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)

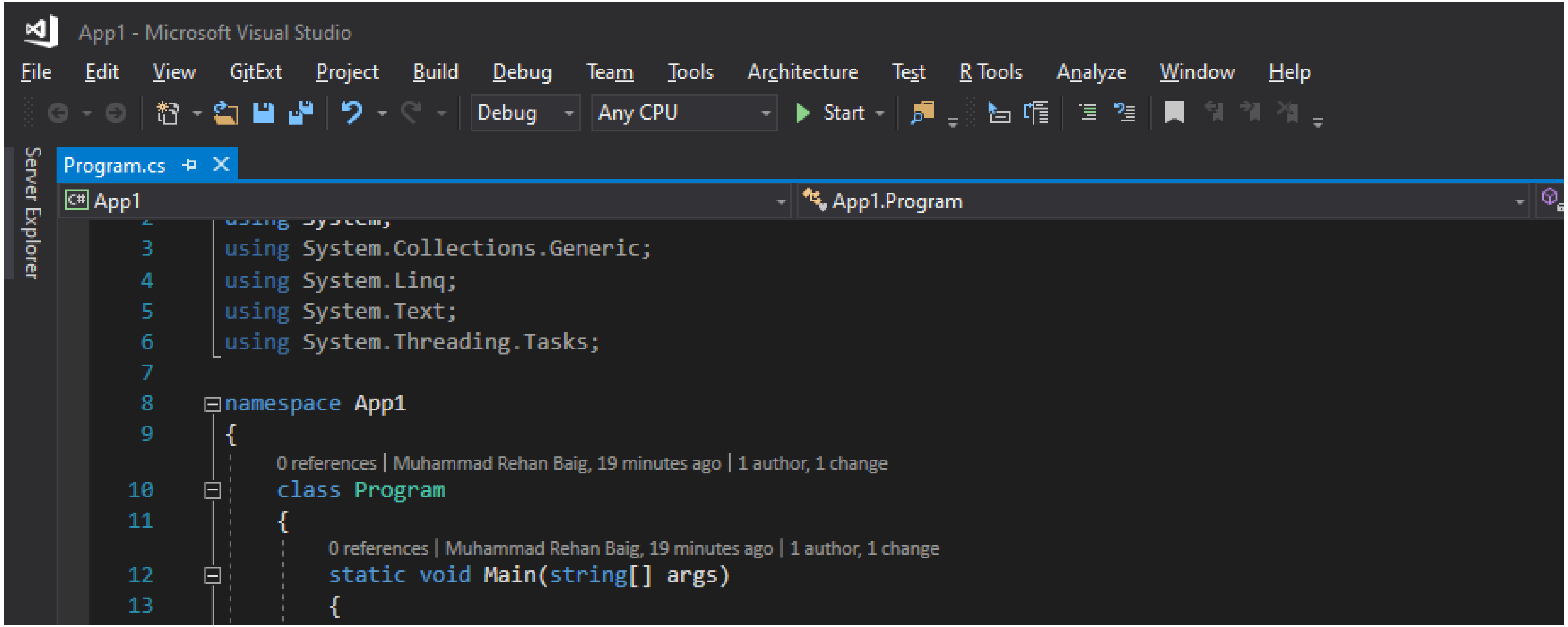
- Shared Libraries
- Plugins
- Extensions: eg: (NewtonSoft Json) we have used in previous lectures task

Creating Reusable Software Example

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

Example Step by Step

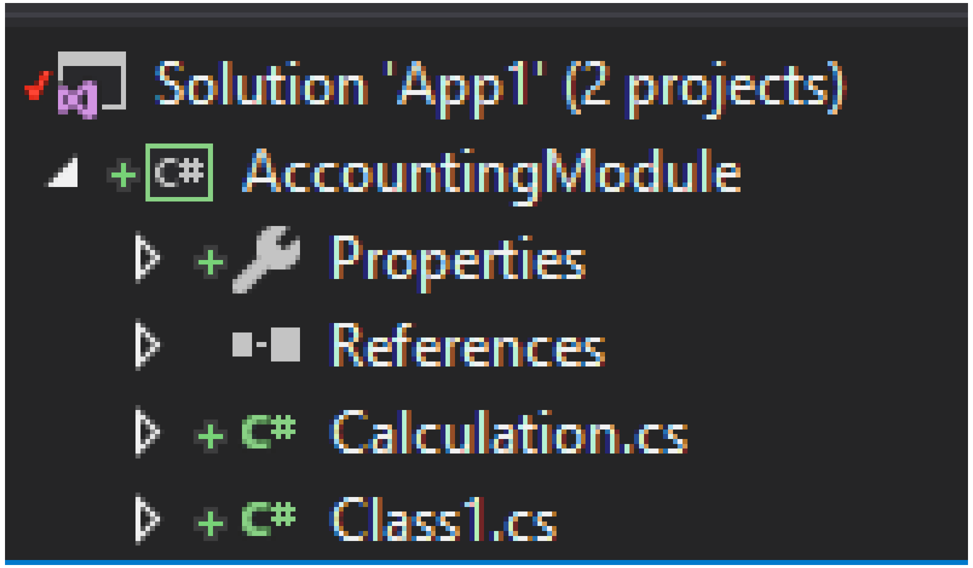
Step1 – Creating Project 1



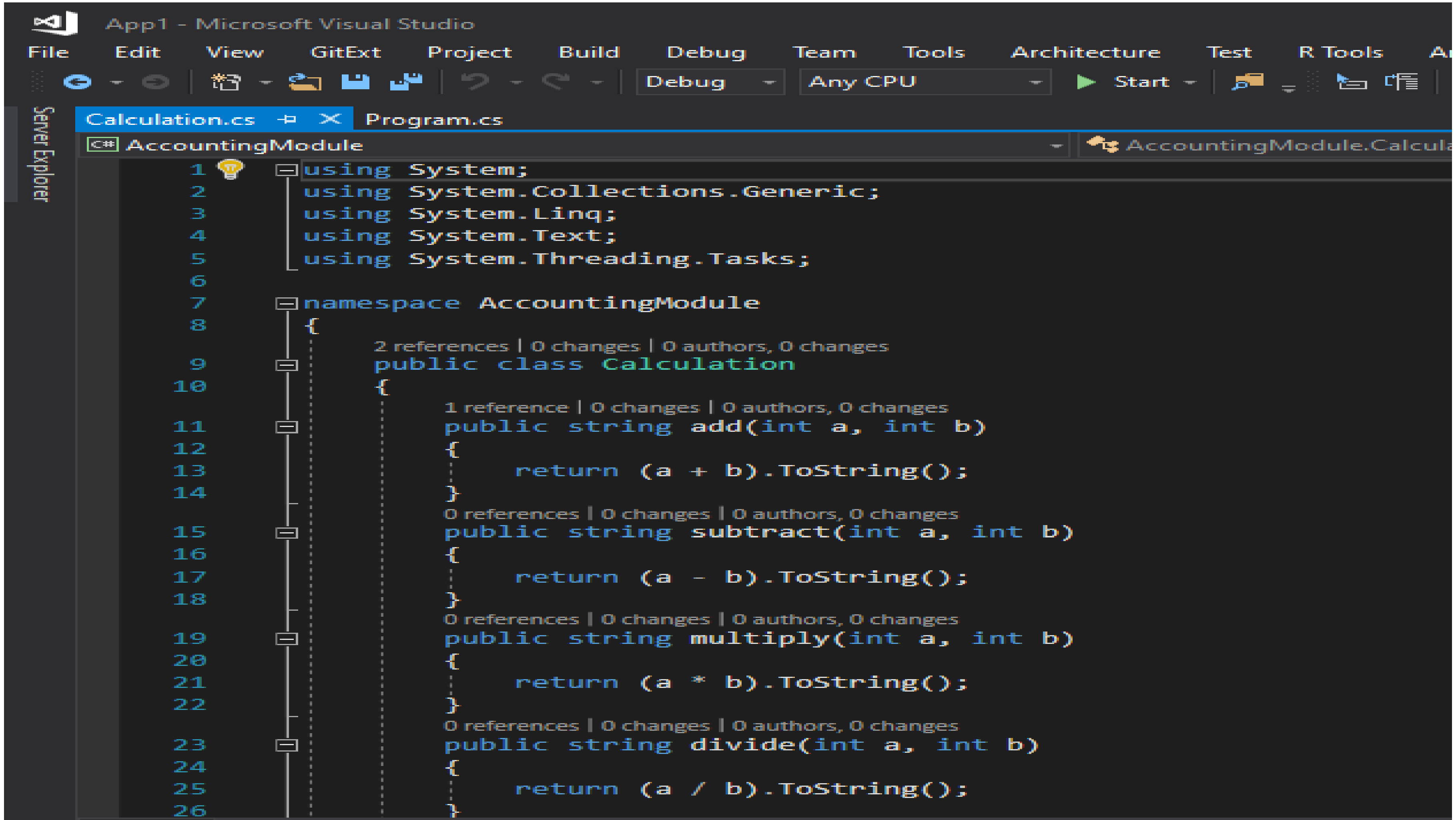
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace App1
{
    0 references | Muhammad Rehan Baig, 19 minutes ago | 1 author, 1 change
    class Program
    {
        0 references | Muhammad Rehan Baig, 19 minutes ago | 1 author, 1 change
        static void Main(string[] args)
        {
```

Step2 – Adding Class library Accounting as a Project in 1st Project for making Calculation DLL



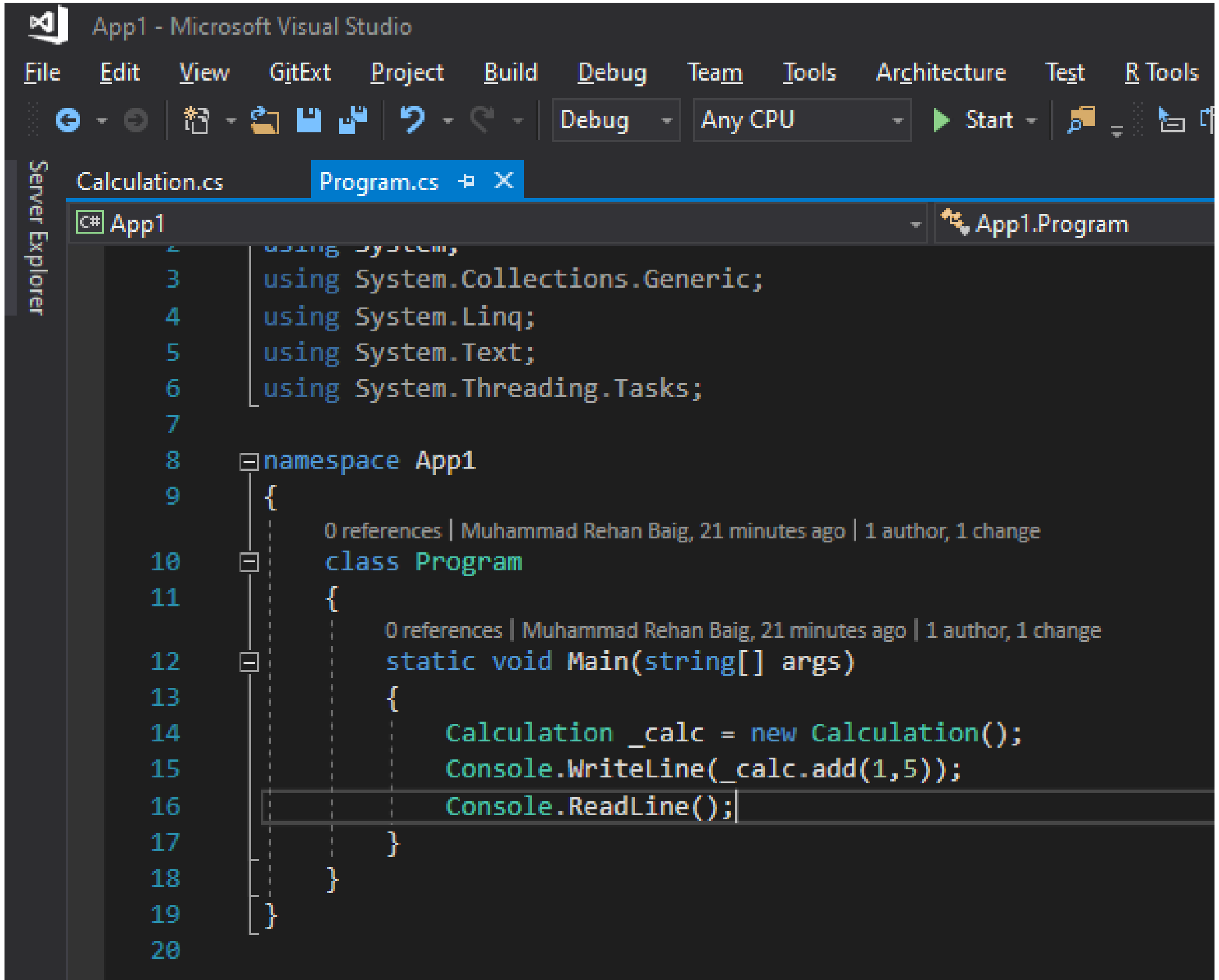
Step2 – Adding Class library Accounting as a Project in 1st Project for making Calculation DLL



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AccountingModule
{
    2 references | 0 changes | 0 authors, 0 changes
    public class Calculation
    {
        1 reference | 0 changes | 0 authors, 0 changes
        public string add(int a, int b)
        {
            return (a + b).ToString();
        }
        0 references | 0 changes | 0 authors, 0 changes
        public string subtract(int a, int b)
        {
            return (a - b).ToString();
        }
        0 references | 0 changes | 0 authors, 0 changes
        public string multiply(int a, int b)
        {
            return (a * b).ToString();
        }
        0 references | 0 changes | 0 authors, 0 changes
        public string divide(int a, int b)
        {
            return (a / b).ToString();
        }
```

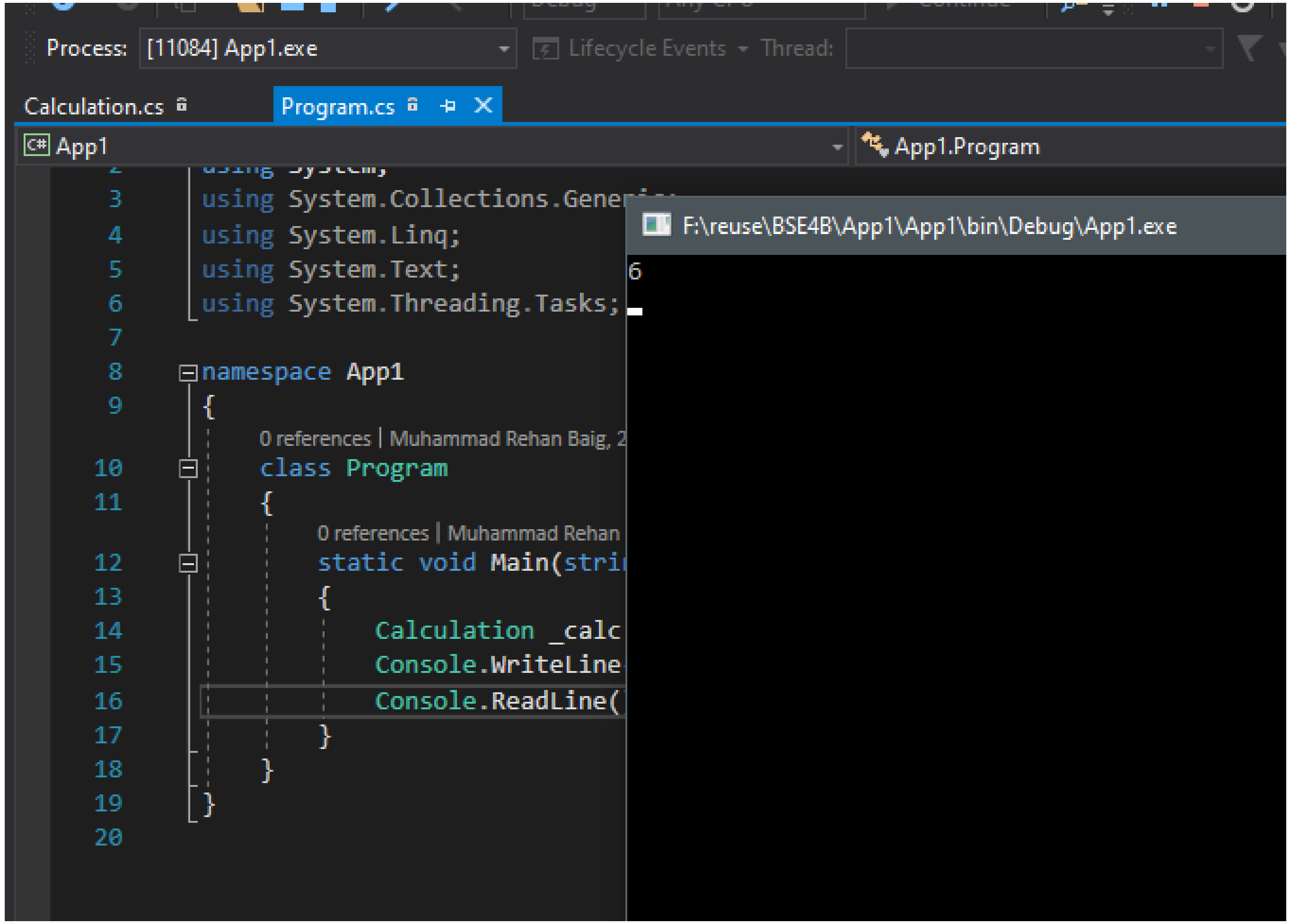
Step3 – Consuming App1 Accounting Calculation


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
App1.Program

```
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



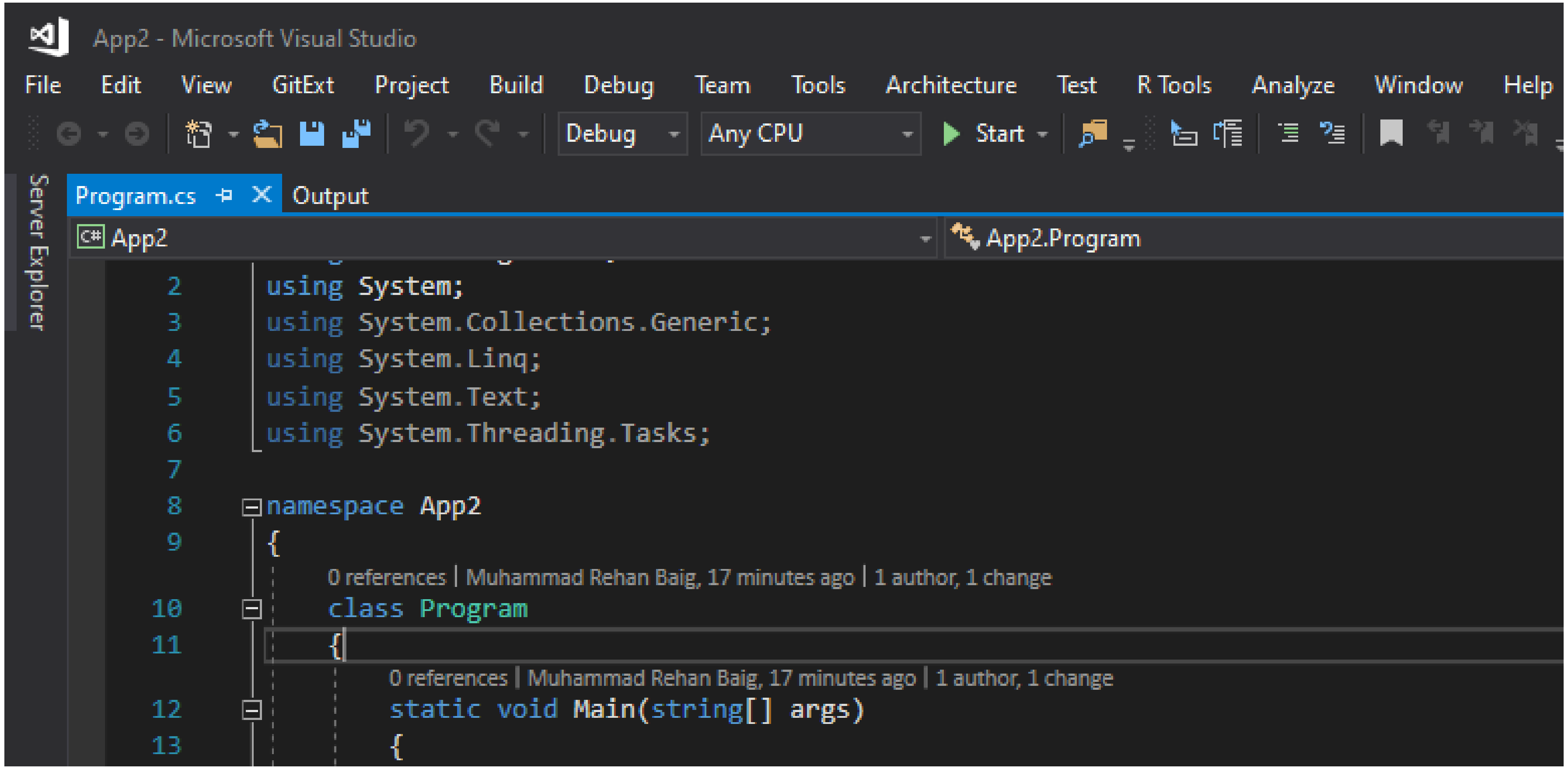
```
using System;
using System.Collections.Gener
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace App1
{
    0 references | Muhammad Rehan Baig, 2
    class Program
    {
        0 references | Muhammad Rehan
        static void Main(stri
        {
            Calculation _calc
            Console.WriteLine
            Console.ReadLine(
        }
    }
}
```

F:\reuse\BSE4B\App1\App1\bin\Debug\App1.exe

```
6
```

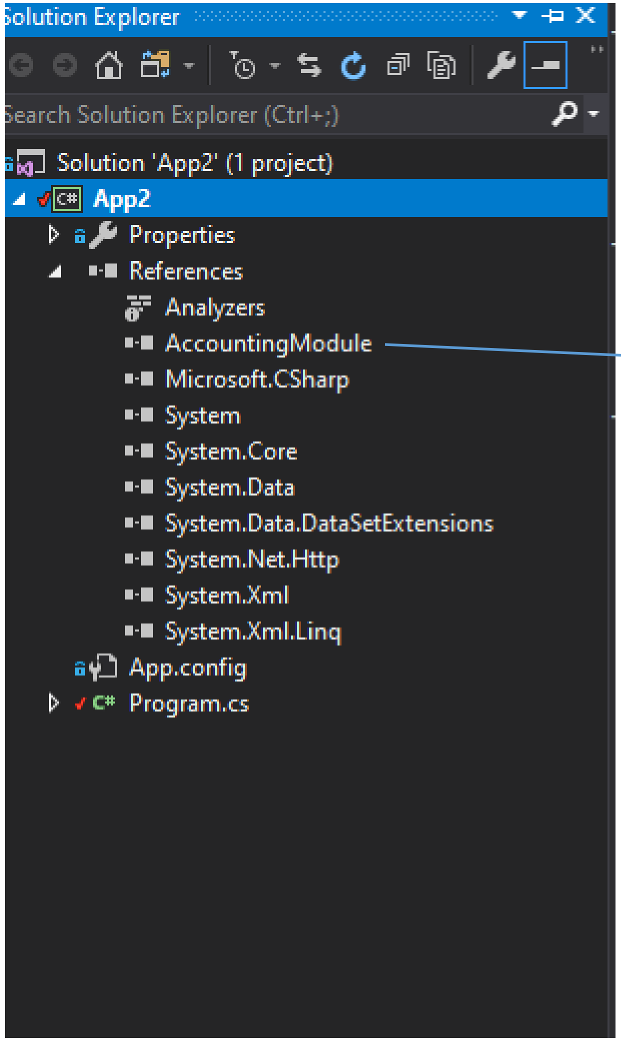
Step4 – Creating 2nd Project



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace App2
{
    0 references | Muhammad Rehan Baig, 17 minutes ago | 1 author, 1 change
    class Program
    {
        0 references | Muhammad Rehan Baig, 17 minutes ago | 1 author, 1 change
        static void Main(string[] args)
        {
```

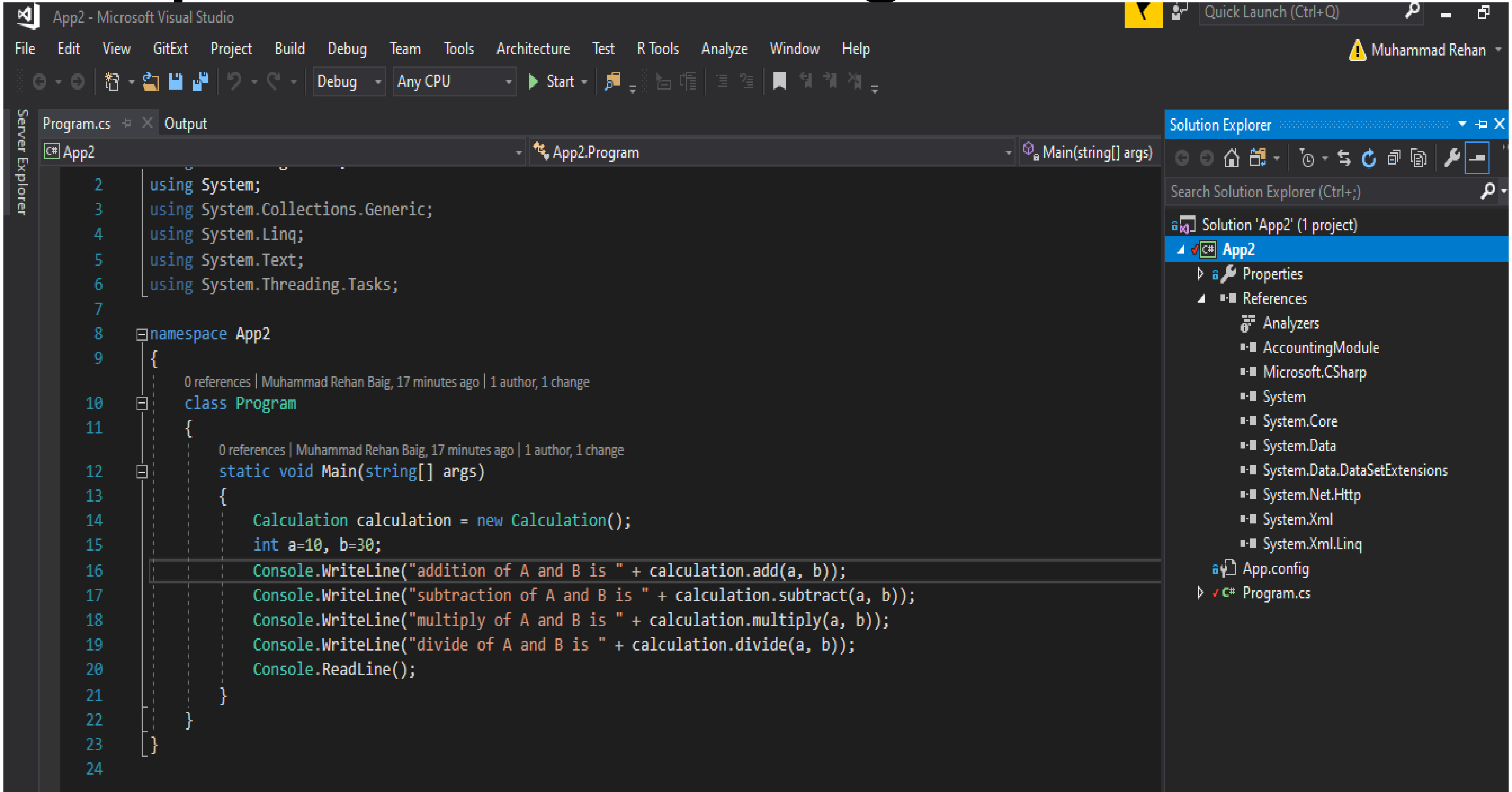
Step5 – Adding DLL (References) for Reuse of Calculation class



- Solution 'App2' (1 project)
  - App2
    - Properties
    - References
      - Analyzers
      - AccountingModule
      - Microsoft.CSharp
      - System
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Net.Http
      - System.Xml
      - System.Xml.Linq
    - App.config
    - Program.cs

Adding Accounting from Project 1
As a Reference DLL
Shared Library

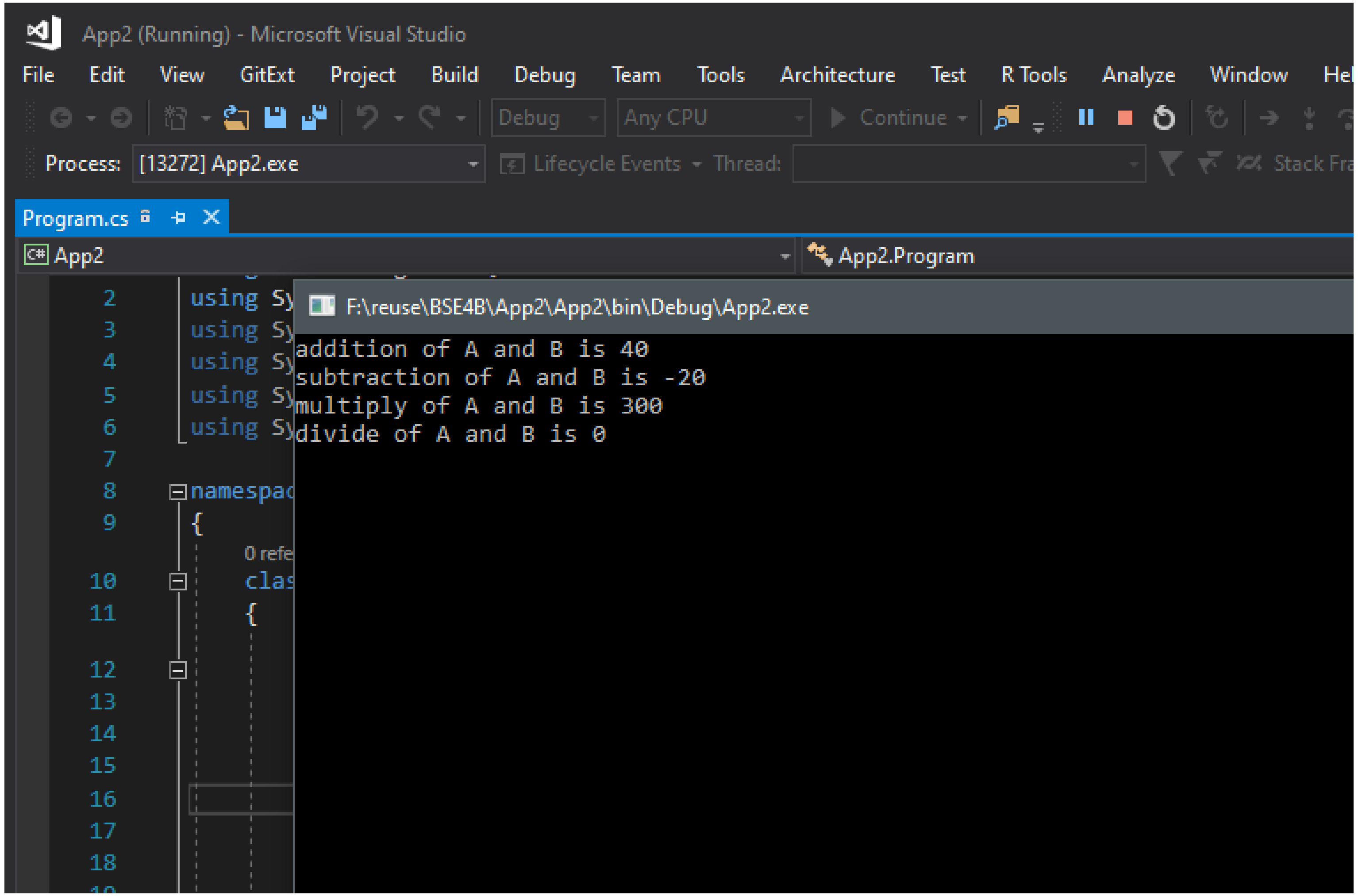
Step5 – Reuse Existing Code



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace App2
{
    0 references | Muhammad Rehan Baig, 17 minutes ago | 1 author, 1 change
    class Program
    {
        0 references | Muhammad Rehan Baig, 17 minutes ago | 1 author, 1 change
        static void Main(string[] args)
        {
            Calculation calculation = new Calculation();
            int a=10, b=30;
            Console.WriteLine("addition of A and B is " + calculation.add(a, b));
            Console.WriteLine("subtraction of A and B is " + calculation.subtract(a, b));
            Console.WriteLine("multiply of A and B is " + calculation.multiply(a, b));
            Console.WriteLine("divide of A and B is " + calculation.divide(a, b));
            Console.ReadLine();
        }
    }
}
```

Output



App2 (Running) - Microsoft Visual Studio
File Edit View GitExt Project Build Debug Team Tools Architecture Test R Tools Analyze Window Hel
Debug
Any CPU
Continue
Process: [13272] App2.exe
Lifecycle Events
Thread:
Stack Fra
Program.cs
App2
App2.Program
F:\reuse\BSE4B\App2\App2\bin\Debug\App2.exe

```
using Sy
using Sy
using Sy
using Sy
using Sy

namespac
{
    0 refe
    clas
    {
```

```
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.