Calculator Library

The Calculator Library project will be the main functional part of this solution. It will be a very simple .NET class library (DLL).

Create the project.

```
PS C:\Tools\mycalculator> dotnet new classlib -n Calculator
The template "Class Library" was created successfully.
```

Then add it to the solution.

```
PS C:\Tools\mycalculator> dotnet sln add .\Calculator\
Project `Calculator\Calculator.csproj` added to the solution.
```

Move into the Calculator directory and inspect the project files.

```
PS C:\Tools\mycalculator> cd Calculator
PS C:\Tools\mycalculator\Calculator> ls

Directory: C:\Tools\mycalculator\Calculator

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----          14/11/2022    14:40                obj
-a---          14/11/2022    14:40            215 Calculator.csproj
-a---          14/11/2022    14:40             55 Class1.cs
```

Rename `Class1.cs` to `Calculator.cs` and then open it for editing (I'm using VS Code, but any text editor will work).

```
PS C:\Tools\mycalculator\Calculator> move .\Class1.cs .\Calculator.cs
PS C:\Tools\mycalculator\Calculator> code .
```

For simplicity, I'm just going to write two methods - Add and Subtract. Their function is self-explanatory.

```
namespace Calculator;

public class Calculator
{
    public int Add(int num1, int num2)
        => num1 + num2;

    public int Subtract(int num1, int num2)
        => num1 - num2;
}
```

We can now build the project to a `.dll` file.

```
PS C:\Tools\mycalculator\Calculator> dotnet build
MSBuild version 17.4.0-preview-22470-08+6521b1591 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
C:\Program Files\dotnet\sdk\7.0.100-
rc.2.22477.23\Sdks\Microsoft.NET.Sdk\targets\Microsoft.NET.RuntimeIdentifierInferen
ce.targets(257,5): message NETSDK1057: You are using a preview version of .NET. See: https://aka.ms/dotnet-support-
poli
cy [C:\Tools\mycalculator\Calculator\Calculator.csproj]
Calculator -> C:\Tools\mycalculator\Calculator\bin\Debug\net7.0\Calculator.dll

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:03.23
```

We can then use reflection in PowerShell Core to load the DLL and call its methods.

```
PS C:\Users\Daniel>
[System.Reflection.Assembly]::LoadFrom("C:\Tools\mycalculator\Calculator\bin\Debug\net6.0\Calculator.dll")

GAC    Version        Location
---    -------        --------
False  v4.0.30319     C:\Tools\mycalculator\Calculator\bin\Debug\net6.0\Calculator.dll

PS C:\Users\Daniel> $calc = New-Object Calculator.Calculator
PS C:\Users\Daniel> $calc.Add(5, 5)
10
```

Finally, commit this to GitLab.

```
PS C:\Tools\mycalculator> git add . && git commit -m "create calculator project"
PS C:\Tools\mycalculator> git push -u origin main
```