Visual Studio IDE, Visual Code & Git installation

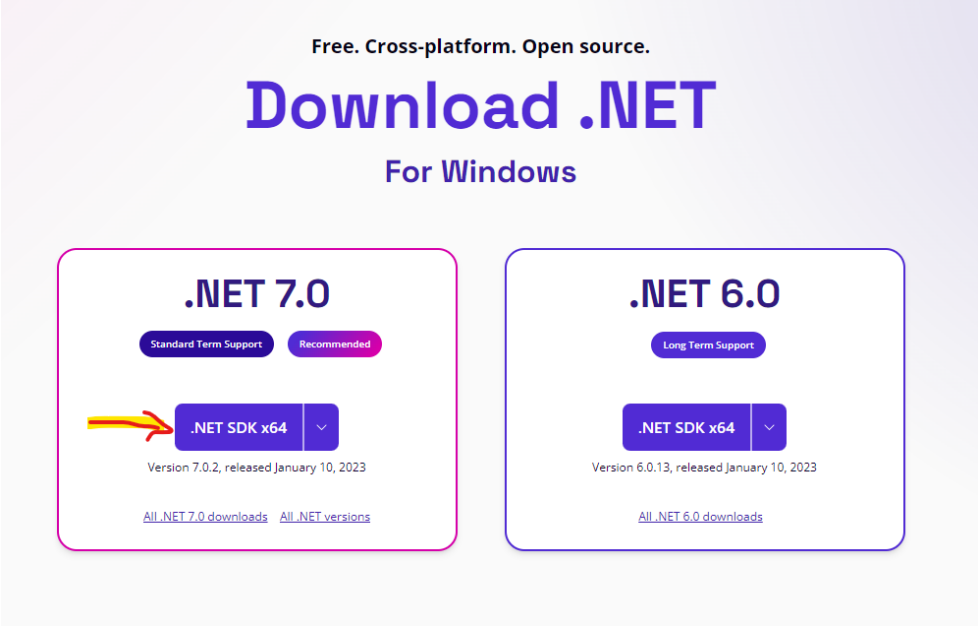
YKUMAR 2023-01-16

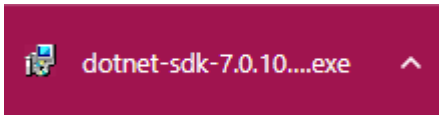
Note: if you already have Visual Studio, Visual Studio Code and GIT installed on your machine, run any C# program on both environments, take screenshot of your results, submit to Canvas and help others with the installation.

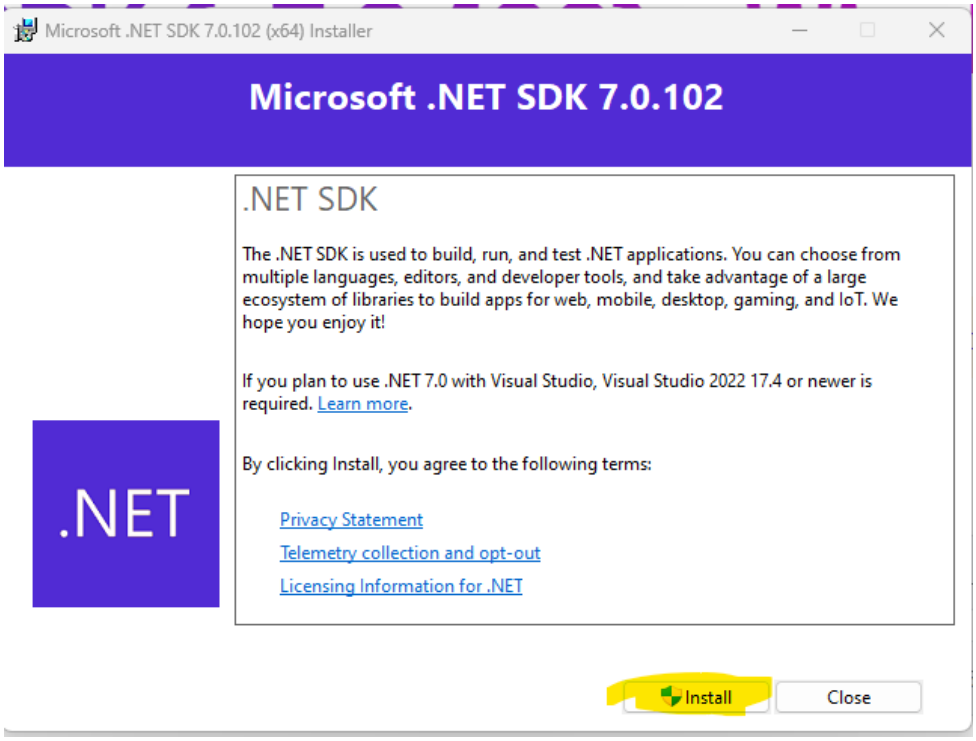
Contents

1. .NET 7 installation and Console App

Open the link https://dotnet.microsoft.com/download
Download according to your machine OS



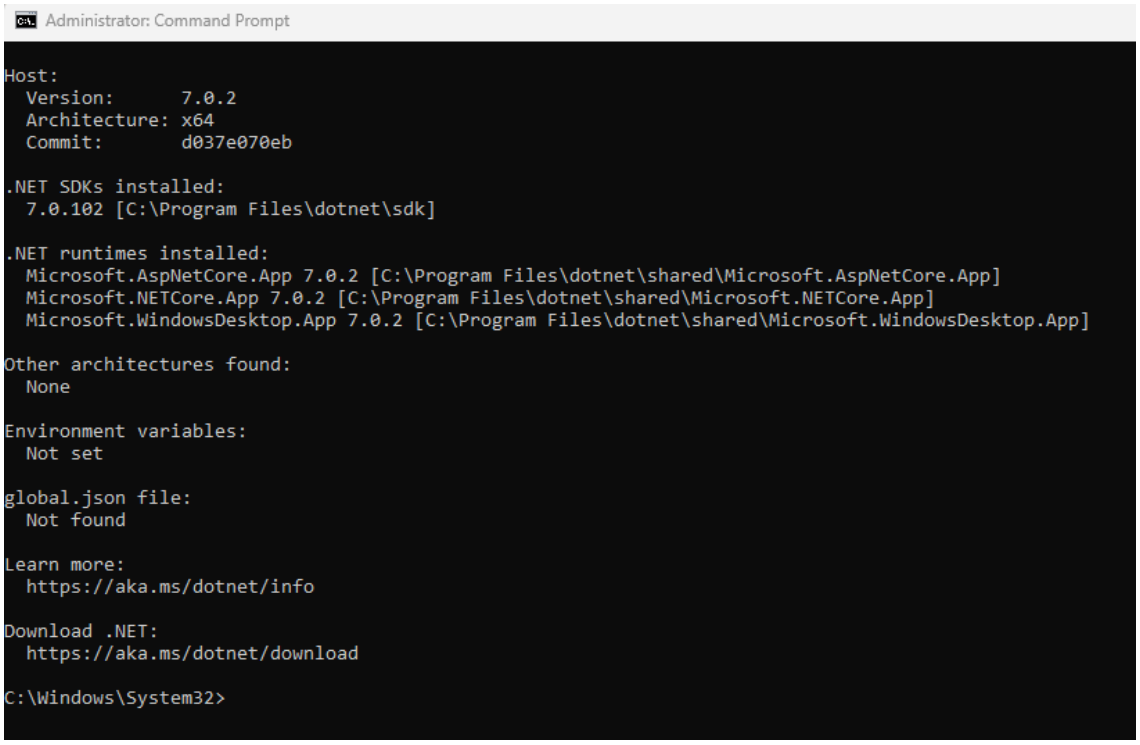
Download SDK, Installer appeared, run it 



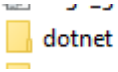
Click Install, **allow changes**, once done – Close, Type **cd** in search on your machine

For Windows: Open Command prompt as administrator, allow changes,
for Mac/Linux use Terminal and Linux commands (cd, ls, mkdir, etc)
In search type **cd**, choose to open Command Prompt as administrator

Type **dotnet --info**



Take a screenshot of your result

Type: create a new folder **dotnet** 

under C:\Users\yourUserName
Go there:

```
C:\Windows\System32>cd C:\Users\galae

C:\Users\galae>mkdir dotnet
```

Administrator: Command Prompt

```
C:\Users\galae>cd dotnet

C:\Users\galae\dotnet>dir
 Volume in drive C is Windows
 Volume Serial Number is E8A6-2481

 Directory of C:\Users\galae\dotnet

01/16/2023  02:53 AM    <DIR>          .
01/16/2023  02:53 AM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  194,061,385,728 bytes free

C:\Users\galae\dotnet>
```

Create a new Console C# app

Type command: **dotnet new console -o MyApp**

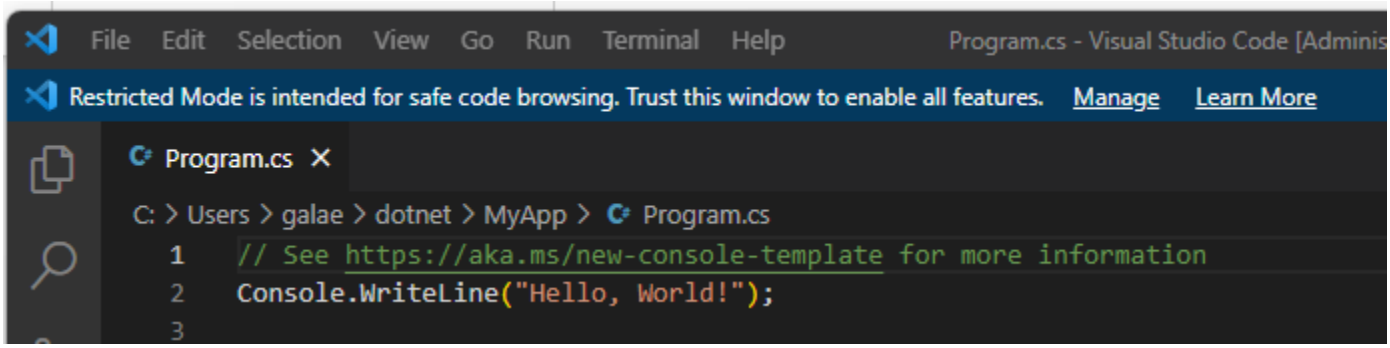
Go to the newly created folder: **cd MyApp**

Type **Program.cs**

```
C:\Users\galae\dotnet>cd MyApp

C:\Users\galae\dotnet\MyApp>Program.cs
```

The program below should appear:



Press on command prompt Enter.

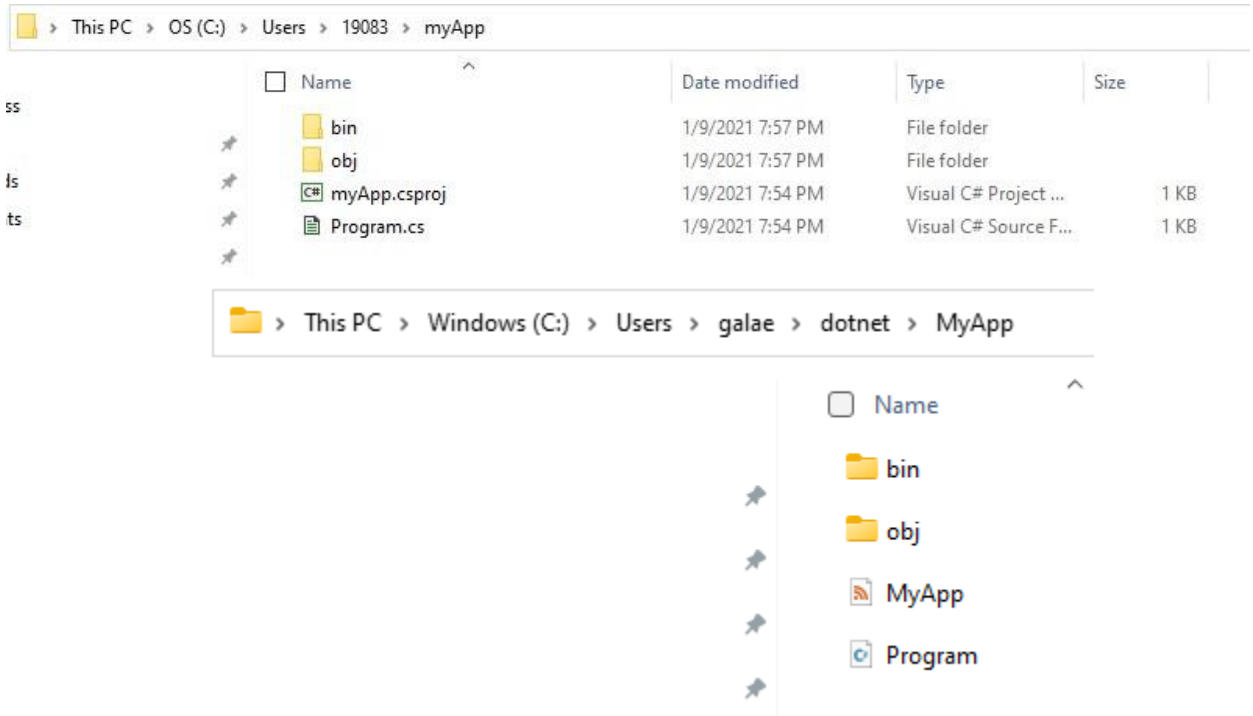
Type the command

dotnet run

```
C:\Users\galae\dotnet\MyApp>dotnet run
Hello, World!

C:\Users\galae\dotnet\MyApp>
```

You can further edit the Program.cs under C:\Users\yourUserName\myApp



-Add one more line of code to the program to print your name on Console.

-Add an additional line to your code and save the file:

```
Console.WriteLine("Version: {0}", Environment.Version.ToString());
```

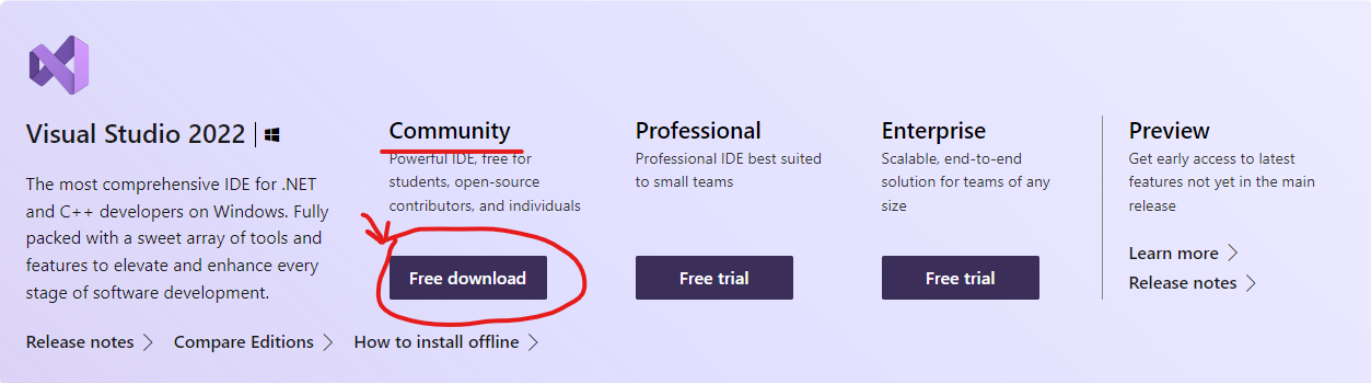
Do not forget to save the file

Take a screenshot of your code and output. Make sure the version starts with 7.

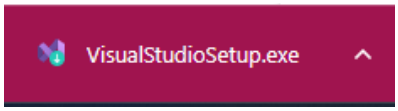
TASK 1. Provide screenshot from console / terminal proving that .NET 7 is installed on your machine. Provide source code and output of your Program.cs, running on Console (Command Prompt or Terminal)

2. Visual Studio 2022 installation

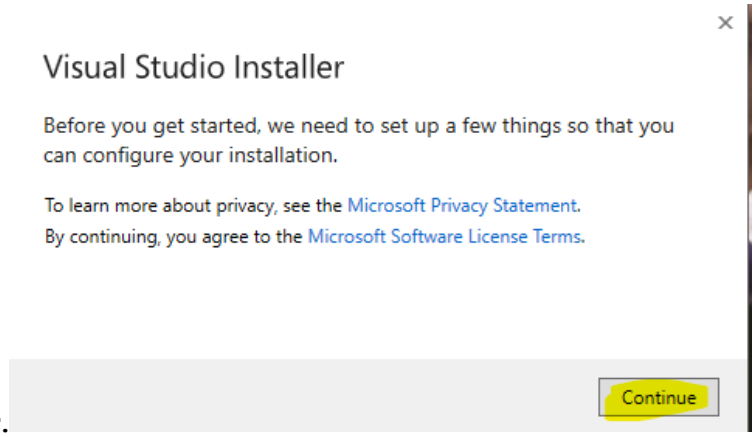
Open the link https://visualstudio.microsoft.com/downloads/

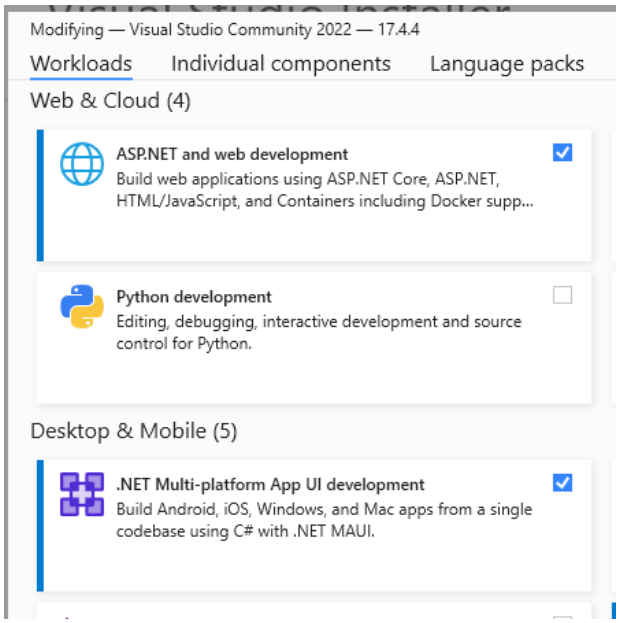


Some instructions for Mac are attached in a separate file. For Windows: **Click Download** the Community edition, free

Your installer file will be downloaded 

Run it, allow make changes on your computer.





Check 2 checkboxes, click **install**. It is recommended **to download all first and then install it.**

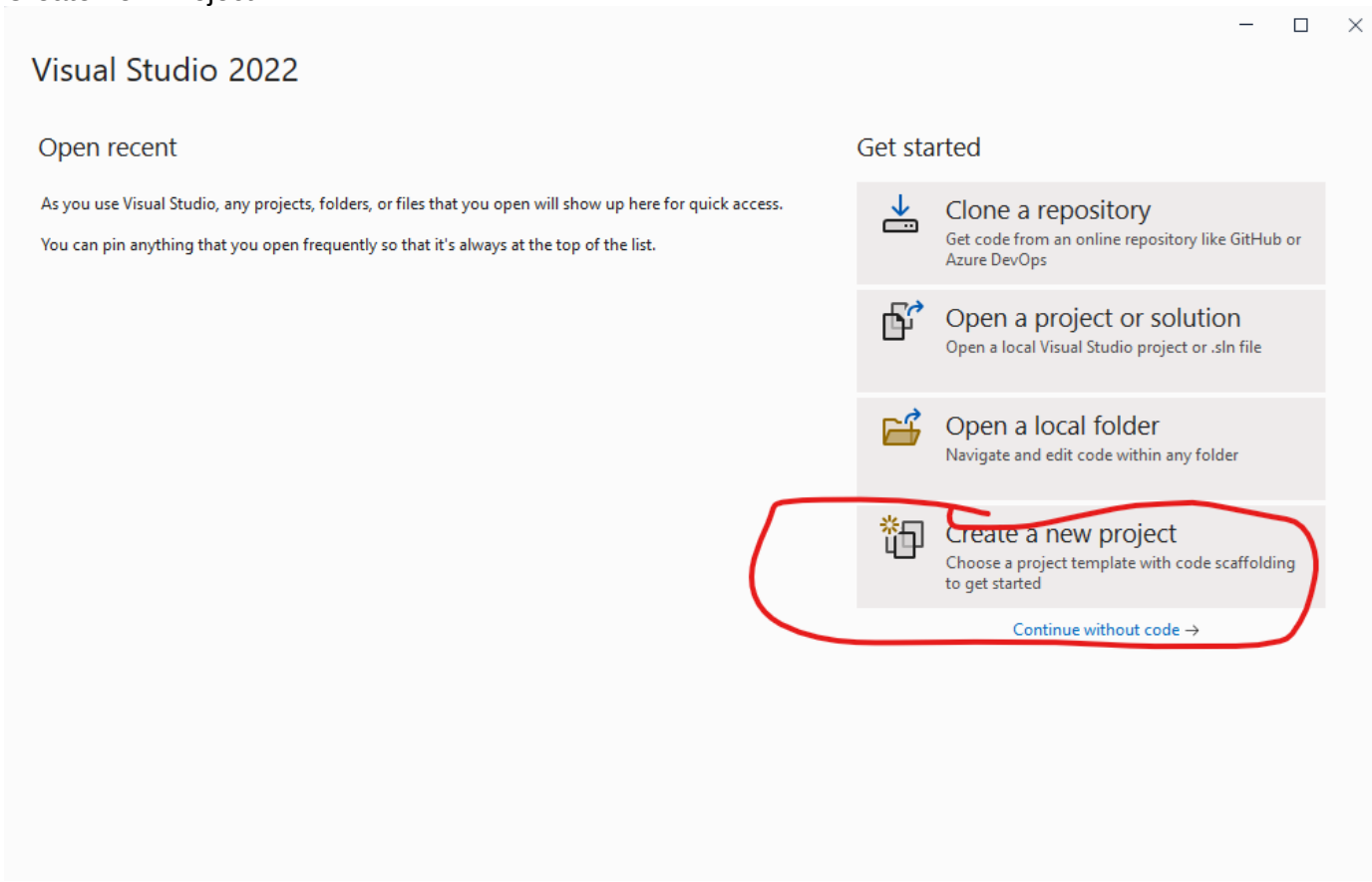
The installation will take some time, once done - **Launch**.

You can create an account or skip for now.

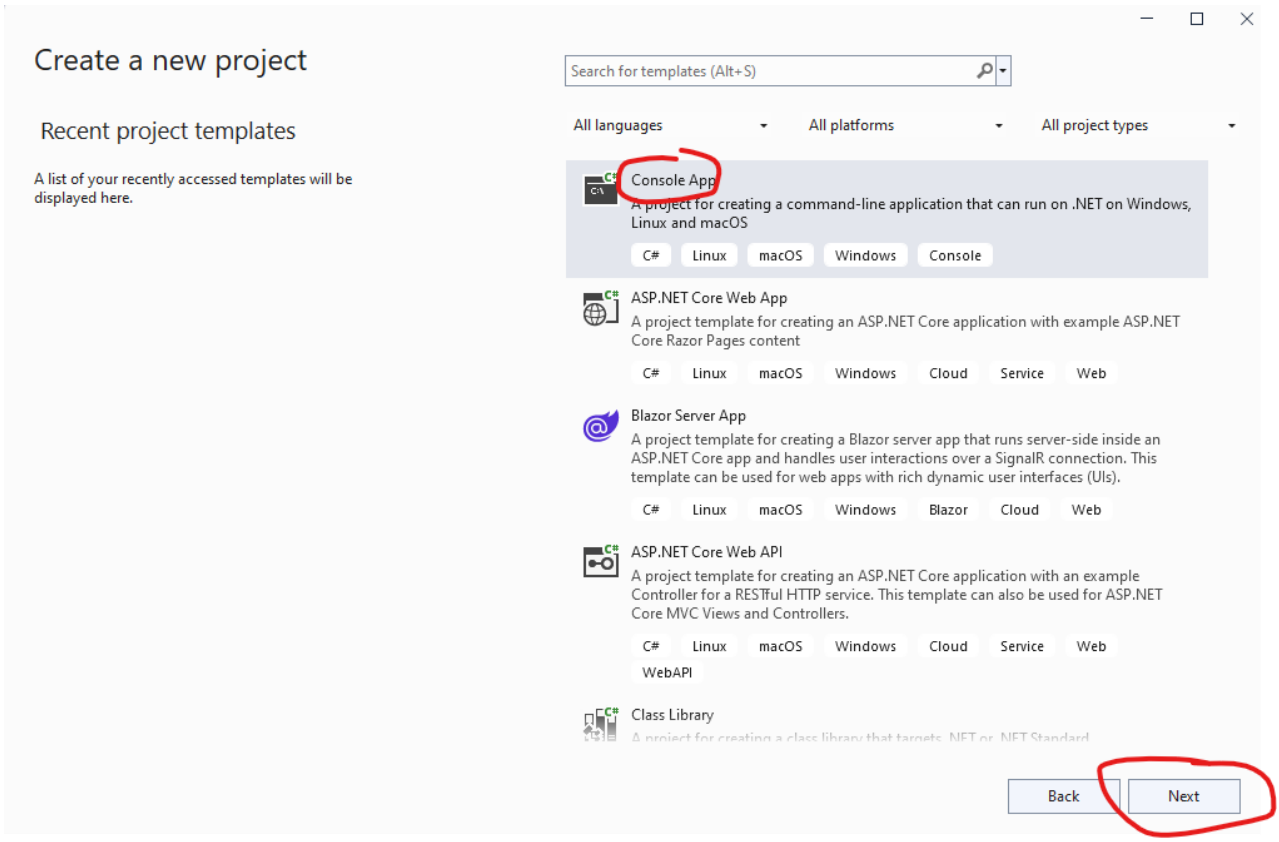
Choose a background you like

& Start Visual studio.

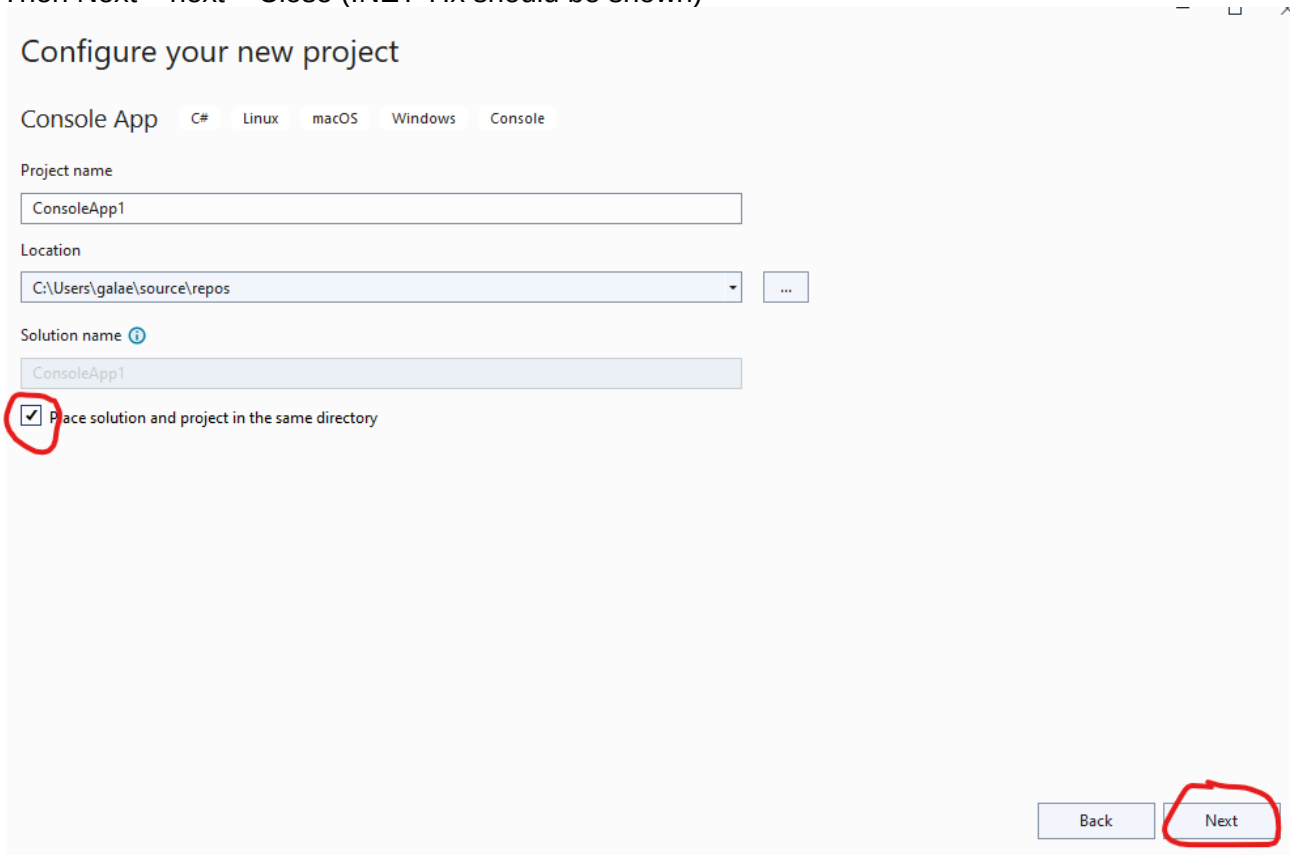
Create new Project

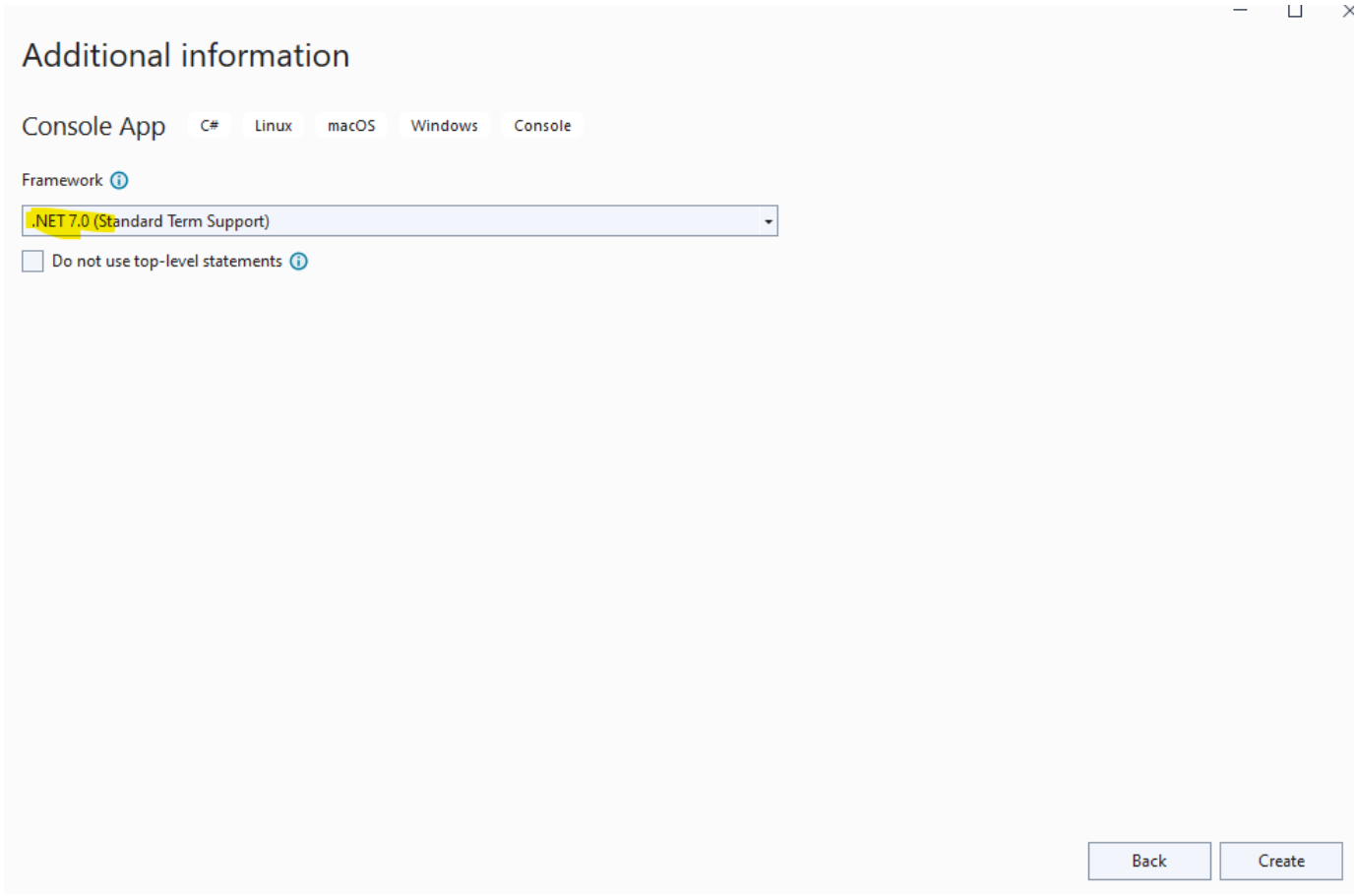


Choose **Console App**



Then Next – next – Close (.NET 7.x should be shown)

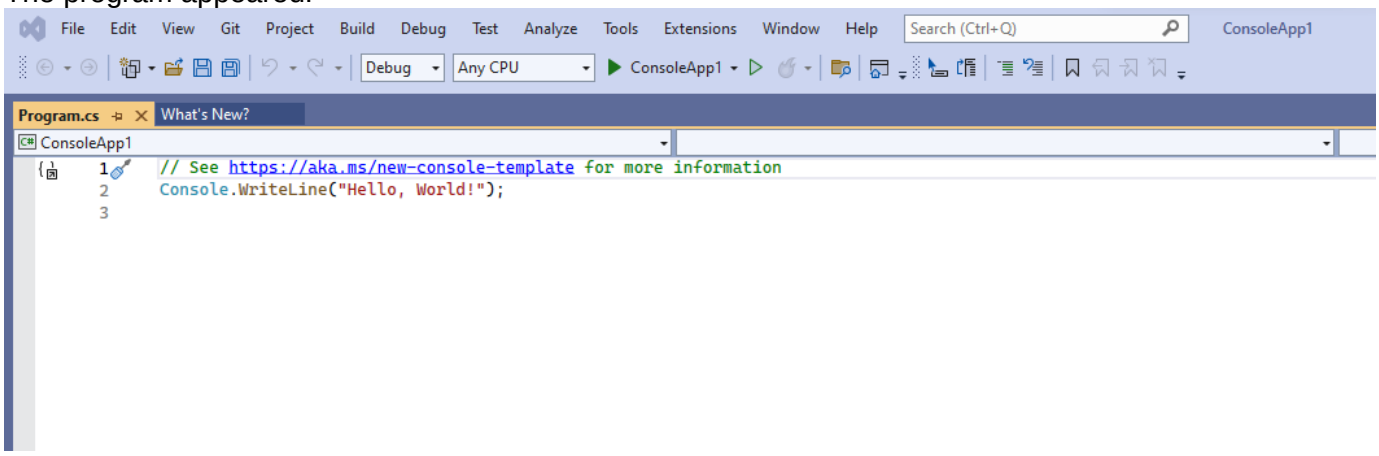




Click **Create**.

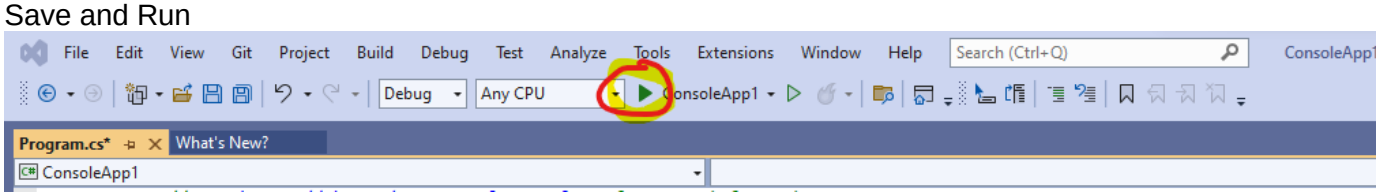
.

The program appeared:

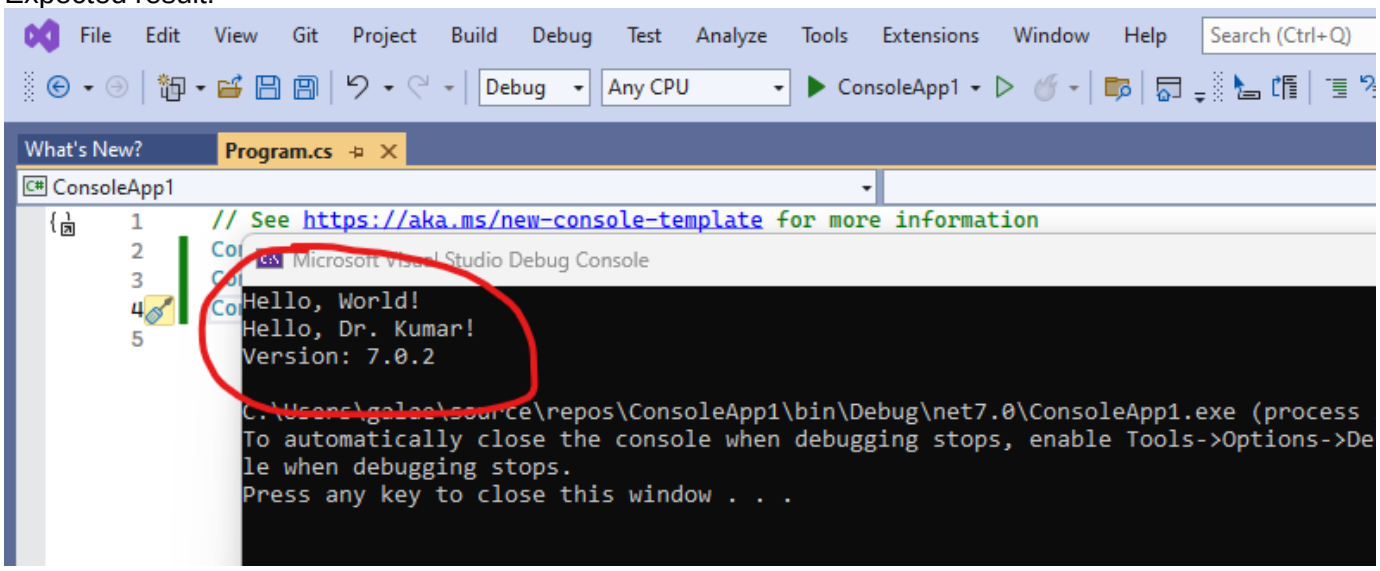


Add the same lines of code we added to Console (the output should have your name) & save:

Save and Run



Expected result:



Congratulations! Your first C# program runs.

We will add some code from Open Educational resource (OER) W3Schoolds

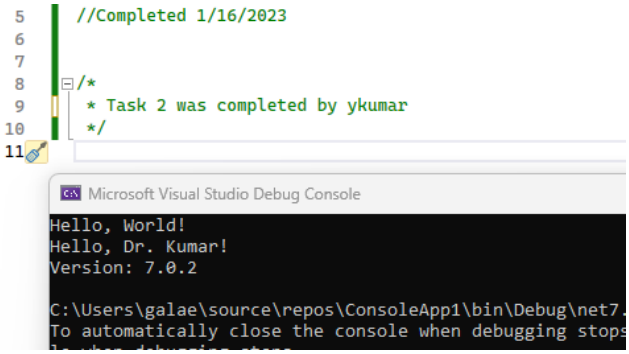
a) Open https://www.w3schools.com/cs/cs_comments.php, skim through the page

And add a single line comment to your program – today's date

Add a multiline comment stating that TASK 2 is complete by yourName.

Make sure your code still runs

For example:



b) Open https://www.w3schools.com/cs/cs_variables.php, skim through the page

Add any of the code snippets or all of them to your page. You can add your own example with variables. Make sure the app still runs

c) Open https://www.w3schools.com/cs/cs_arrays, skim through the page

We will create our own array:

Example:

```
/*
 * Task 2 was completed by ykumar
 */
int myNum = 15;
Console.WriteLine("myNum was: " + myNum);
myNum = 20; // myNum is now 20
Console.WriteLine("myNum is now: " + myNum);

string[] sections = { "CPS*3330*01", "CPS*3330*03", "TECH*4982*01"};

//print the length
Console.WriteLine("The number of sections is " + sections.Length);

//reverse it
Array.Reverse(sections);
Console.Write("\nIn reverse: ");

//print them all using foreach loop
foreach (var e in sections)
    Console.Write(e.ToString() + " ");

Console.WriteLine("\nUsing foreach method:");
sections.ToList().ForEach(e => Console.Write(e.ToString() + " "));

Console.WriteLine("\n\nUsing Array class and its method:");
Array.ForEach(sections, Console.WriteLine);
```

```
Microsoft Visual Studio Debug Console
Hello, World!
Hello, Dr. Kumar!
Version: 7.0.2
myNum was: 15
myNum is now: 20
The number of sections is 3

In reverse: TECH*4982*01 CPS*3330*03 CPS*3330*01
Using foreach method:
TECH*4982*01 CPS*3330*03 CPS*3330*01

Using Array class and its method:
TECH*4982*01
CPS*3330*03
CPS*3330*01

C:\Users\galae\source\repos\ConsoleApp1\bin\Debug\r
To automatically close the console when debugging s
le when debugging stops.
Press any key to close this window . . .
```

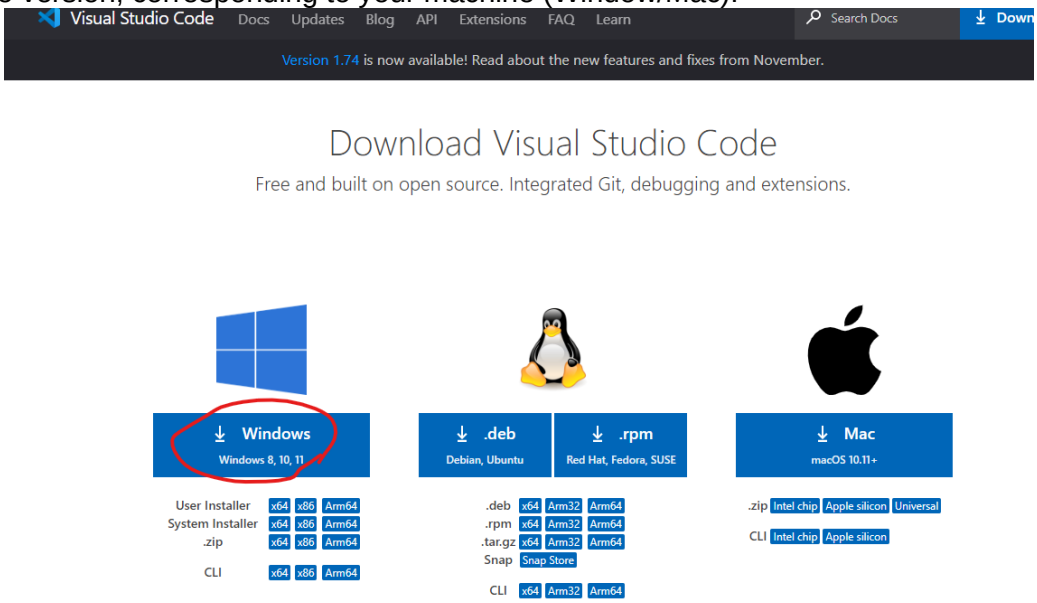
Right-click on the icon at the bottom of the screen – **Pin Visual Studio icon** to taskbar 

TASK 2. Provide Source code and output of your program, created in Visual studio Preview (your name should be displayed in both)

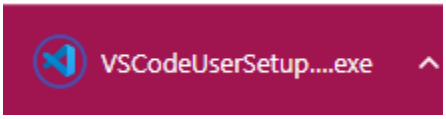
3. Visual Studio Code installation

Open the link https://code.visualstudio.com/download

Download the version, corresponding to your machine (Window/Mac).



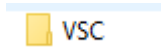
For Windows: Double click, accept the agreement , proceed with defaults (do not change anything but click next) unless changes are preferred.



Install and Launch

It is very important to open & select a proper folder on your machine in VSC,

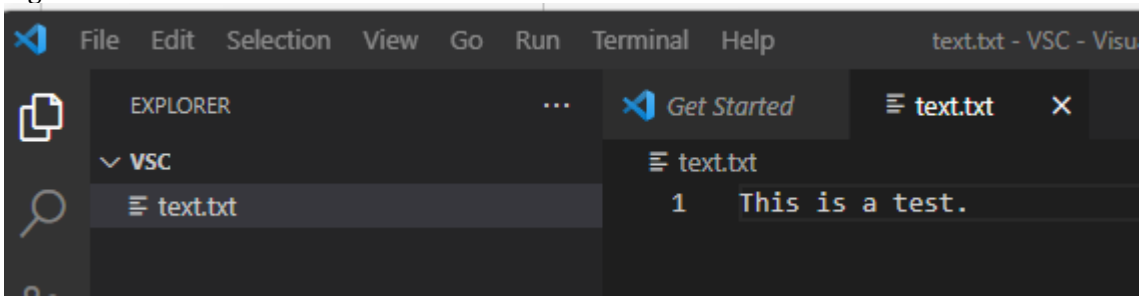
you can create a new folder VSC under Users/yourUserName manually and select it



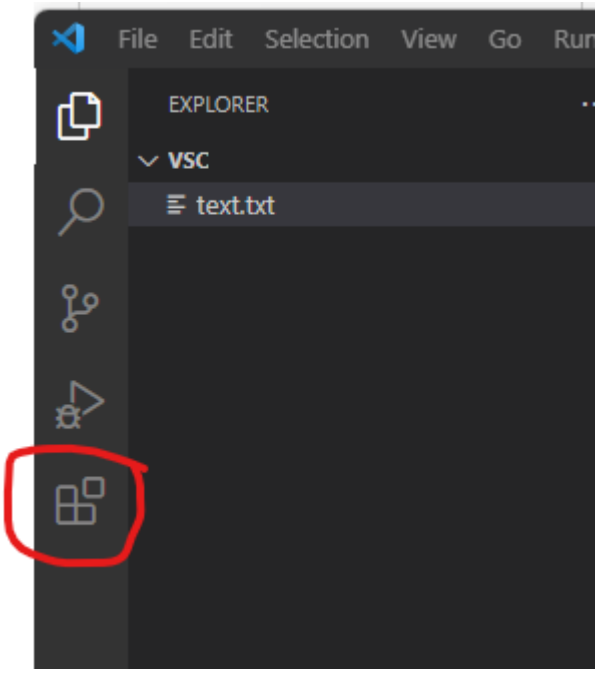
Launch VSC, choose to **Open folder**, browse to it and then **Select the folder**
Trust the authors if asked to confirm

Stand on VSC folder, choose to create a new file (the first icon to the right), name it text.txt, add there the text "This is a test".

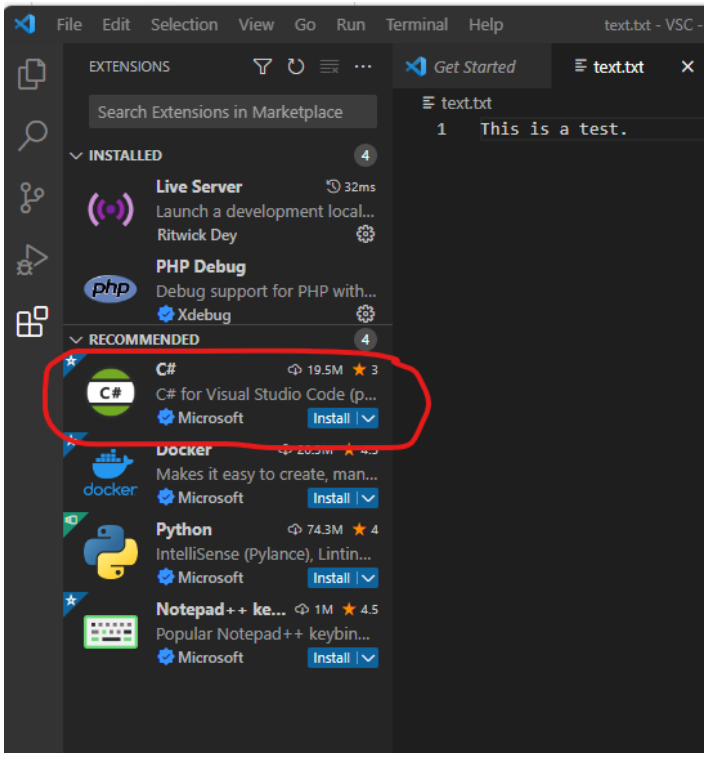
Save it trough File -> Save



File -> Save
Find extensions icon under documents.

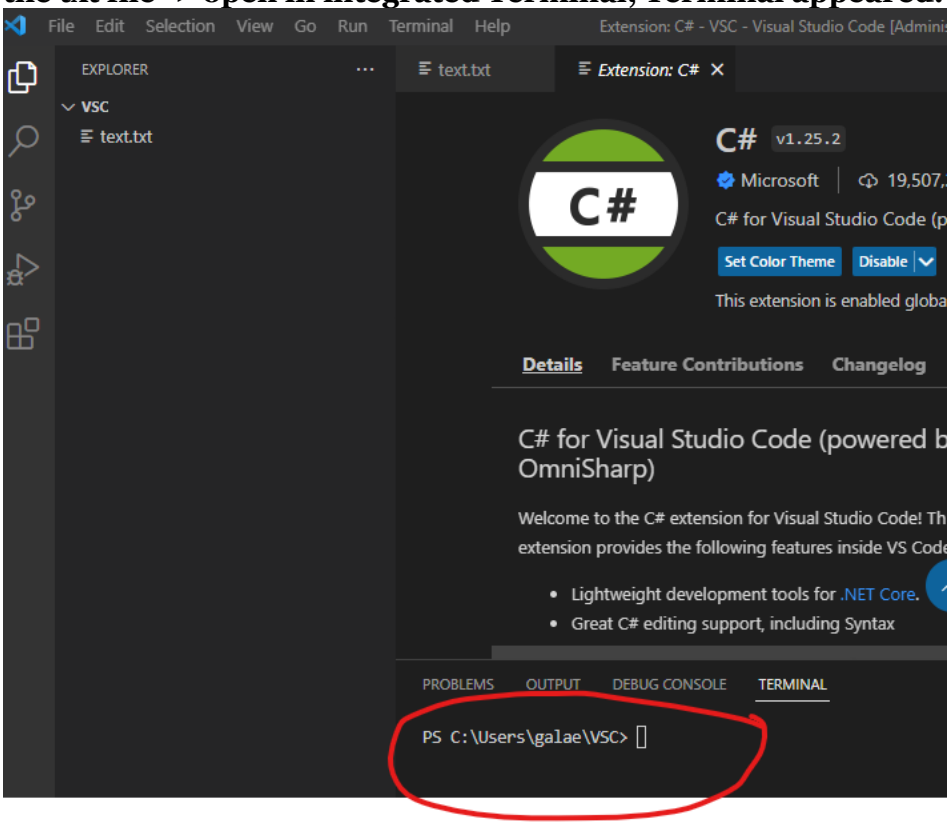


Search for and install C# extension



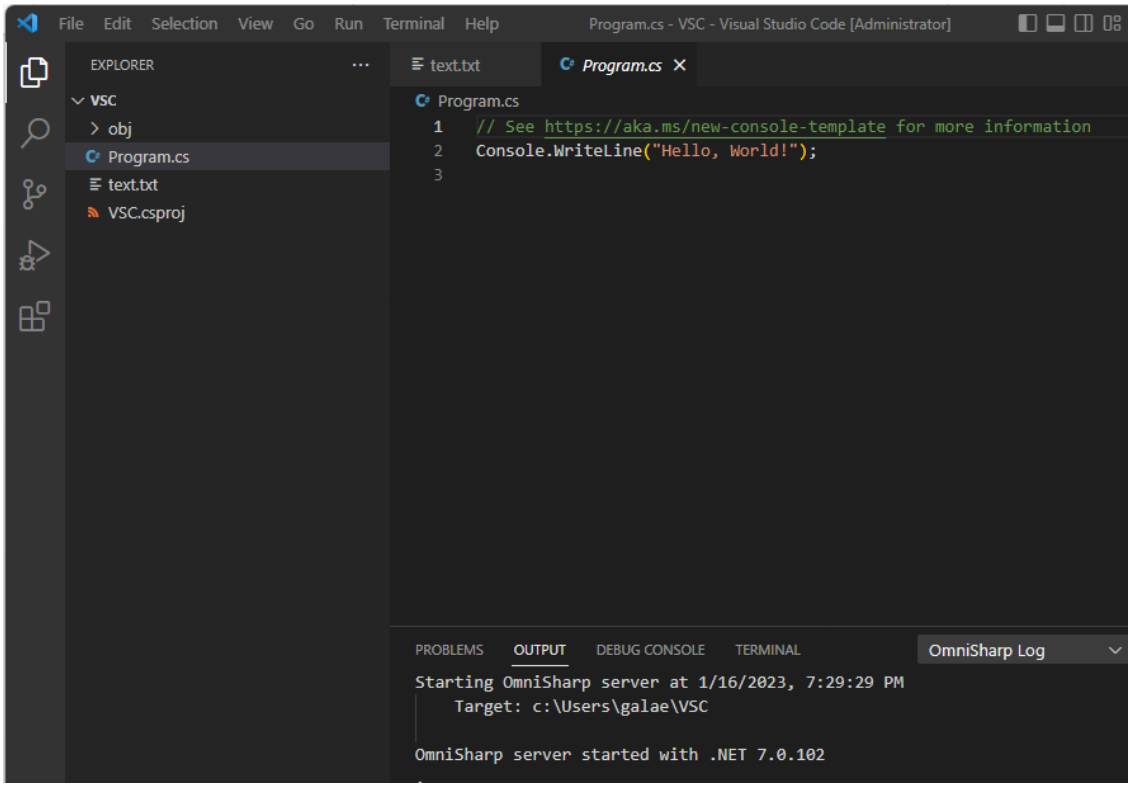
Go back to files 

Right click on the txt file -> open in integrated Terminal, Terminal appeared:



Create a new Console App:

Type **dotnet new console**



Go to Program.cs

- Add two lines we added on the previous tasks (your name and version)
- Also Add 2 following lines:

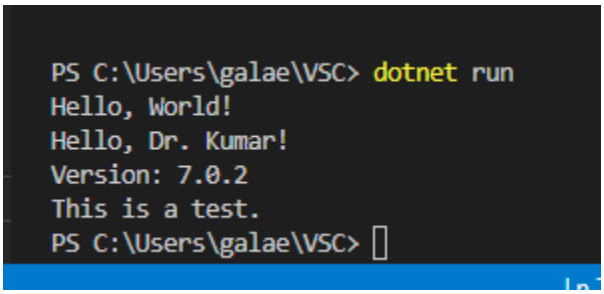
string text = System.IO.File.ReadAllText("text.txt");

Console.WriteLine(text);

Save the file

Type on the terminal: **dotnet run**

Expected result



Make sure the code shows .NET 7

Find VSC in the search / on the Taskbar and **pin in to Taskbar**

TASK 3. Provide Source code and output of your program, created in VSC

4. Download Git and Create a GitHub account

Sign in or sign up for GitHub for free. Create a GitHub account at https://github.com/

Install Git from https://git-scm.com/downloads.

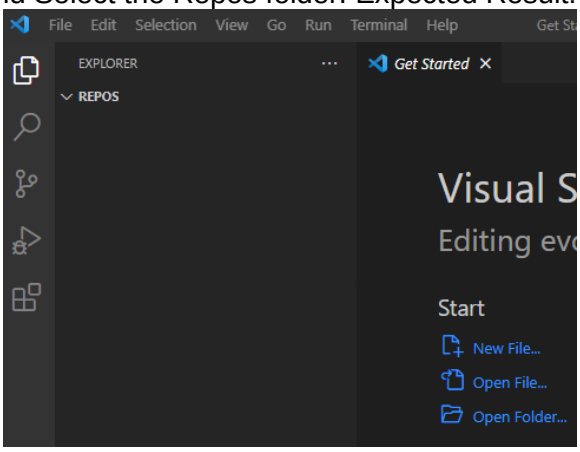
Double click on the executable file and start installation. Allow the application make changes on your machine and accept the license. Proceed with defaults (click **next** many times without changing anything, then install).

TASK 4. Provide screenshots proving that part 4 was completed

5. Cloning the book repositories

Create a folder named **Repos** under your user.

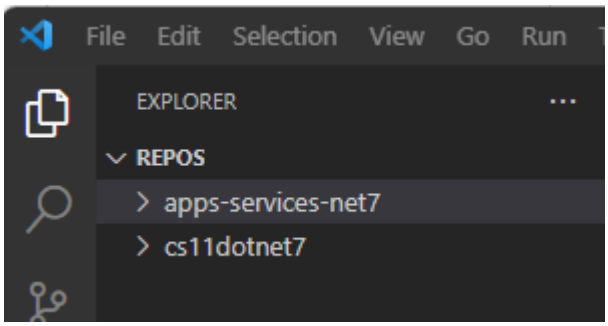
In Visual Studio Code, open and Select the Repos folder. Expected Result:



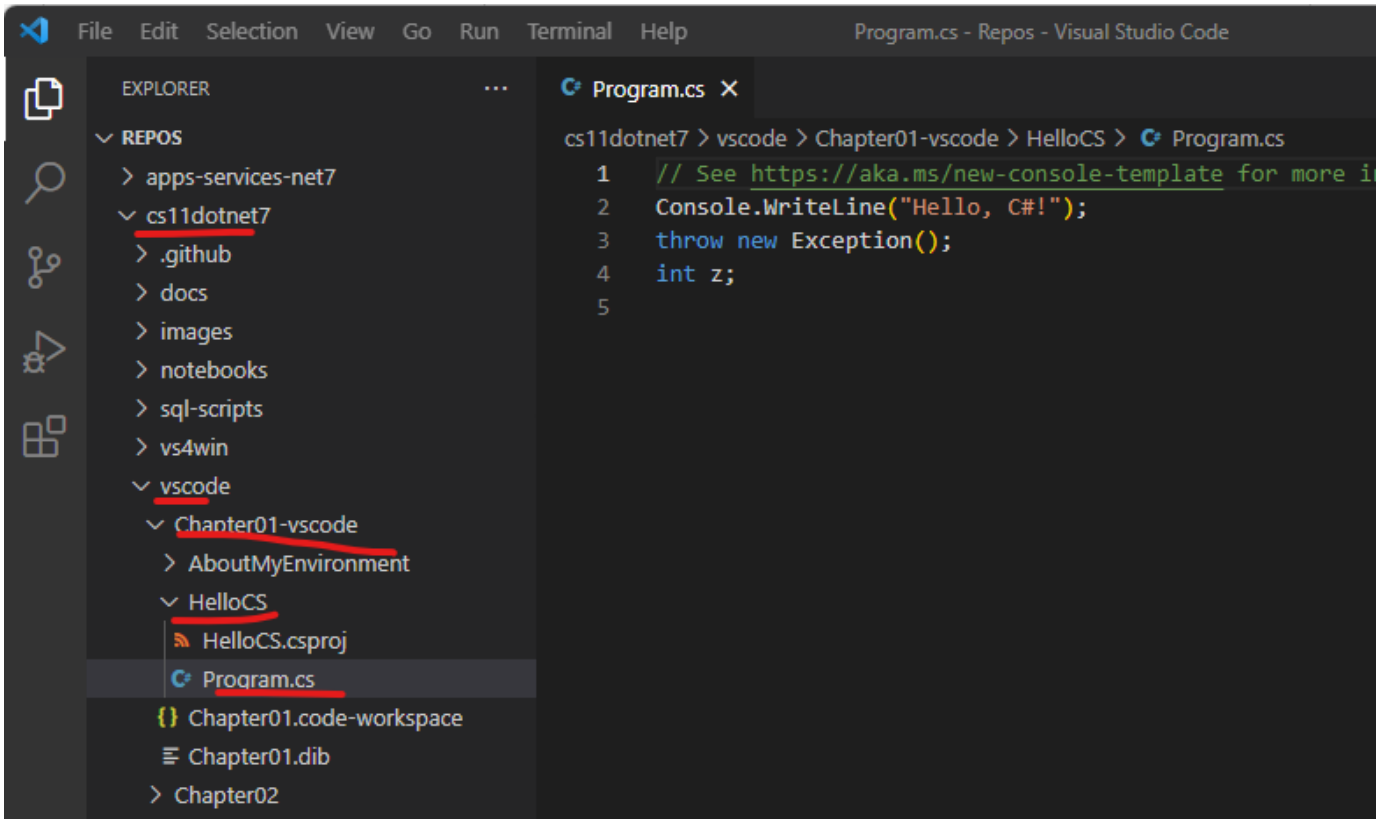
Navigate to View | Terminal, and enter the following commands

```
git clone https://github.com/markjprice/cs11dotnet7
git clone https://github.com/markjprice/apps-services-net7
```

Expected result



Open **cs11dotnet7**
Go to



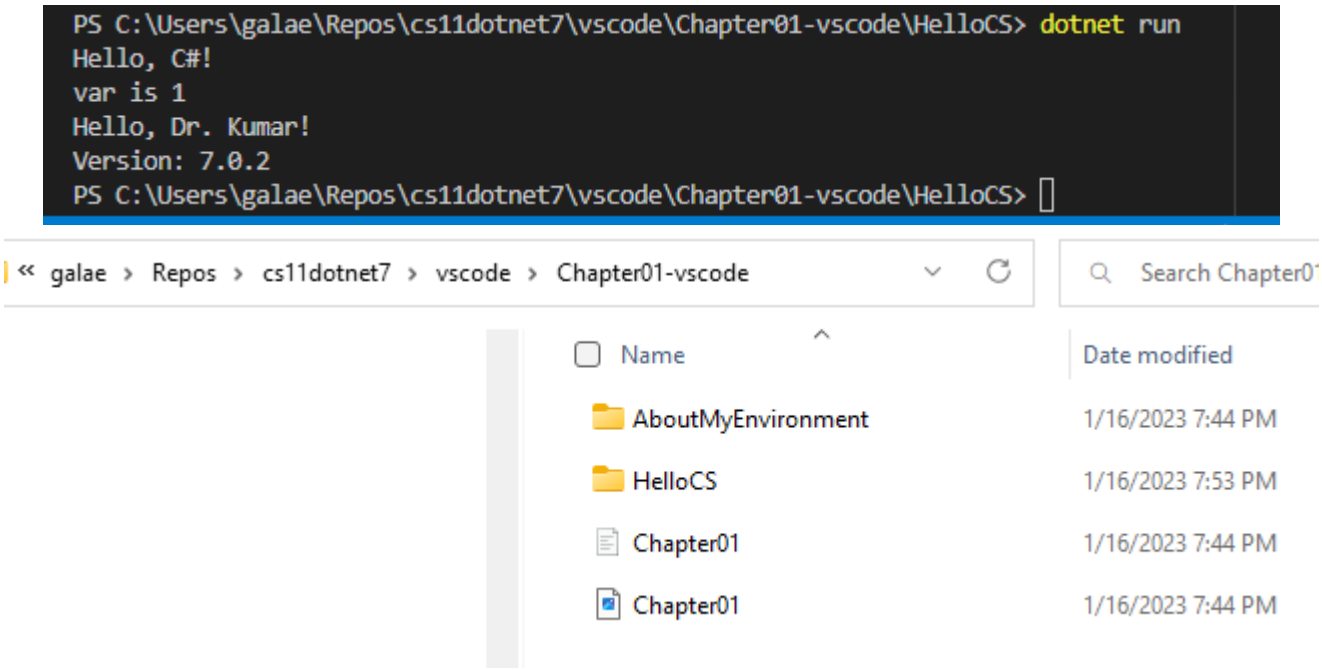
Add the same two lines we added at previous steps to this file

Comment out line 3, assign and print z or any other variable of your choice.

Right-click on Program.cs -> Open in Integrated Terminal

Run the program (Do not forget to save)

Take a screenshot of your source code and output, verify that filed appeared under **Repos** folder

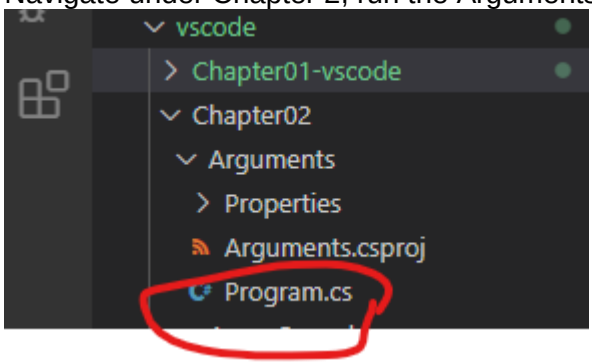


Take a screenshot

TASK 5. Provide screenshots proving that part 5 was completed

6. Test both books code

Navigate under Chapter 2, run the Arguments Console App:



Open the Program.cs on integrated terminal. It takes 3 arguments like that:
dotnet run **red yellow 50**
Expected result:

```
PS C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter02\Arguments> dotnet run red yellow 50
C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter02\Arguments\Program.cs(27,3): w
arning CA1416: This call site is reachable on all platforms. 'Console.CursorSiz
e.set' is only supported on: 'windows'. [C:\Users\galae\Repos\cs11dotnet7\vscod
e\Chapter02\Arguments\Arguments.csproj]
There are 3 arguments.
red
yellow
50
PS C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter02\Arguments>
```

You can ignore the warning, it says that you can acyually cannot change the font size.

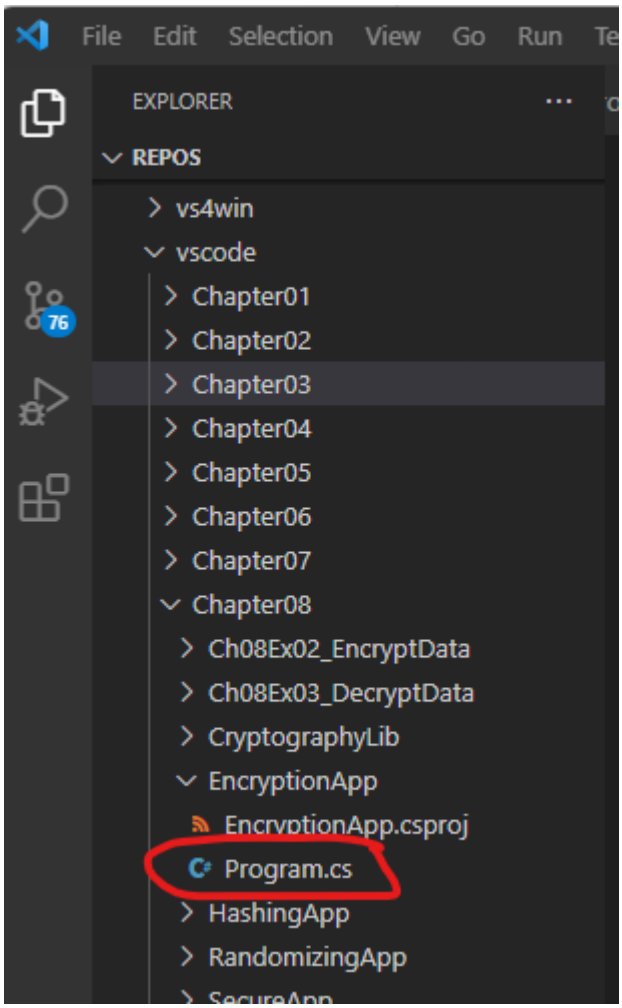
Chose any other colors and call the program again.

If there is a problem with .NET version, you should open .obj file and change manually version 6.0 (or 5.0) to 7.0, ignore this part as no problems appear.

cs11dotnet7 > vscode > Chapter02 > Arguments > Arguments.csproj

```
1  <Project Sdk="Microsoft.NET.Sdk">
2
3    <PropertyGroup>
4      <OutputType>Exe</OutputType>
5      <TargetFramework>net7.0</TargetFramework>
6      <ImplicitUsings>enable</ImplicitUsings>
7      <Nullable>enable</Nullable>
8    </PropertyGroup>
9
10   <ItemGroup>
11     <Using Include="System.Console" Static="true" />
12   </ItemGroup>
13
14
15 </Project>
16
```

Go to second repository
Navigate to chapter 8 and open **EnctyptionApp**



Open in integrated terminal and run the Program.cs on Console
Use your name as a message, use any password of your choice

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\Users\galae\Repos\apps-services-net7\vscode\Chapter08\EncryptionApp> dotnet run
Enter a message that you want to encrypt: ykumar
Enter a password: Bananas
```

Expected result:

```
PS C:\Users\galae\Repos\apps-services-net7\vscode\Chapter08\EncryptionApp> dotnet run
Enter a message that you want to encrypt: ykumar
Enter a password: Bananas
PBKDF2 algorithm: SHA256, Iteration count: 150,000
121 milliseconds to generate Key and IV.
Encryption algorithm: AES-256, CBC mode with PKCS7 padding.
Encrypted text: THwAcTYzTSvaCgeYg+FtRg==
Enter the password:
```

Use the same password

```
PS C:\Users\galae\Repos\apps-services-net7\vscode\Chapter08\EncryptionApp> dotnet run
Enter a message that you want to encrypt: ykumar
Enter a password: Bananas
PBKDF2 algorithm: SHA256, Iteration count: 150,000
121 milliseconds to generate Key and IV.
Encryption algorithm: AES-256, CBC mode with PKCS7 padding.
Encrypted text: THwAcTYzTSvaCgeYg+FtRg==
Enter the password: Bananas
Decrypted text: ykumar
PS C:\Users\galae\Repos\apps-services-net7\vscode\Chapter08\EncryptionApp>
```

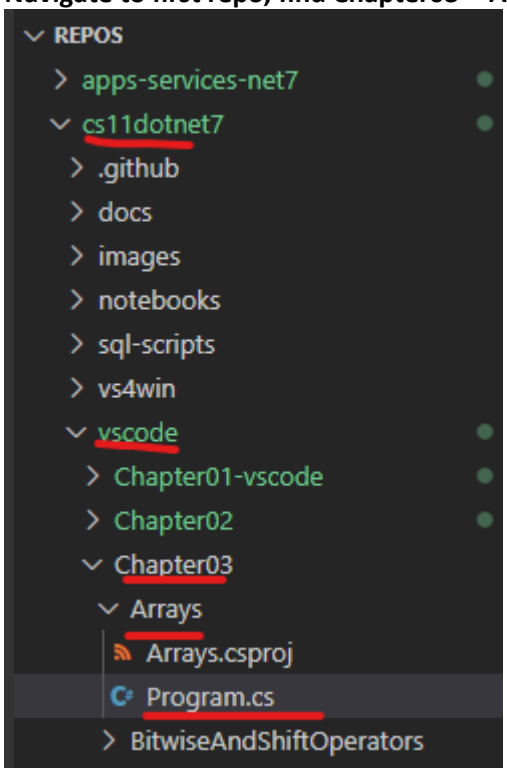
Take screenshots of all source code and output used in this Task.

TASK 6. Provide screenshots proving that part 6 was completed

7. Push code into GitHub

Log in into your Github Account
https://github.com/login

Navigate to first repo, find Chapter03 – Arrays App



Attempt to run the code. Take screenshot
Open .csproj file and change 6.0 to 7.0

Program.cs ...\AboutMyEnvironment | Program.cs ...\Arguments | Argume

cs11dotnet7 > vscode > Chapter03 > Arrays > Arrays.csproj

```
<Project Sdk="Microsoft.NET.Sdk">

    <PropertyGroup>
        <OutputType>Exe</OutputType>
        <TargetFramework>net7.0</TargetFramework>
        <ImplicitUsings>enable</ImplicitUsings>
        <Nullable>enable</Nullable>
    </PropertyGroup>

    <ItemGroup>
        <Using Include="System.Console" Static="true" />
    </ItemGroup>

</Project>

```

Expected output:

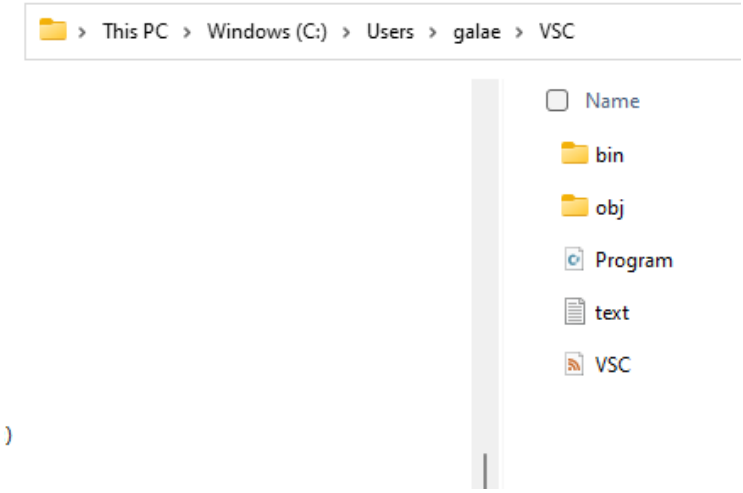
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter03\Arrays> dotnet run
Kate
Jack
Rebecca
Tom
Lower bound of the first dimension is: 0
Upper bound of the first dimension is: 2
Lower bound of the second dimension is: 0
Upper bound of the second dimension is: 3
Row 0, Column 0: Alpha
Row 0, Column 1: Beta
Row 0, Column 2: Gamma
Row 0, Column 3: Delta
Row 1, Column 0: Anne
Row 1, Column 1: Ben
Row 1, Column 2: Charlie
Row 1, Column 3: Doug
Row 2, Column 0: Aardvark
Row 0, Column 1: Beta
Row 0, Column 2: Gamma
Row 1, Column 0: Anne
Row 1, Column 1: Ben
Row 1, Column 2: Charlie
Row 1, Column 3: Doug
Row 2, Column 0: Aardvark
Row 2, Column 1: Bear
sequentialNumbers: Contains 1, 2, any range including empty, 10.
oneTwoNumbers: Contains 1 then 2.
oneTwoTenNumbers: Contains 1, 2, any range including empty, 10.
oneTwoThreeTenNumbers: Contains 1, 2, any single number, 10.
primeNumbers: Starts with 2, then 9 more numbers.
fibonacciNumbers: Starts with 0, then any range of numbers.
emptyNumbers: Empty array
threeNumbers: Contains 9 then 7 then 5.
sixNumbers: Any items in any order.
PS C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter03\Arrays>
```

Go to the beginning of the file and add 2 lines we added everywhere throughout the lab (your name and .NET version)

The file contains several arrays and switch statement at the end. Skim through the code line by line and modify each of these arrays – add your own data to them.

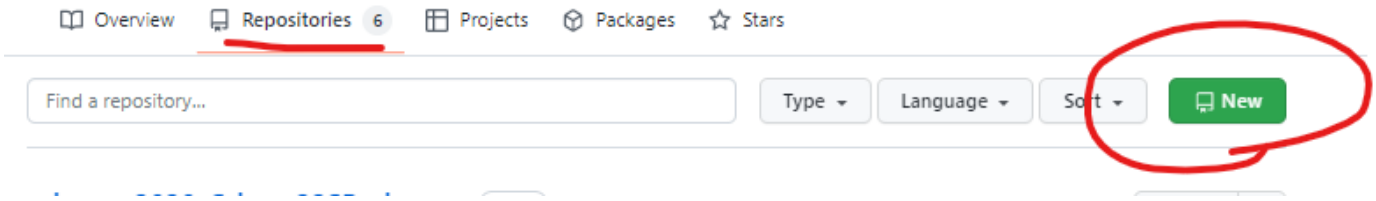
Once you complete copy-paste your Program.cs code go back to the app we created in VSC folder. Open VSC folder in VSC. Add your new code to the Program.cs there.



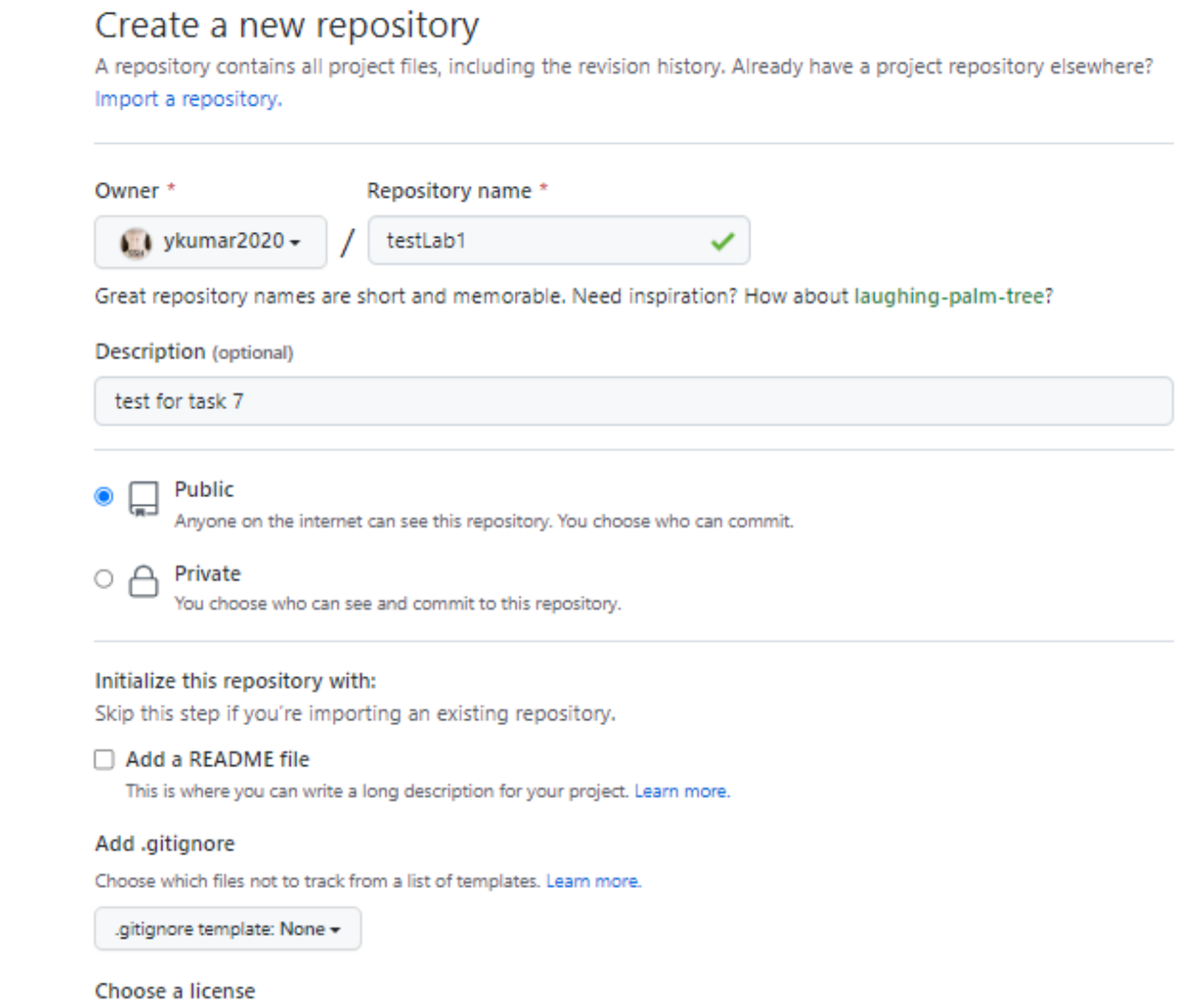
)

Add `using static System.Console;` on top of the file
Run the code to make sure it runs without errors.
Take screenshots of your source code and output.

When done – go to GitHub and create a new repository

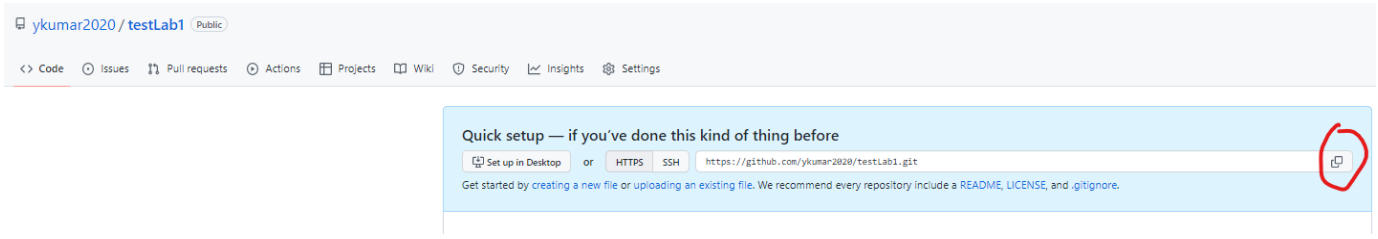


Give it a name



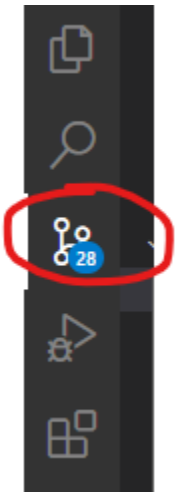
Create repository

Copy link to it to buffer

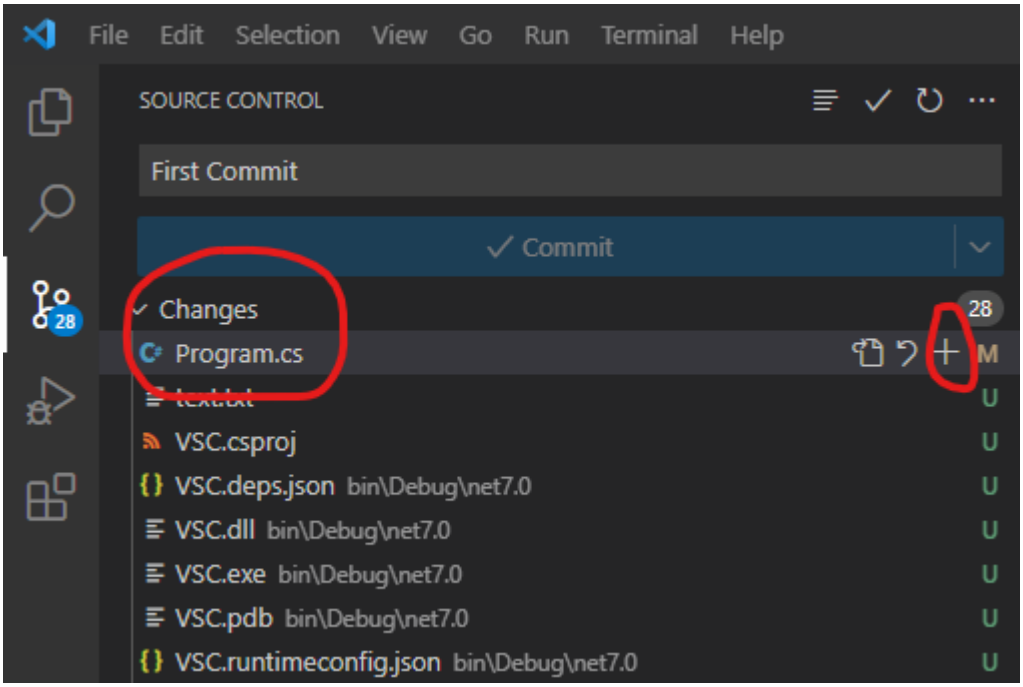


Add some spaces to Program.cs to make sure it has changes.

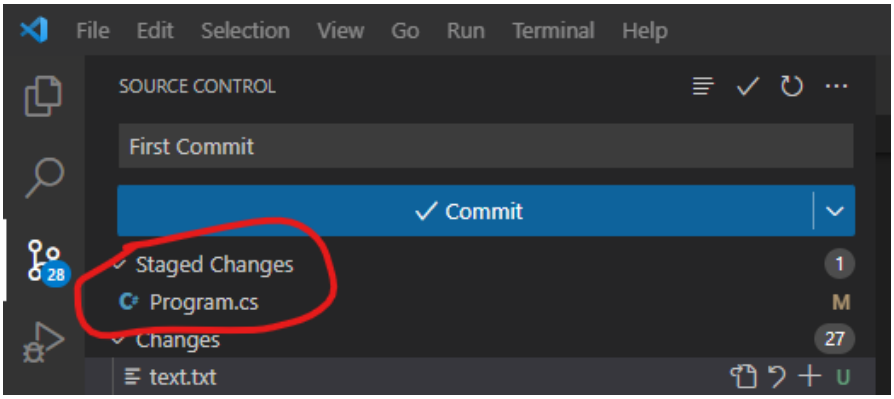
Save Program.cs, go to source control icon



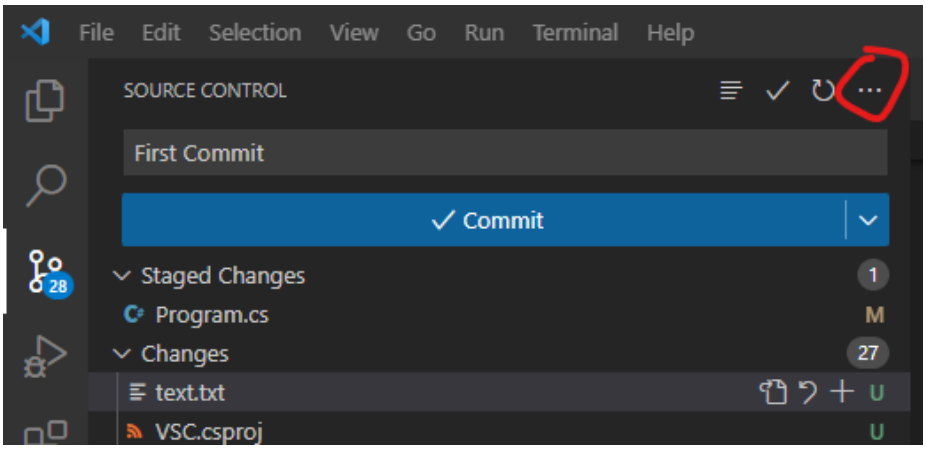
Click on + sign near Program.cs



Result



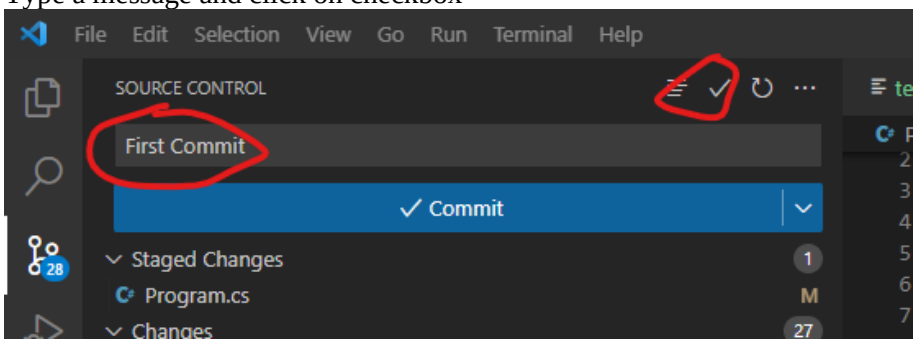
Click on … on the right



Choose Remote -> Add Remote, paste a link from your buffer

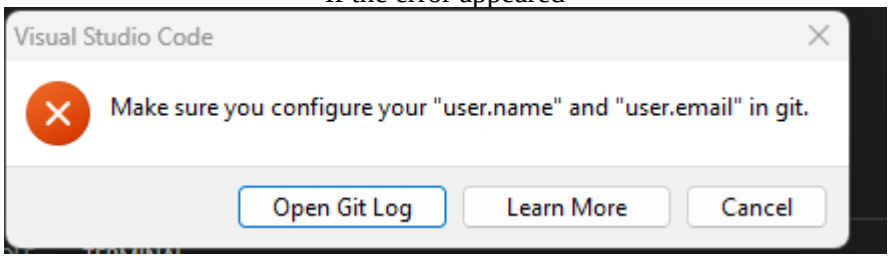
Give this connection a name (for example, **test** or **testing**) – type it in

Type a message and click on checkbox



Allow to sign in using the GitHub (your GitHub account should be open in a browser tab with you logged in)

If the error appeared



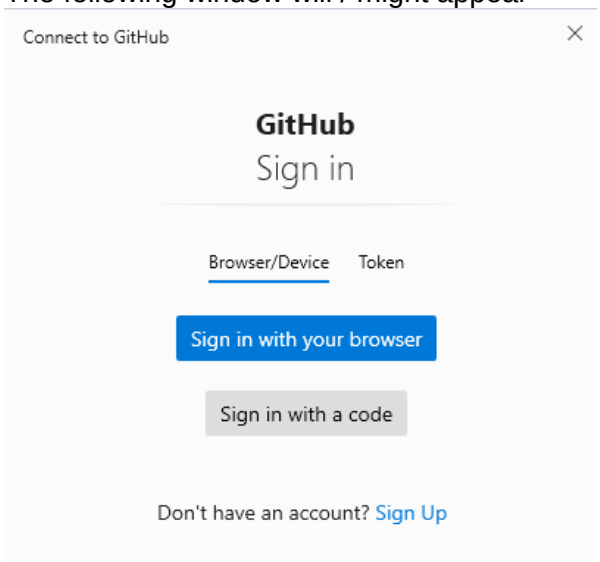
Type the following on terminal one line at a time:

```
git config --global user.name "John Doe"
git config --global user.email "johndoe@email.com"
```

Where username and email your GitHub credentials

```
PS C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter03\Arrays> git config --global user.name "ykumar2020"
PS C:\Users\galae\Repos\cs11dotnet7\vscode\Chapter03\Arrays> git config --global user.email "yuliakumar
```

Now give a commit message, press the “Ctrl + Enter” keyword, and then click on “Yes.” As you can see, it will start committing.
The following window will / might appear



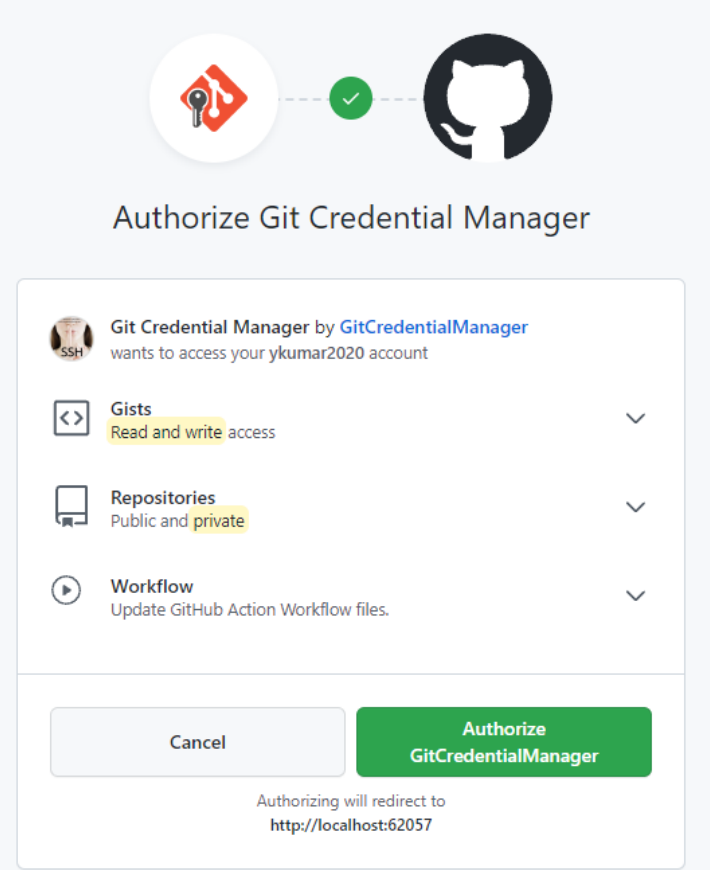
Workaround:

Click on the … -> Pull, Push -> Push to

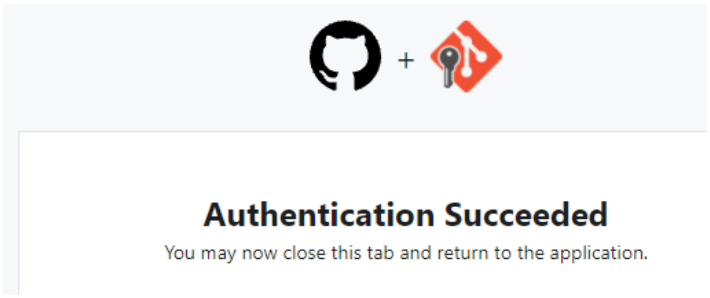
Click add remote

click on “Push To.” The URL will be shown in the selection area appended with the Remote name. Choose the URL that you have pasted and press “Enter.” After clicking on push, it’ll take some time to update the repository on Github.

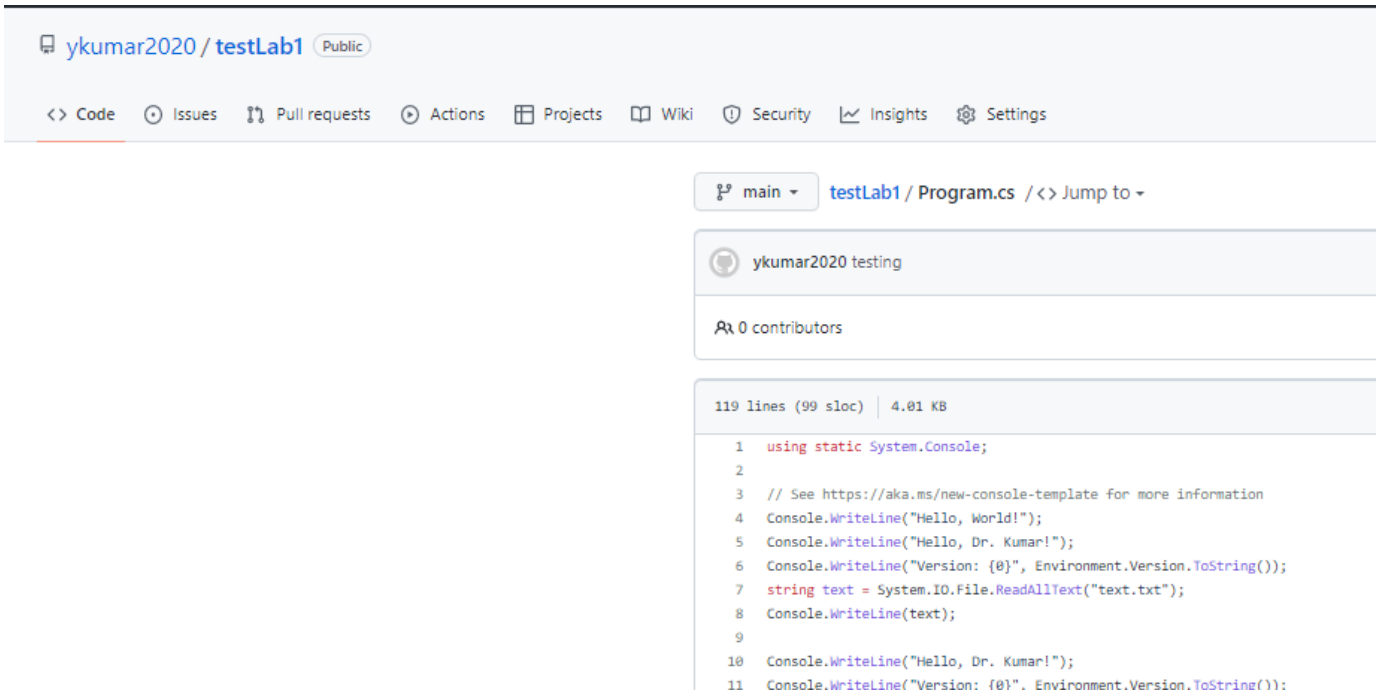
Sign in if prompted



then click on “Push to.” It will show you a message and click on “Add Remote” and paste the URL of that Github Repository and press “Enter.” Now, Enter the name of Remote and press “Enter.”



The code should appear on GitHub



PS. You might prefer to type commands on terminal or push file in any other manner. This is fine if this is **not a manual upload**

TASK 7. Provide screenshots proving that part 7 was completed, provide a link to your GitHub

Submit all 7 tasks in 1 file (PDF or Doc)