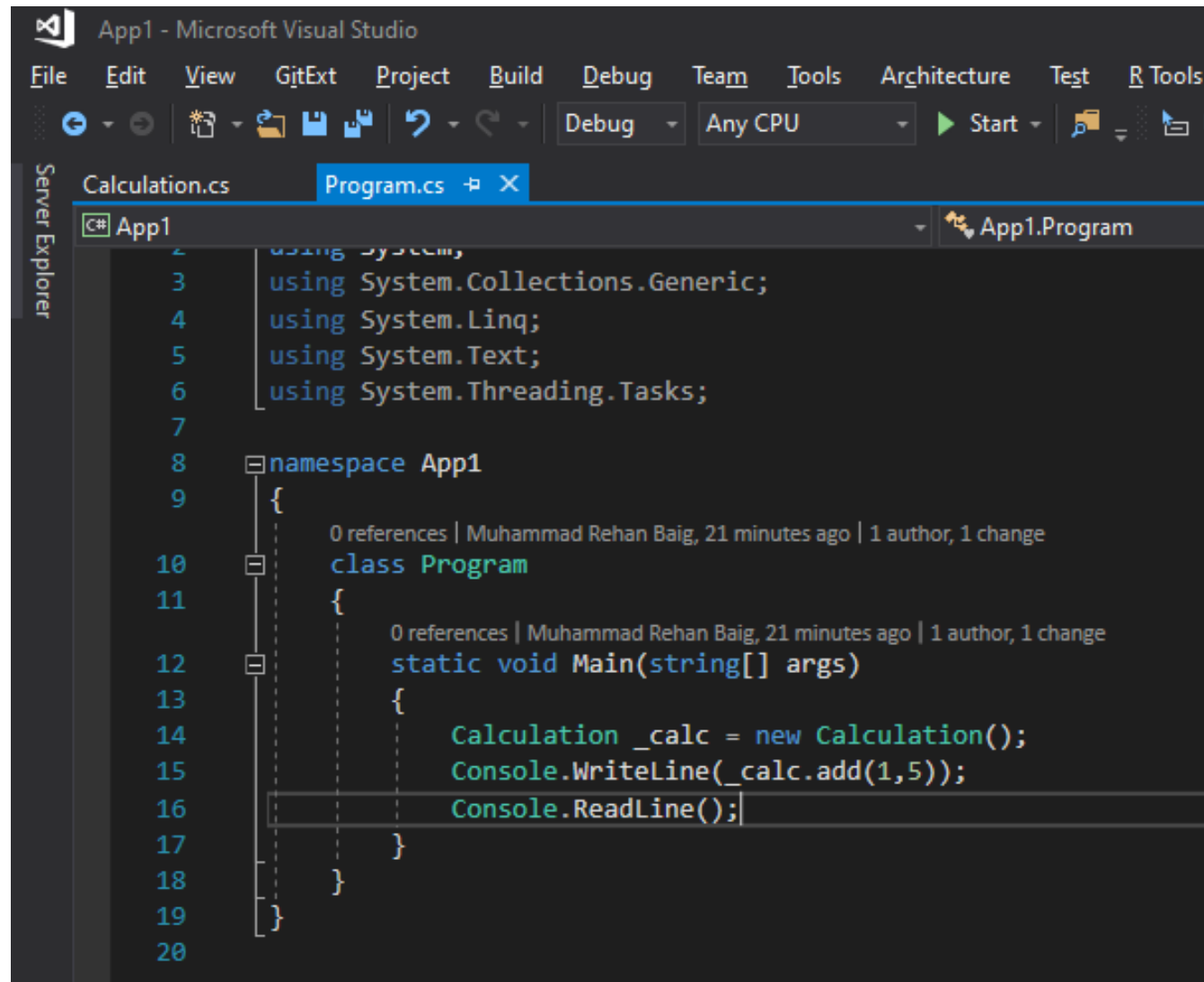


# Step2 – Adding Class library Accounting as a Project in 1<sup>st</sup> Project for making Calculation DLL



```
App1 - Microsoft Visual Studio
File Edit View GitExt Project Build Debug Team Tools Architecture Test R Tools A
Calculation.cs Program.cs
C# AccountingModule AccountingModule.Calcula
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace AccountingModule
8 {
9     2 references | 0 changes | 0 authors, 0 changes
10     public class Calculation
11     {
12         1 reference | 0 changes | 0 authors, 0 changes
13         public string add(int a, int b)
14         {
15             return (a + b).ToString();
16         }
17         0 references | 0 changes | 0 authors, 0 changes
18         public string subtract(int a, int b)
19         {
20             return (a - b).ToString();
21         }
22         0 references | 0 changes | 0 authors, 0 changes
23         public string multiply(int a, int b)
24         {
25             return (a * b).ToString();
26         }
27         0 references | 0 changes | 0 authors, 0 changes
28         public string divide(int a, int b)
29         {
30             return (a / b).ToString();
31         }
32     }
33 }
```

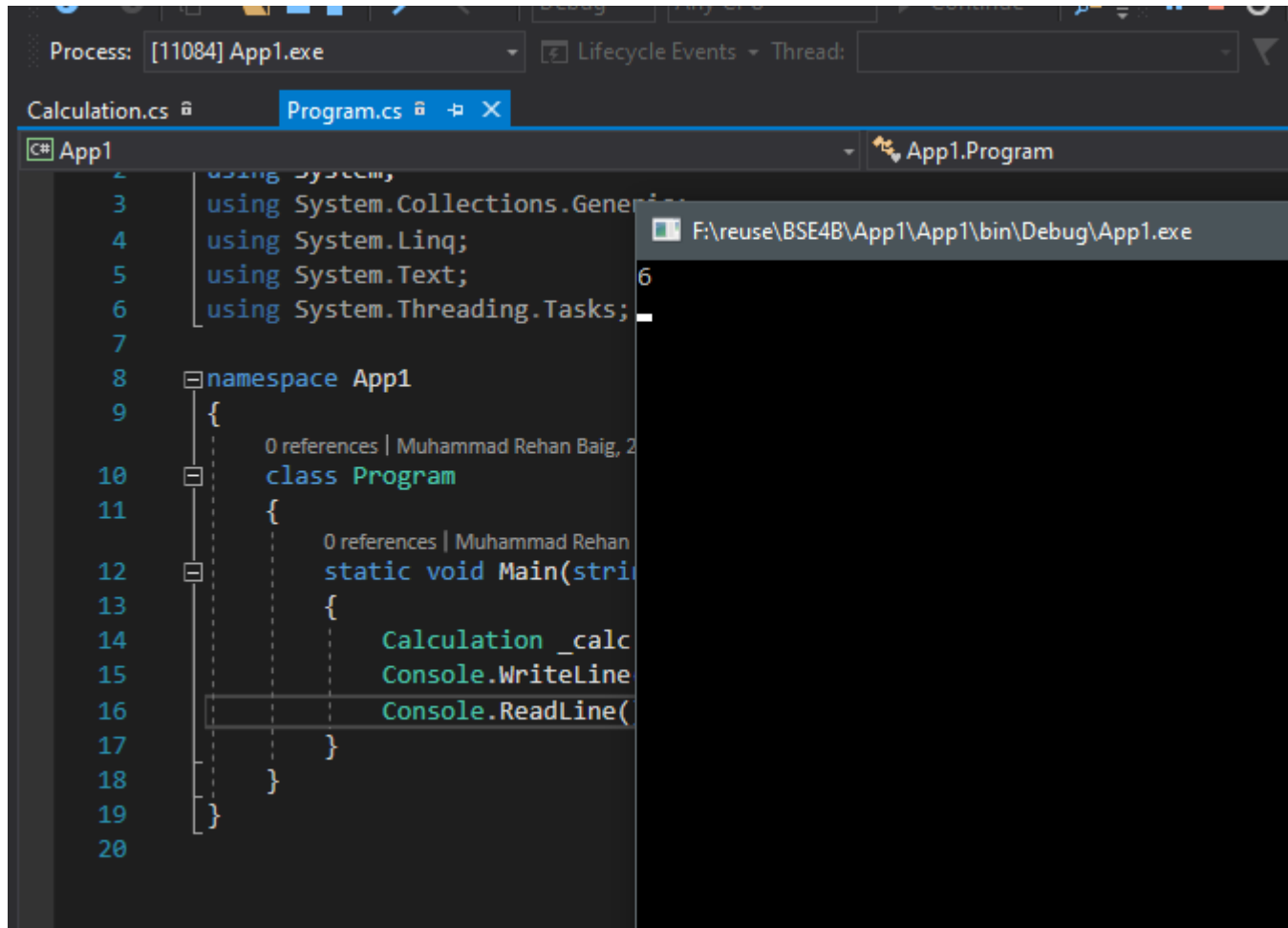
# Step3 – Consuming App1 Accounting Calculation



```
App1 - Microsoft Visual Studio
File Edit View GitExt Project Build Debug Team Tools Architecture Test R Tools
Debug Any CPU Start
Server Explorer
Calculation.cs Program.cs
App1
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace App1
{
    0 references | Muhammad Rehan Baig, 21 minutes ago | 1 author, 1 change
    class Program
    {
        0 references | Muhammad Rehan Baig, 21 minutes ago | 1 author, 1 change
        static void Main(string[] args)
        {
            Calculation _calc = new Calculation();
            Console.WriteLine(_calc.add(1,5));
            Console.ReadLine();
        }
    }
}
```

# Output

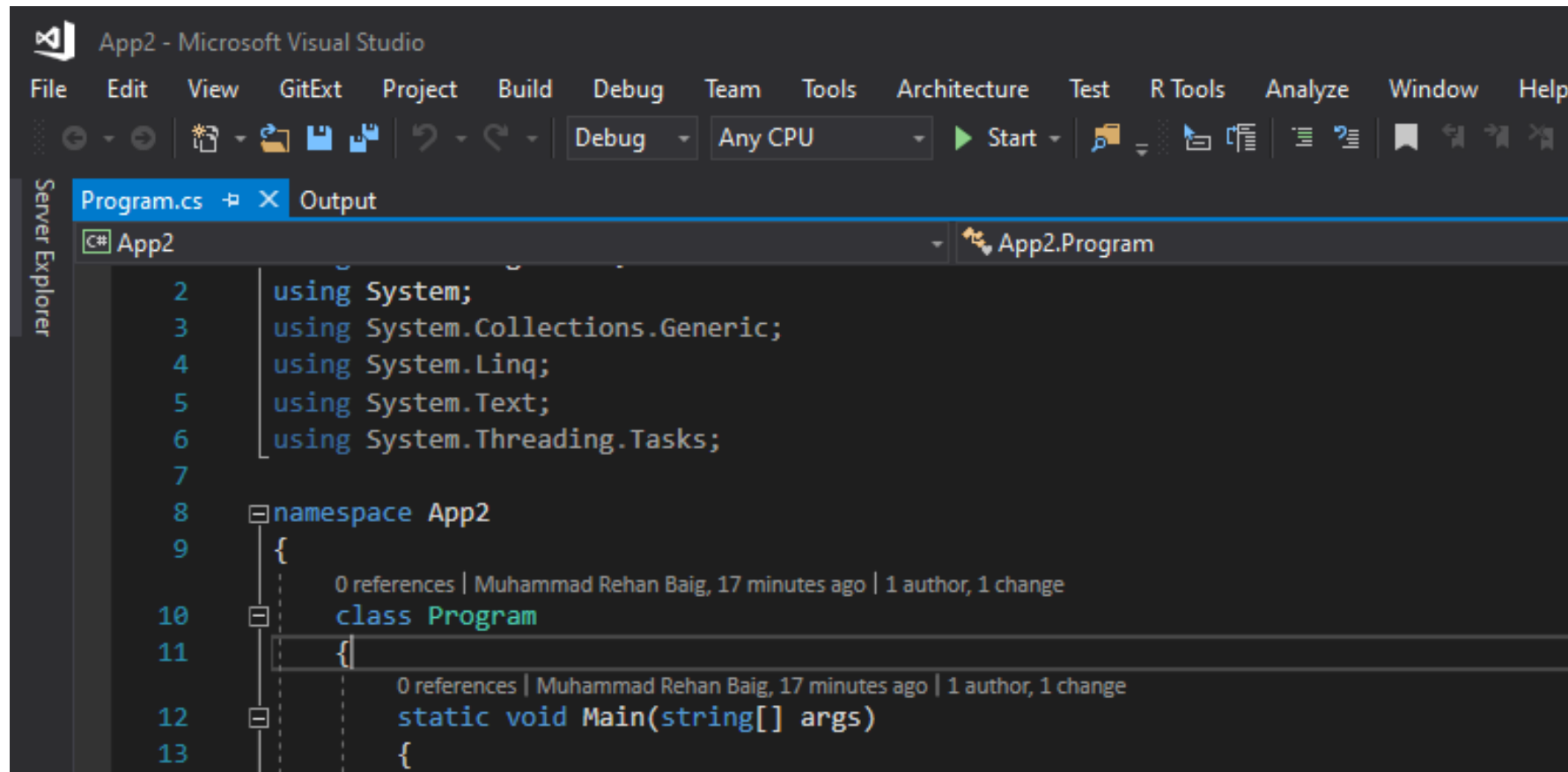


The screenshot shows the Visual Studio IDE with a C# project named 'App1'. The 'Program.cs' file is open, displaying the following code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7
8 namespace App1
9 {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             Calculation _calc = new Calculation();
15             Console.WriteLine(_calc.Calculate(1, 2, 3, 4, 5));
16             Console.ReadLine();
17         }
18     }
19 }
20
```

The 'App1' folder in the Solution Explorer is expanded, showing the 'Program.cs' file. The 'App1.Program' class is selected in the Solution Explorer. The 'Output' window is open, showing the path 'F:\reuse\BSE4B\App1\bin\Debug\App1.exe' and the output '6'.

# Step4 – Creating 2<sup>nd</sup> Project

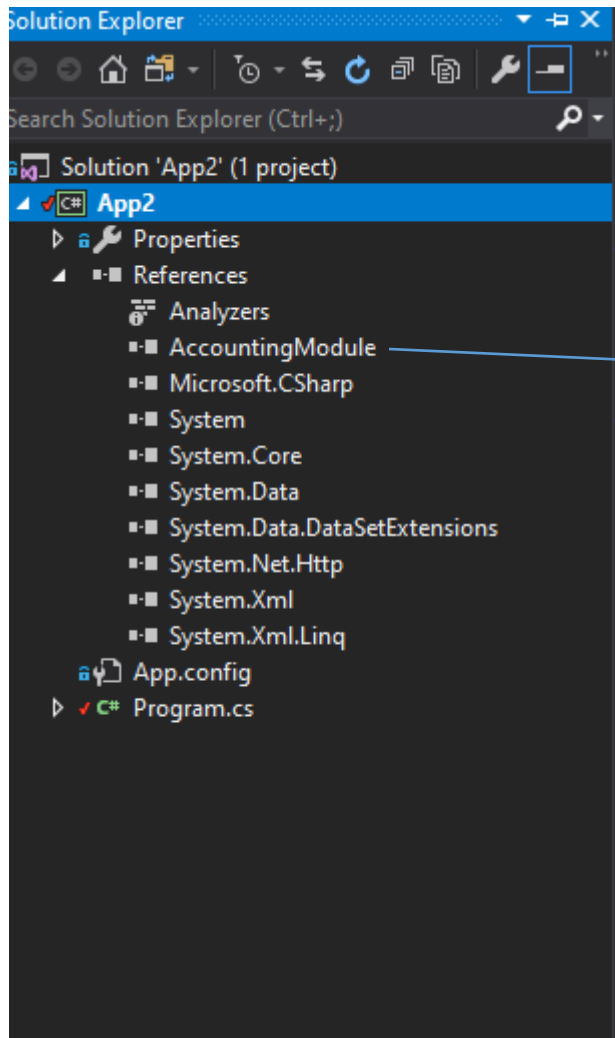


The screenshot shows the Microsoft Visual Studio IDE with a project named 'App2'. The 'Program.cs' file is open, and the code is as follows:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7
8  namespace App2
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
```

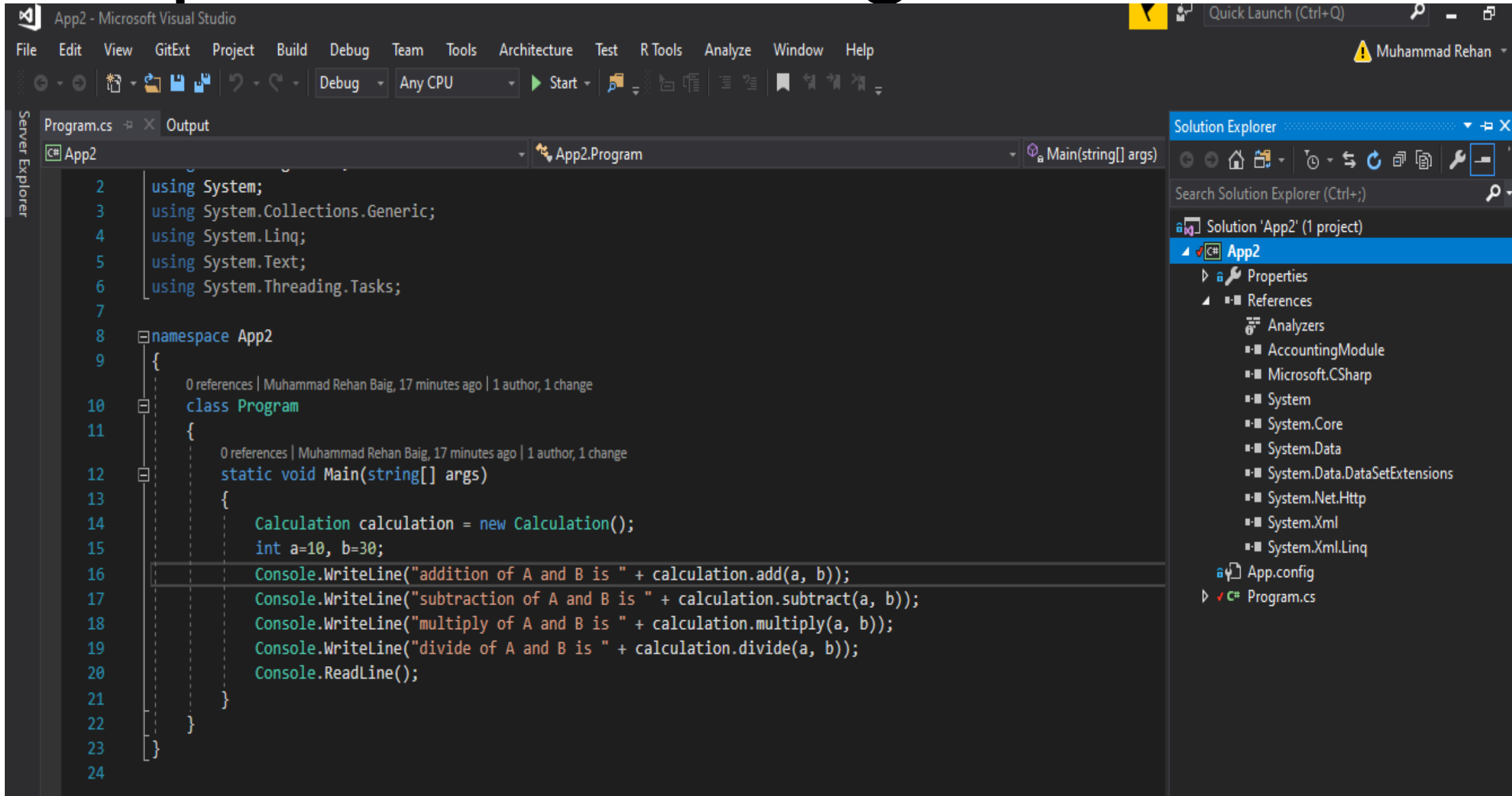
The code is written in C# and includes using statements for various System namespaces. It defines a namespace 'App2' and a class 'Program' with a static 'Main' method. The code is currently incomplete, with the 'Main' method body starting with an opening curly brace on line 13.

# Step5 – Adding DLL (References) for Reuse of Calculation class



Adding Accounting from Project 1  
As a Reference DLL  
Shared Library

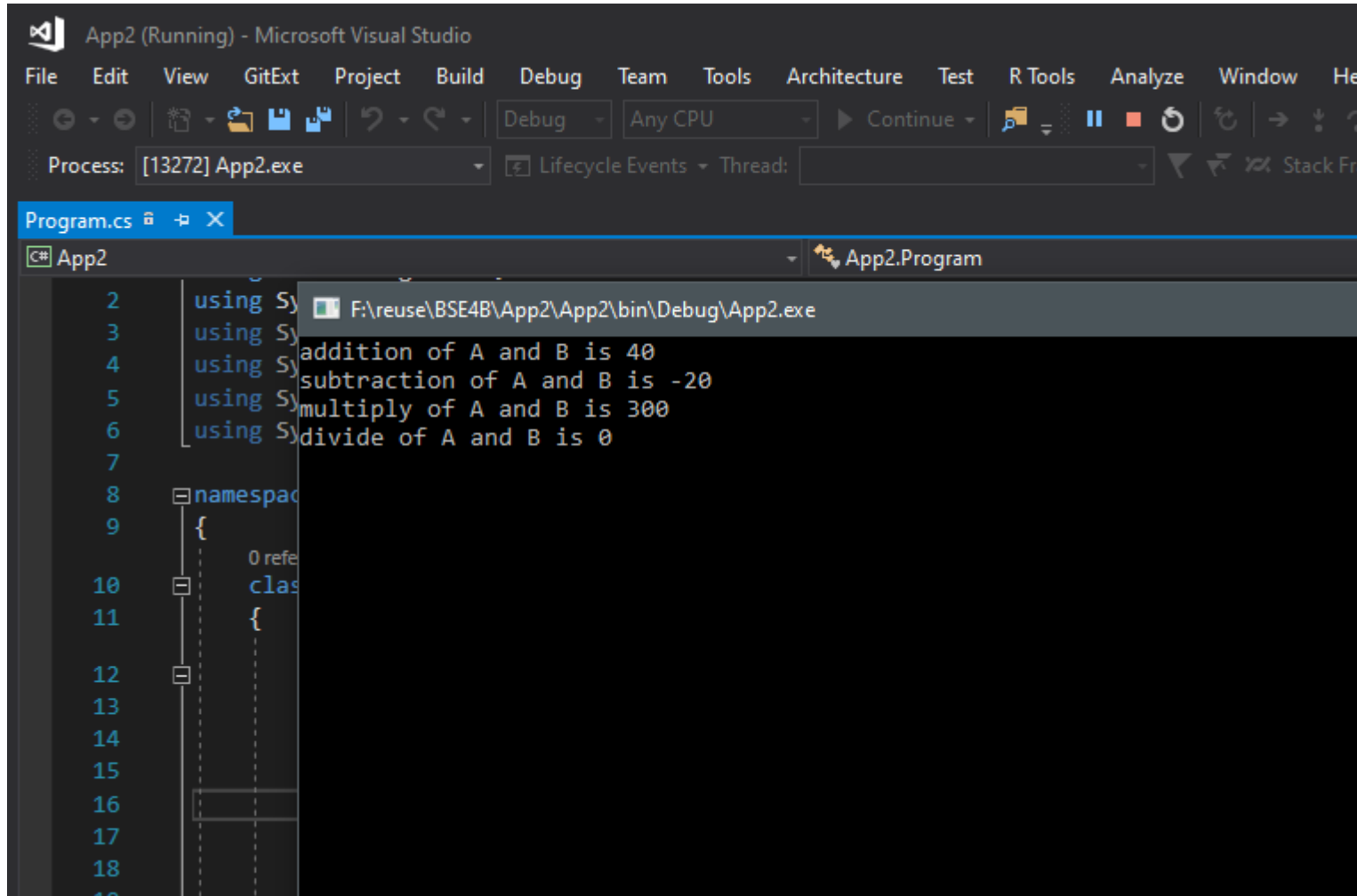
# Step5 – Reuse Existing Code



The screenshot displays the Microsoft Visual Studio IDE with a C# project named 'App2'. The main window shows the source code for 'Program.cs', which includes using statements for System, System.Collections.Generic, System.Linq, System.Text, and System.Threading.Tasks. The code defines a namespace 'App2' containing a class 'Program'. Inside the 'Program' class, there is a static method 'Main' that creates an instance of a 'Calculation' class and uses its methods to perform arithmetic operations on the values 10 and 30. The results are printed to the console using 'Console.WriteLine'. The 'Solution Explorer' on the right shows the project structure, including references to various .NET Framework assemblies like System, System.Core, System.Data, System.Data.DataSetExtensions, System.Net.Http, System.Xml, and System.Xml.Linq. The 'App.config' file is also listed.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7
8 namespace App2
9 {
10     0 references | Muhammad Rehan Baig, 17 minutes ago | 1 author, 1 change
11     class Program
12     {
13         0 references | Muhammad Rehan Baig, 17 minutes ago | 1 author, 1 change
14         static void Main(string[] args)
15         {
16             Calculation calculation = new Calculation();
17             int a=10, b=30;
18             Console.WriteLine("addition of A and B is " + calculation.add(a, b));
19             Console.WriteLine("subtraction of A and B is " + calculation.subtract(a, b));
20             Console.WriteLine("multiply of A and B is " + calculation.multiply(a, b));
21             Console.WriteLine("divide of A and B is " + calculation.divide(a, b));
22             Console.ReadLine();
23         }
24     }
```

# Output



App2 (Running) - Microsoft Visual Studio

File Edit View GitExt Project Build Debug Team Tools Architecture Test R Tools Analyze Window Help

Process: [13272] App2.exe Lifecycle Events Thread: Stack Frame

Program.cs

C# App2

App2.Program

F:\reuse\BSE4B\App2\App2\bin\Debug\App2.exe

```
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace App2
9 {
10     class Program
11     {
12     }
13
14
15
16
17
18
19
```

addition of A and B is 40  
subtraction of A and B is -20  
multiply of A and B is 300  
divide of A and B is 0

# Tasks

1. **Create** Reusable Code/Software for generating Marksheet of a student.  
(Hint: Use Project 1 for calculation and Grading purpose)
2. Consume Google Maps Api in Html Webpage.