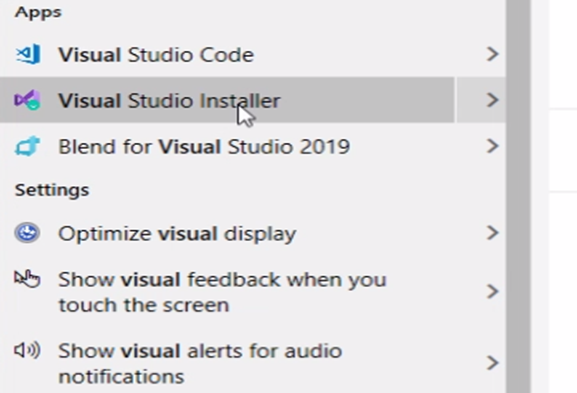
**Visual Studio integration with Git and Github**

**Getting started as first time user**

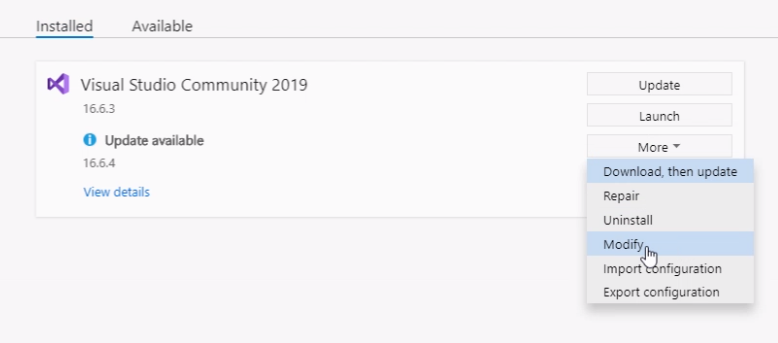
Follow tasks 1 to integrate your local Git repository to a remote Github account repository using Visual Studios.

**Task 1:** Before beginning your program via Visual Studio, it is best practice to verify that you have installed the necessary extensions, these extensions will enable developers to publish projects to a remote Github repositories with a local operating systems.

**Step 1**: Launch Visual Studio Installer to check for modifications

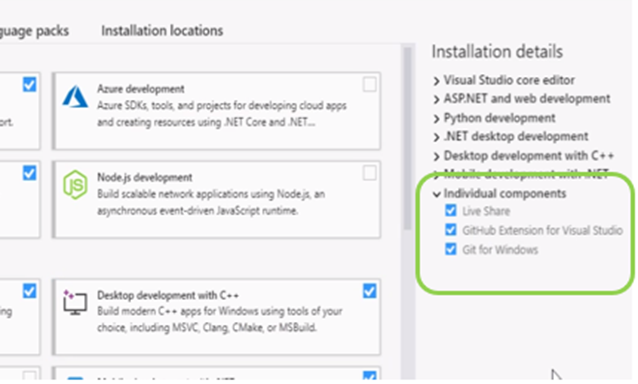


**Step 2:** The window shown below will appear once Visual Installer is launched. To check for your current extensions, **click on the “more” drop-down option and select “Modify”.**



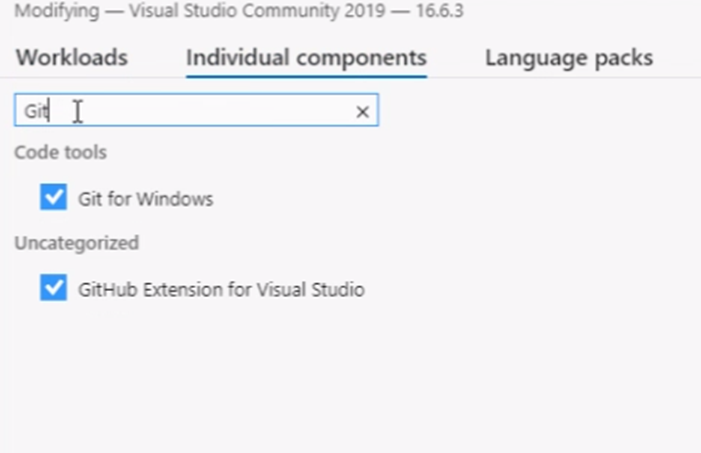
*Your next window will show a list of Workload options, a blue check mark will appear next to anything that has already been installed. The following extensions must be downloaded before beginning your program:*

* **Live share:** an extension that will enable real-time collaboration between developers
* **Github Extension for visual Studio:** add-on to Visual studio that allows you to create, publish, and clone existing repositories to the remote github repo
* **Git for windows :** shell extension that integrates in Windows Explorer and presents a context menu on files and directories.



*The image above, indicates that the necesseray extensions are available to begin a project, if this is the case, then simply close Visual Studio Installer and move on to Task 2.*

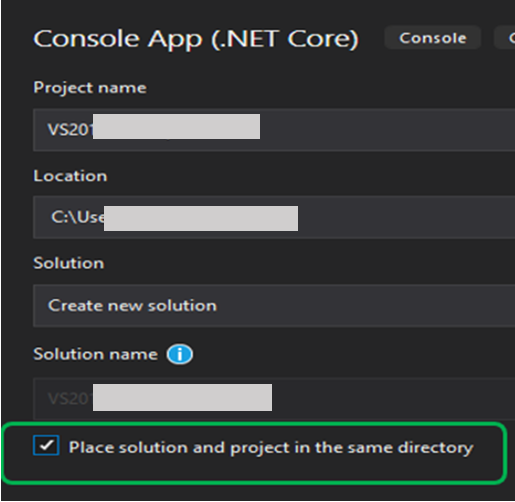
**Step 3:** If either Live Share, Github Extension for Visual Studio , or Git for Windows is not installed, **choose “ Individual component” tab and search for the missing extension, then install.**



**Signing on to your Github with new project via Visual Studio**

**Task 2:** This taskdemonstrateshow to sign on to your Github account with Visual Studio when creating a new project**, if you are working on an existing Github project, please proceed to Task 3 (Page 11)**. The following set of instructions are a necessary step that allows users to work out of local branches and make commits to remote repositories via Visual Studio.

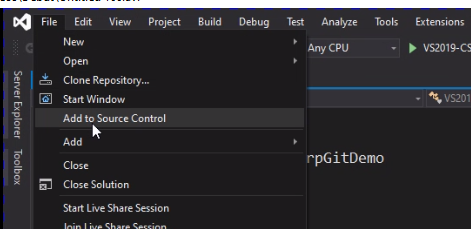
**Step 1:** Begin by opening Visual Studios and creating a new Console App (.Net Core), naming it, and selecting “Place solution and project in the same directory” before clicking create.



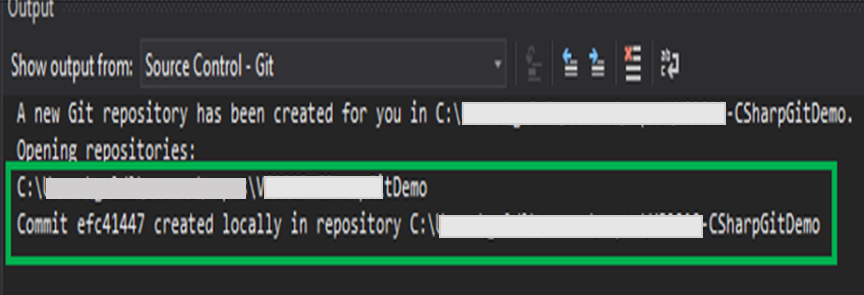
*Name project and click create*

**Step 2: File > Add to Source Control**

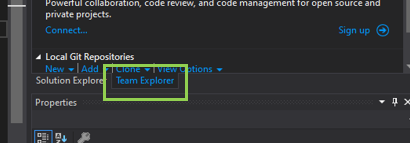
Select ‘Add to source Control’ to add your new project to your local git. This step allows developers to make changes via visual studio and sync those changes to a remote repository shared with other project collaborators



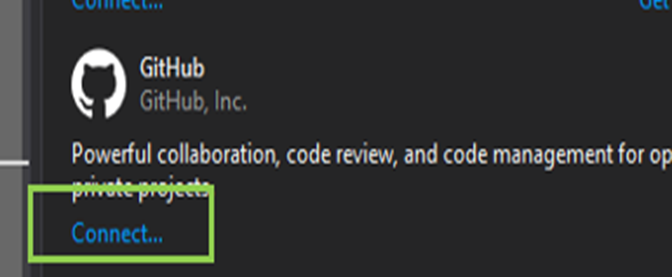
*Clicking on “Add source Control” will create a local repository that makes a commit and creates a folder that is the same name as your project, your output screen should look like the image below.*



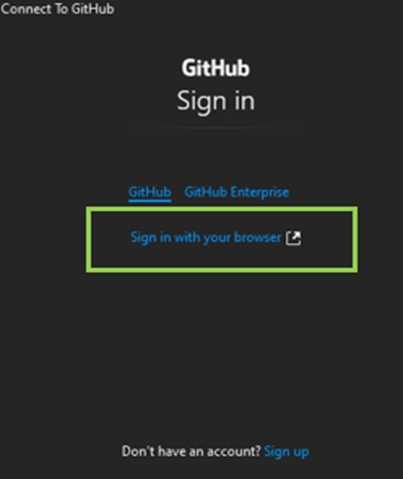
**Step 3:** Go to Team explorer to display Github connection options



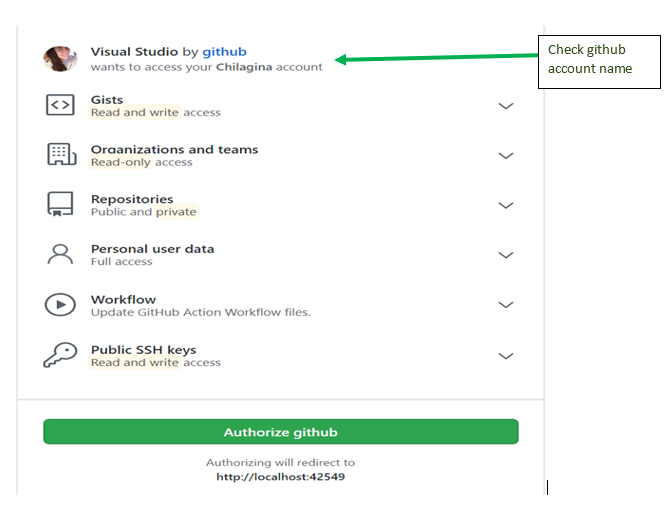
**Step 4:** Click the ‘Connect’ button under Github to begin connection process

****

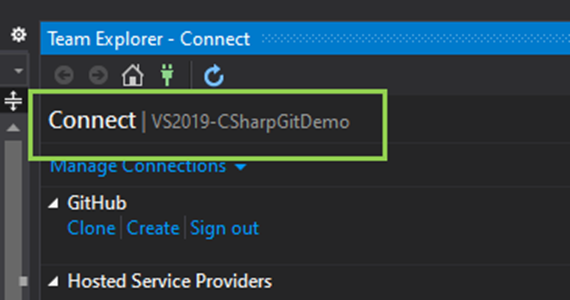
**Step 5:** Sign into your GitHub Account: choose to sign in with your browser. If you do not have an account, you must choose the ‘Sign up’ option at the bottom of the screen before continuing.

****

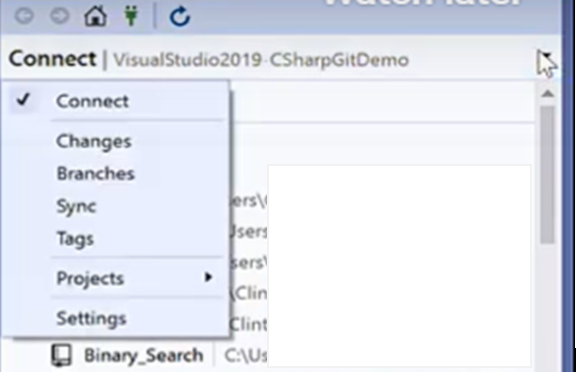
**Step 6**: Make sure that Visual Studio wants to access the correct Github account, then Choose the ‘Authorize github’ button to continue



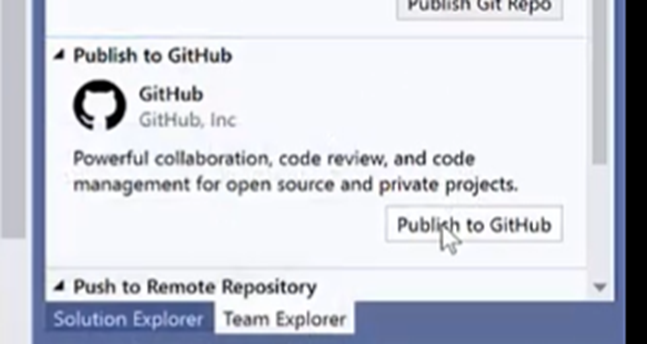
*To view if you are connected locally, check under Team Explorer in your visual studio project. Your project should display the name of the project you created next to ‘Connect’*



**Step 6:** Sync the local Repository you are currently connected to by selecting the drop-down option like the image shown below

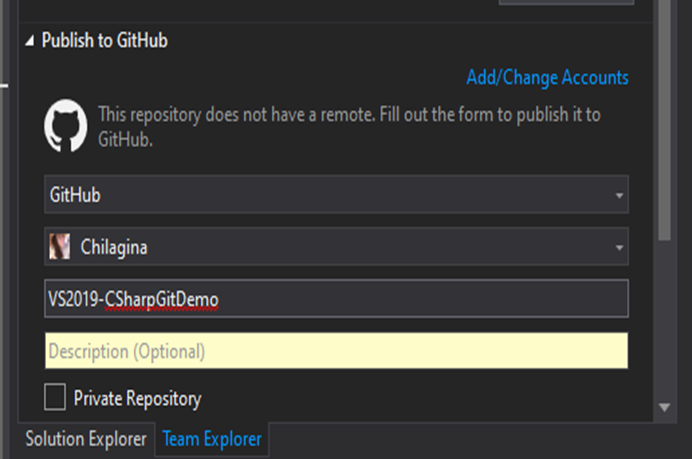


**Step 7:** Click ‘Publish to Github’ to begin publishing process

****

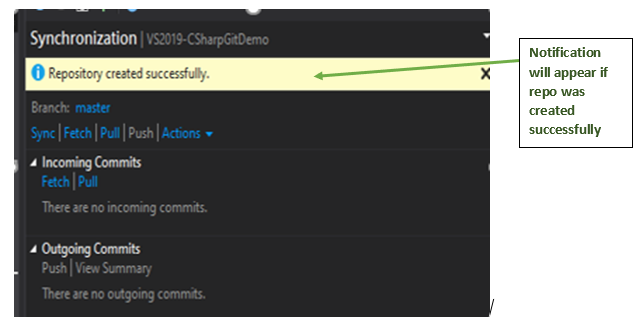
*After clicking ‘Publish to Github’ a window like the image below will appear.*

*Your project name and personal Github account name should appear*

****

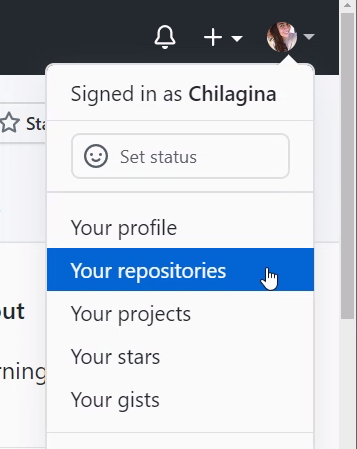
*If the information is correct, select publish.*

*After publishing the notification below will appear on Visual Studio’s Team Explorer.*

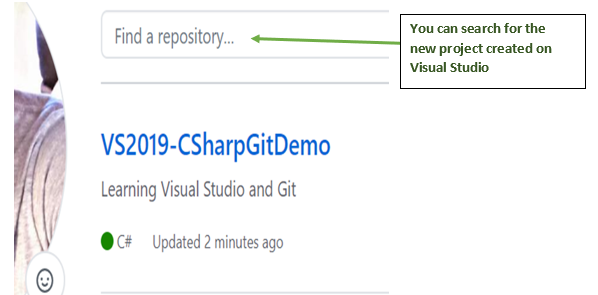


**Step 8:** Check if project was successfully created in Github

To check whether the steps above have worked, go to your Github account on your browser. Select ‘Your repositories’ by selecting the drop-down option next to your profile picture. Your new project should appear in your list.



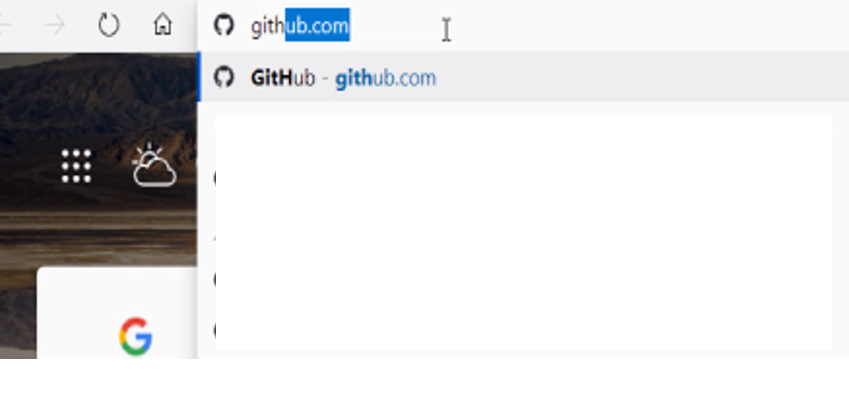
*You can search under ‘Find a repository’ if the name of your project doesn’t appear.*



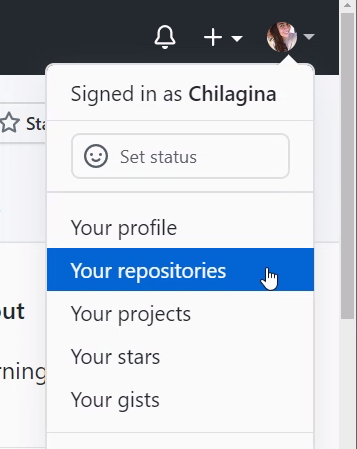
**Connecting to Github via Visual Studio with an existing project**

**Task 3:** The Following task demonstrates how to connect to Github with Visual Studio if developer wishes to work with an existing project.

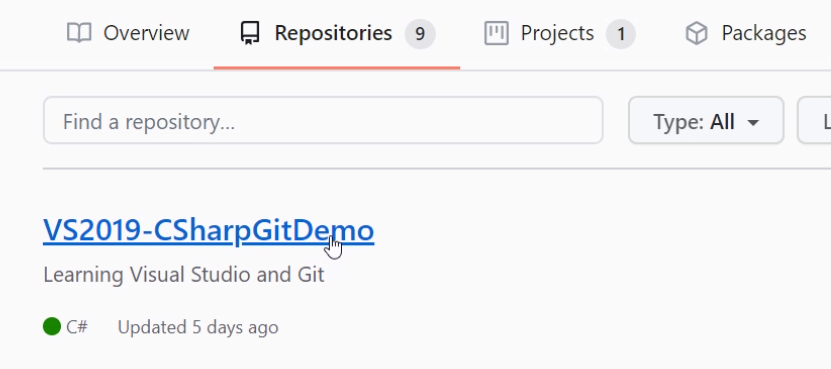
**Step 1**: Sign into you GitHub account through your browser



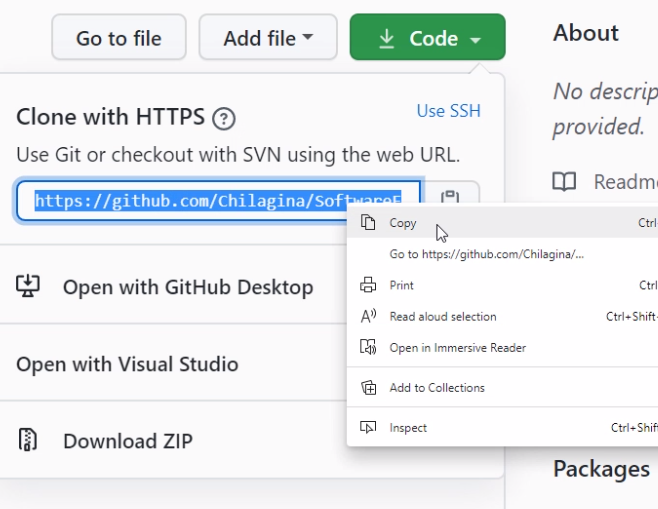
**Step 2:** Go to the drop-down option like the image below and choose ‘Your repositories’



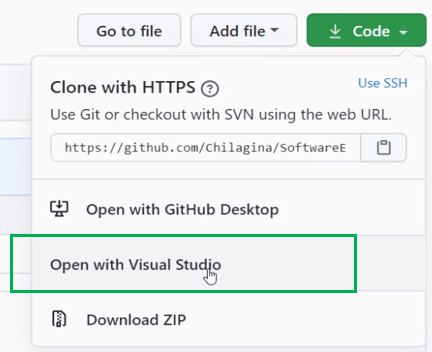
**Step 3:** Click on the project you wish to collaborate on under the repository list



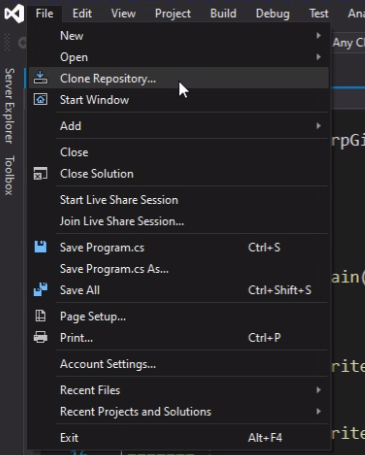
**Step 4**. Click on the ‘Code’ button to view drop-down options and copy the projects repository URL before proceeding to the next step



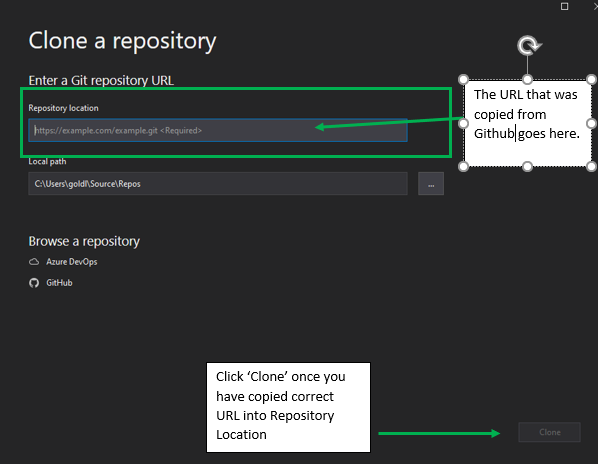
**Step 5:** Select the option to ‘Open with Visual Studio’



**Step 6:** Once Visual Studio is open go to **File > Clone Repository**



**Step 7:** Paste the Git repository URL that you copied in Step 4, then click the ‘Clone’ button

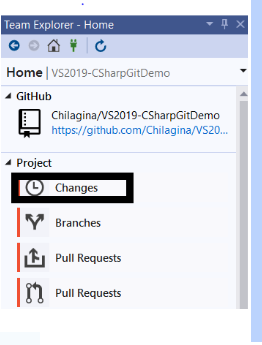


**Committing to changes from Local Visual Studio to Remote Github Repository**

**Task 4**: This task will allow developers to save files on Github with a unique ID. This will help keep track of changes made by other developers, if any**.**

**Step 1:** If you make changes to your program in your visual studio locally and wish to push these changes to a remote Github for other collaborators to see, Open

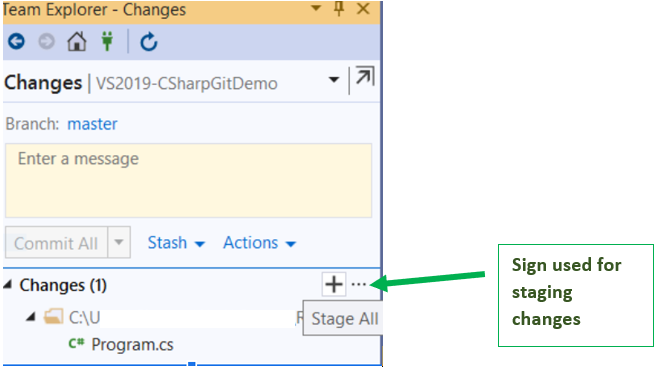
**Team Explorer > Home > Changes**

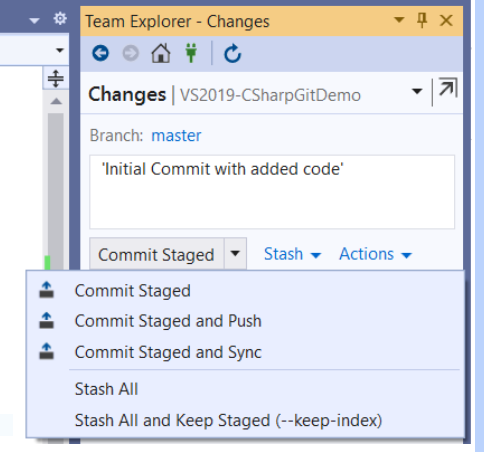


**Step 2**: Click on the ‘+’ sign under changes drop down which is used for staging the changes.

‘*Staging is used to queue all the changes as a group so that they can be committed together at once’*

**Step 3:** Type your commit message under the text box ‘Enter a message’. From here you have three options.





* **Commit Staged** - simply makes a record of your changes that you have made on your local machine. It will not mark the change in the remote repository.
* **Commit and Push** will do the above and push it to the remote repository. This means that any changes you have made will be saved to the remote repository as well.
* **Commit and Sync** does three things. First, it will commit. Second, it will perform a pull (grabs the updated information from the remote repo). Finally, it will push.

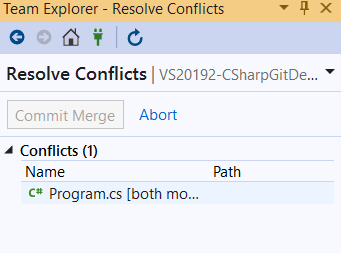
*‘Always perform Commit and Sync if you want to have the update copy locally.’*

**STEP 4:** Correcting a merge conflict

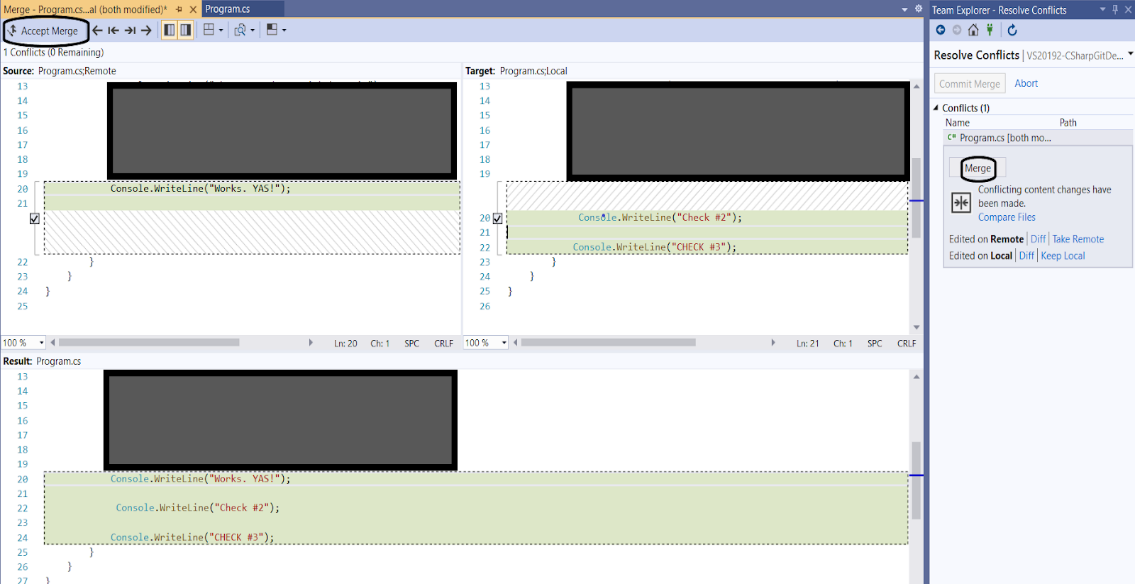
1. Click on ‘Conflicts’ link with attention symbol.



1. Click on the File name [ Here Program.cs]

****

1. Click on Merge on the right side of the visual studio window in Team Explorer. Select the instructions in both files which you want to merge and see the result in the window beneath. If accepts, click on ‘Accept Merge’



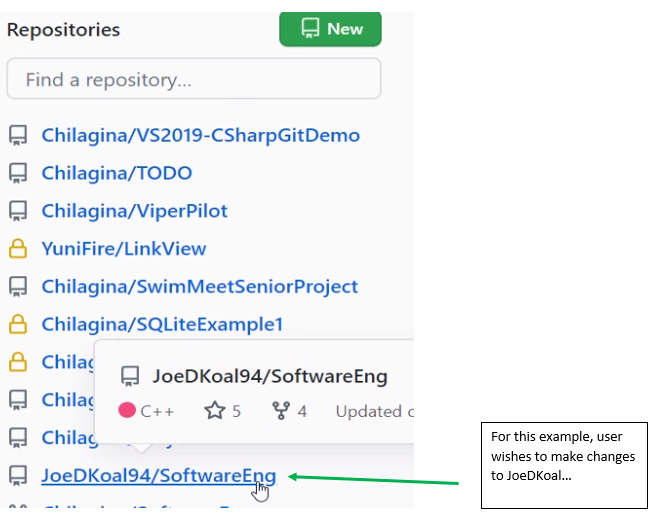
The code is pushed on Github remotely and you have the synced changes remotely.

**How to perform Pull Request via Github**

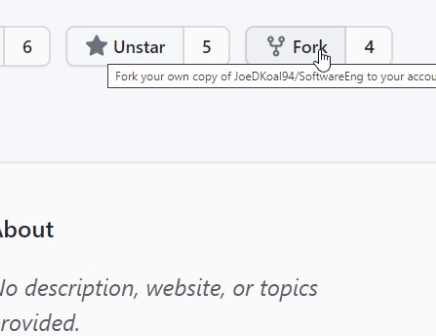
**Task 5:** A pull request is a way to commit to a repository in which you do not have writing permissions. The point of a pull request is to allow other project collaborators to view the request you have made before merging or pulling it with your code or leaving the original as it is.

So, a commit is a discrete change to one of more files that is a critical part of Git, while a pull request is a request to merge one or more commits into a different branch.

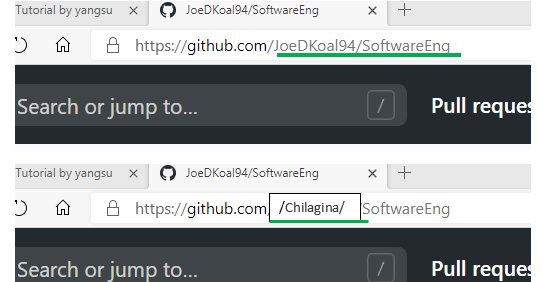
**Step 1:** User will open the repository that they wish to work with via GitHub by choosing ‘My Repository’



**Step 2:** Click on ‘Fork’ button to copy the code you want to work with into your own private repo

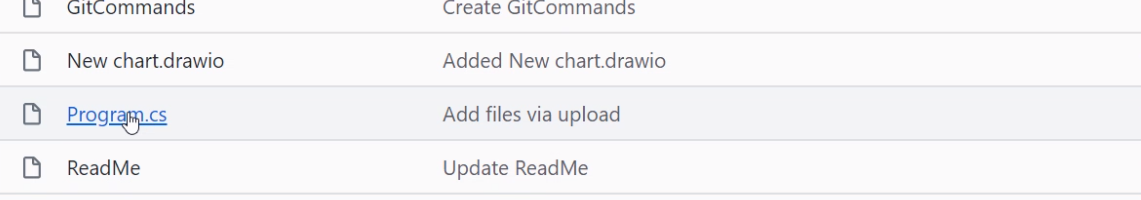


*Once you have clicked ‘Fork’, the name at the end of the URL should change from the original to your name because you have created your own private repo.*

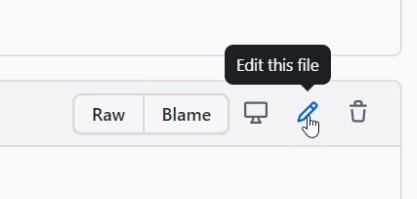


*If you want to make big changes, you can take the link and work on it with it on your local computer, you can make big changes there and once things look ok, you can push it back to Github to your own private repository, or you can make the changes through Github, like the example shown below.*

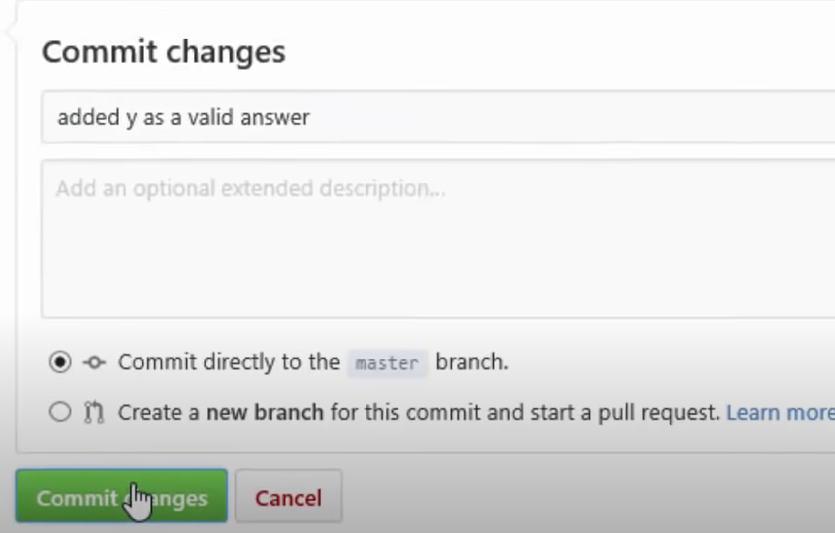
**Step 3:** Click on what you wish to work on in the private repo you have just created.



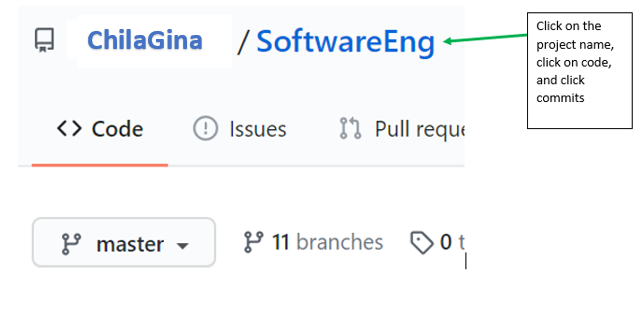
**Step 4:** The code will appear, click on the pencil icon to ‘Edit this file’. It is located on the upper right side of the screen. This step will allow you to make changes to the code via Github. Since this is a private repo, changes will not be made to the original file until they have been approved by the creator.



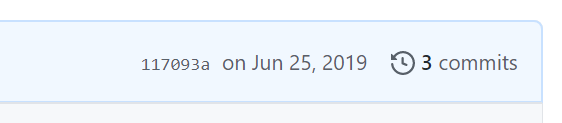
**Step 5**: Once changes have been made to the code, write a short message describing what changes you made to the project and click on ‘Commit’

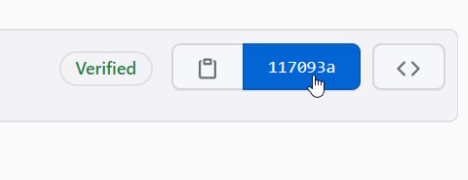


**Step 6:** Check if changes have been made by clicking on project name.



**Step 7:** Click on the ‘commits’ tab to see the changes that have been made to the code





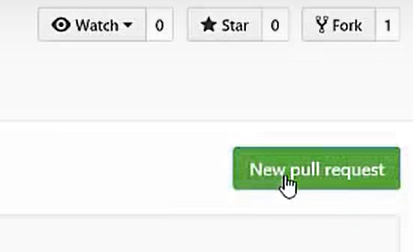
*The changes you have made will be highlighted*



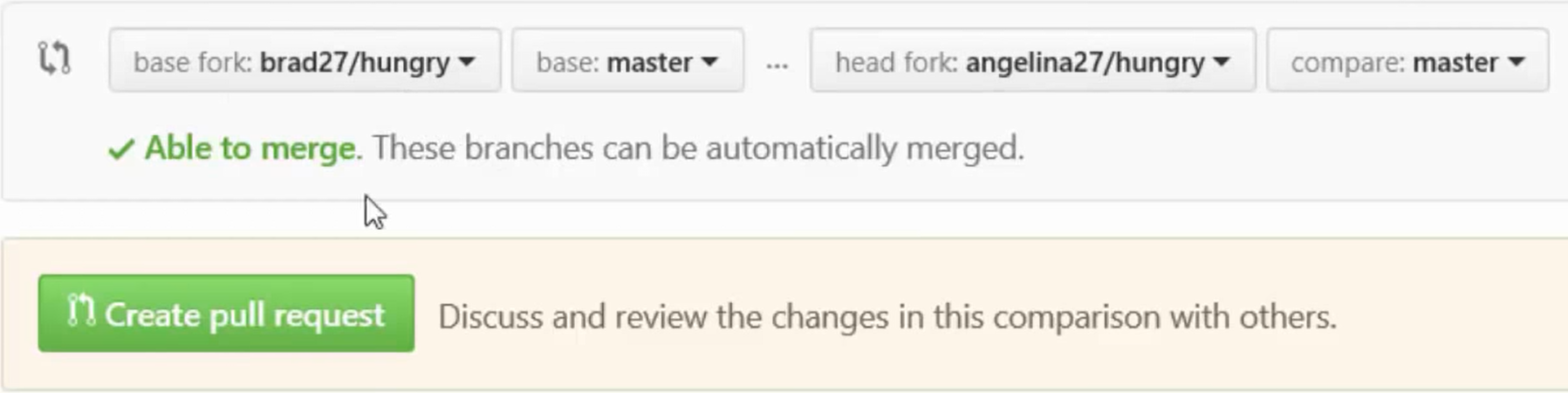
**Step 8:** Now that the changes are ready, you can send a pull request. The pull request is sent so that her changes are merged to the original repository. First click on ‘ Pull request’ tab , located near the top of your screen.



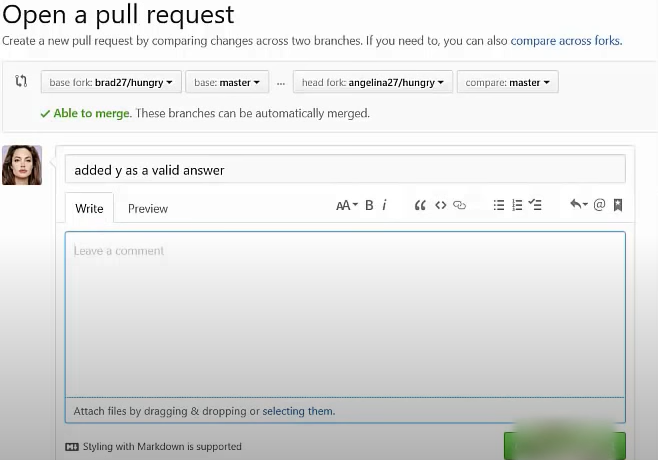
**Step 9:** Click on ‘New pull Request’ button on the upper right side of the screen.



*Once pull request has been clicked, a window like the one below will appear. In the image shown below, ‘angelina27’ will like to merge her repo with ‘brad27’.*



**Step 10:** Click on the ‘Create pull request’ button. The following will appear after clicking on ‘Create pull request’ button, there you can add comments. Adding comments can be very useful so that other collaborators can understand the changes you wish to make to the code. Click create pull request button at the bottom of the screen to proceed.



**Step 11:** Wait for other collaborators to view your changes and verify your contributions to the project.

If you wish to learn how to accept the pull request explained in this task, view explanation here <https://www.youtube.com/watch?v=e3bjQX9jIBk> starting at minute 6:15.