IMAT3904 Lab 1:

C++ refresher, and source code management

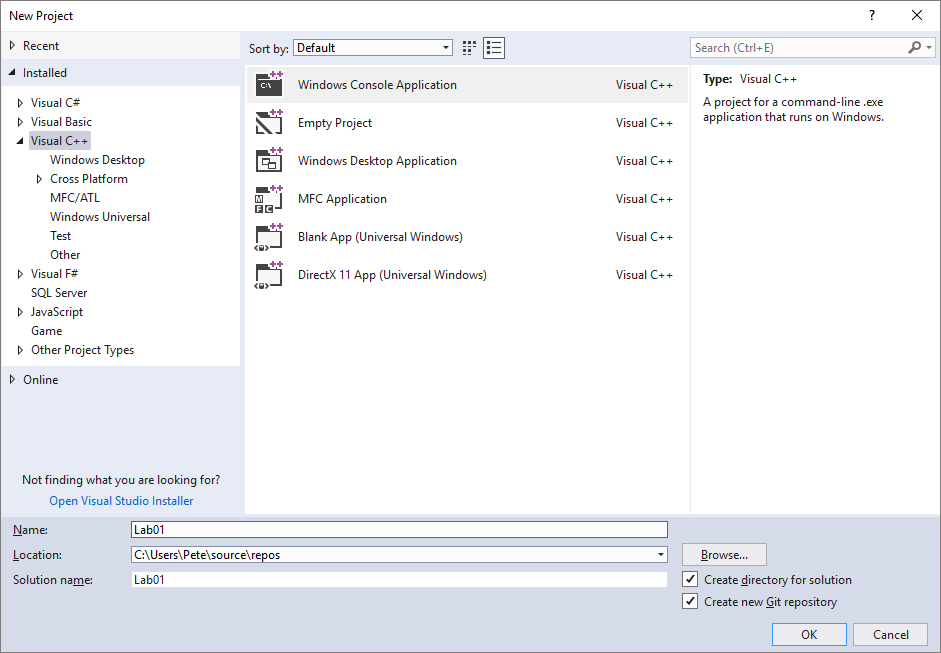
This lab session focuses on using source code management systems to store, collaborate and manage changes/versions for your source code in your projects. There are many online tools and repositories which offer these for free or for a monthly fee, but in these lab sessions a private github repo has been acquired by the university to use on this module. We do not want to make public repos on this module as everyone will be able to see your work and may open you up to plagiarism allegations.

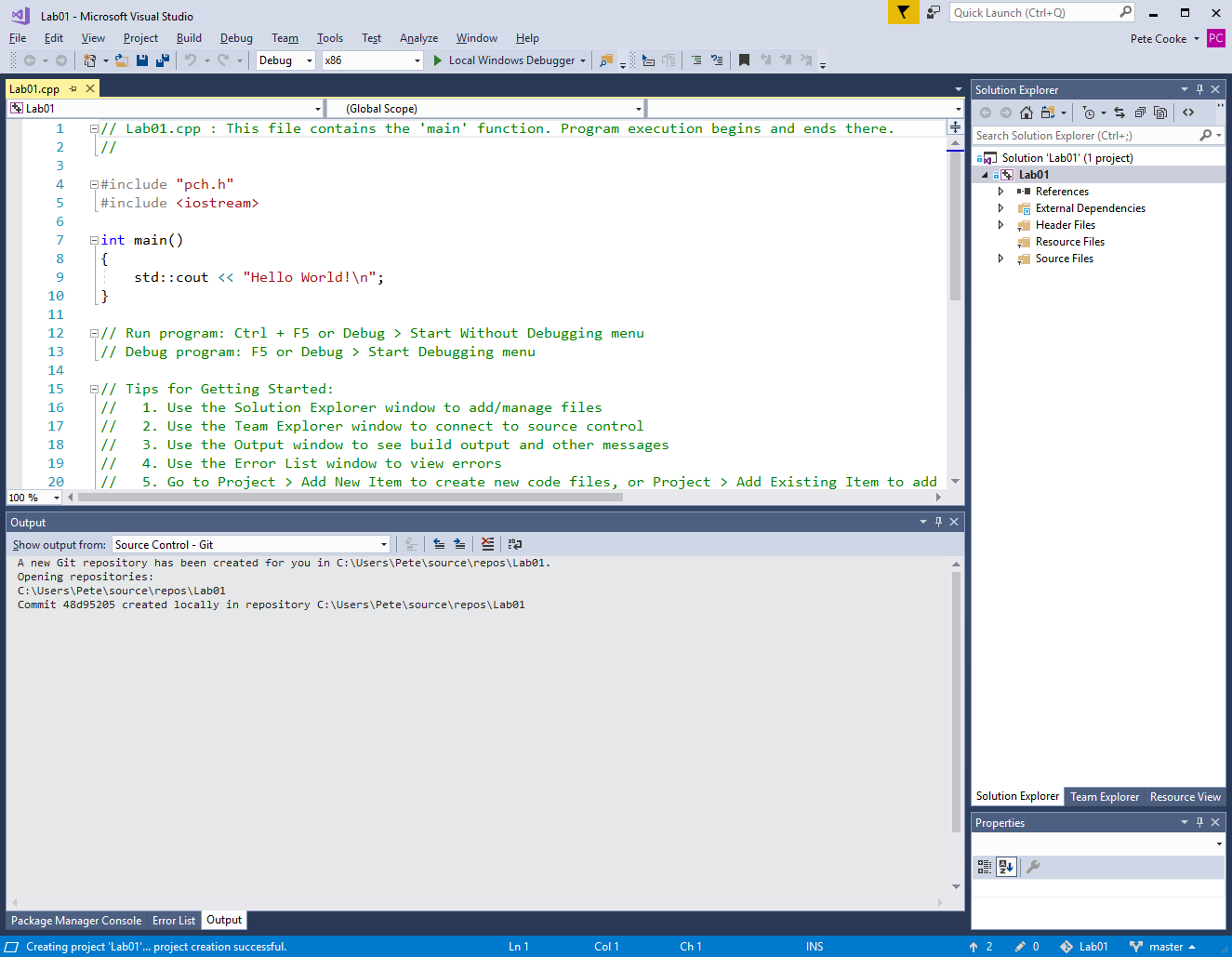
Sign up to the Github repo using the link https://classroom.github.com/a/S4iverf9

Make sure you choose yourself a memorable username and password, as you will be using this account for the entire module. Authorise the github classroom and accept the first assignment.

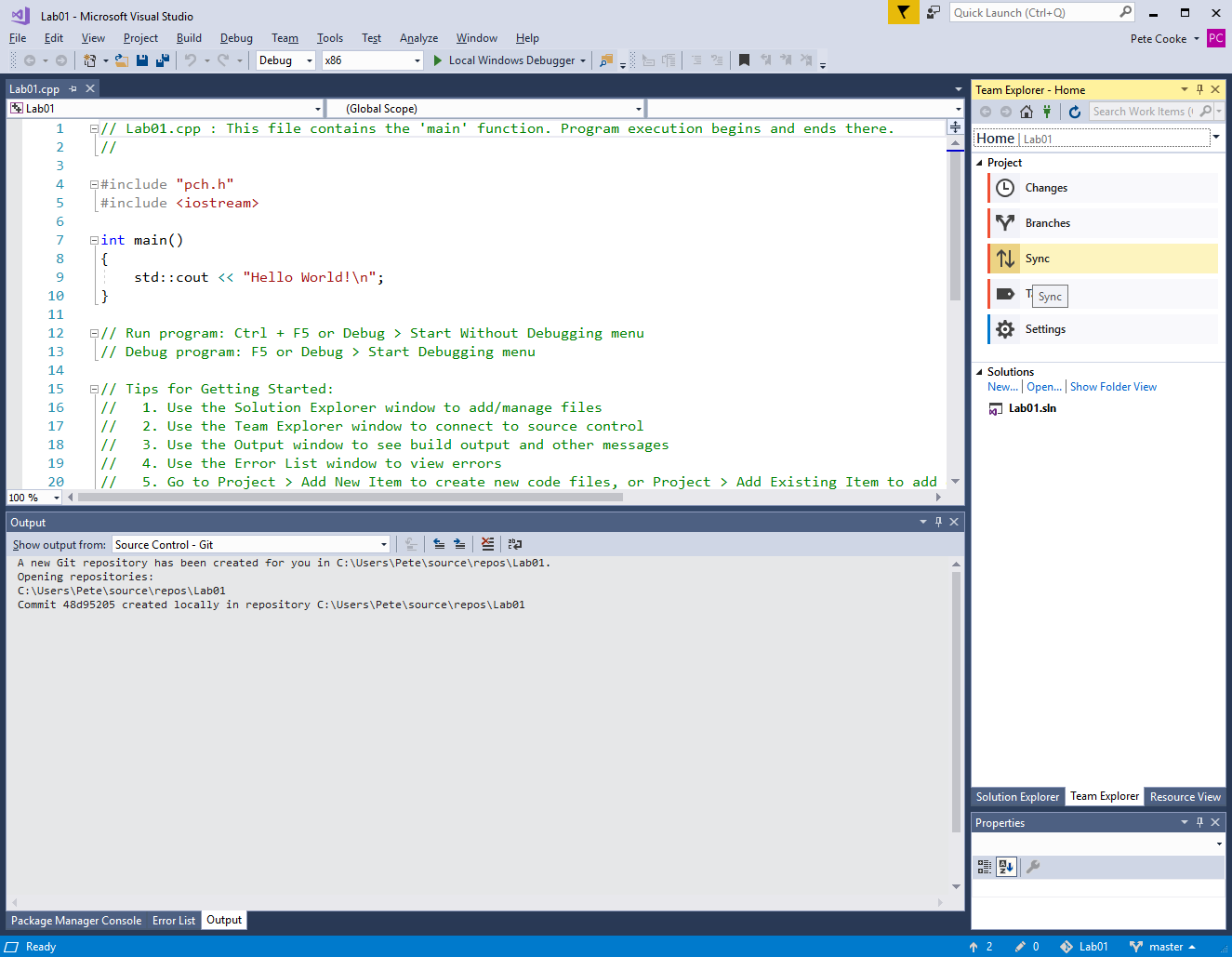
Create a suitable empty folder to store your repo on your H: drive e.g. IMAT3904/Lab1

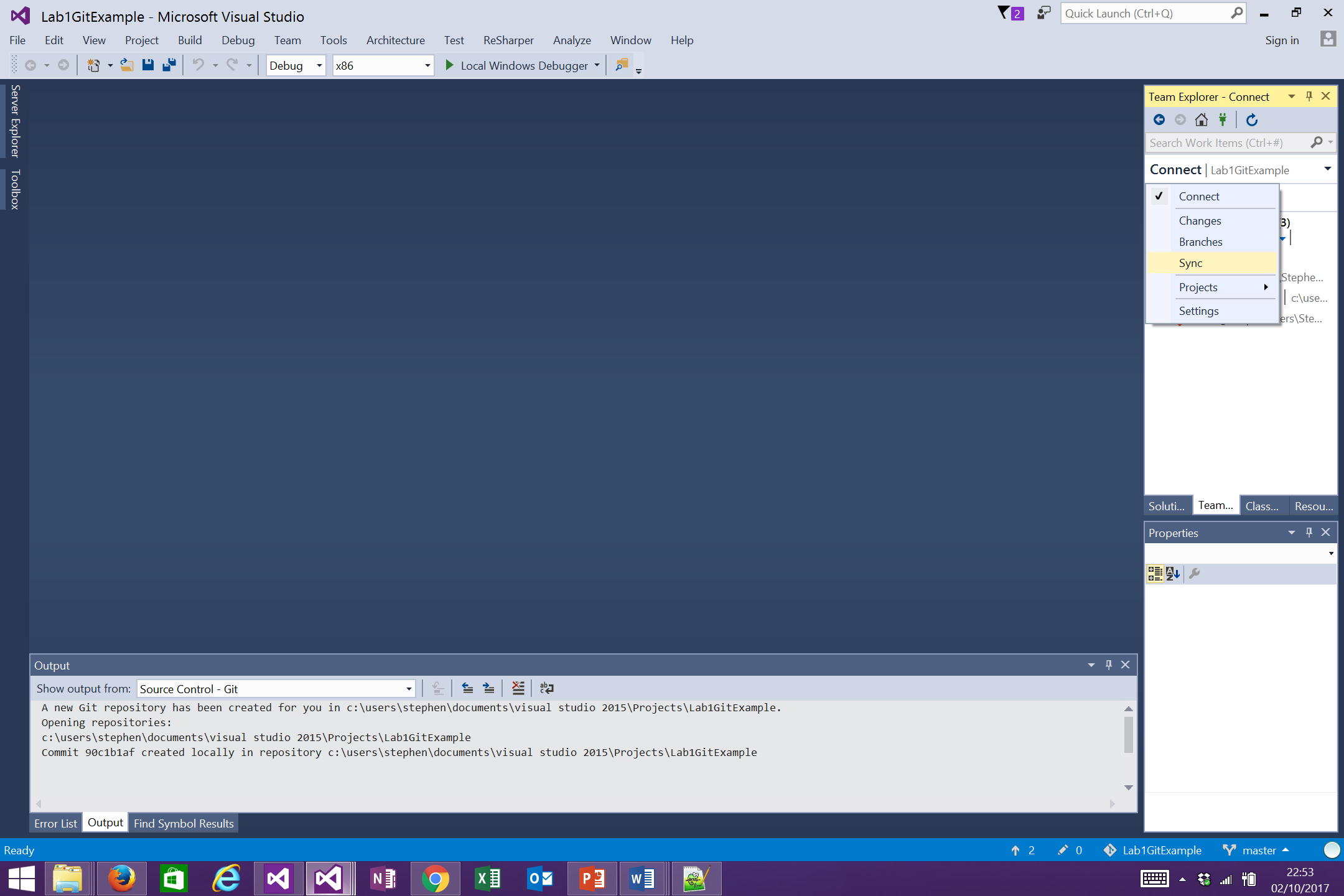
Create a new Visual Studio 2017 project in your new folder. The project should be a C++ console application. Tick the ‘Create new Git repository’ checkbox and click OK



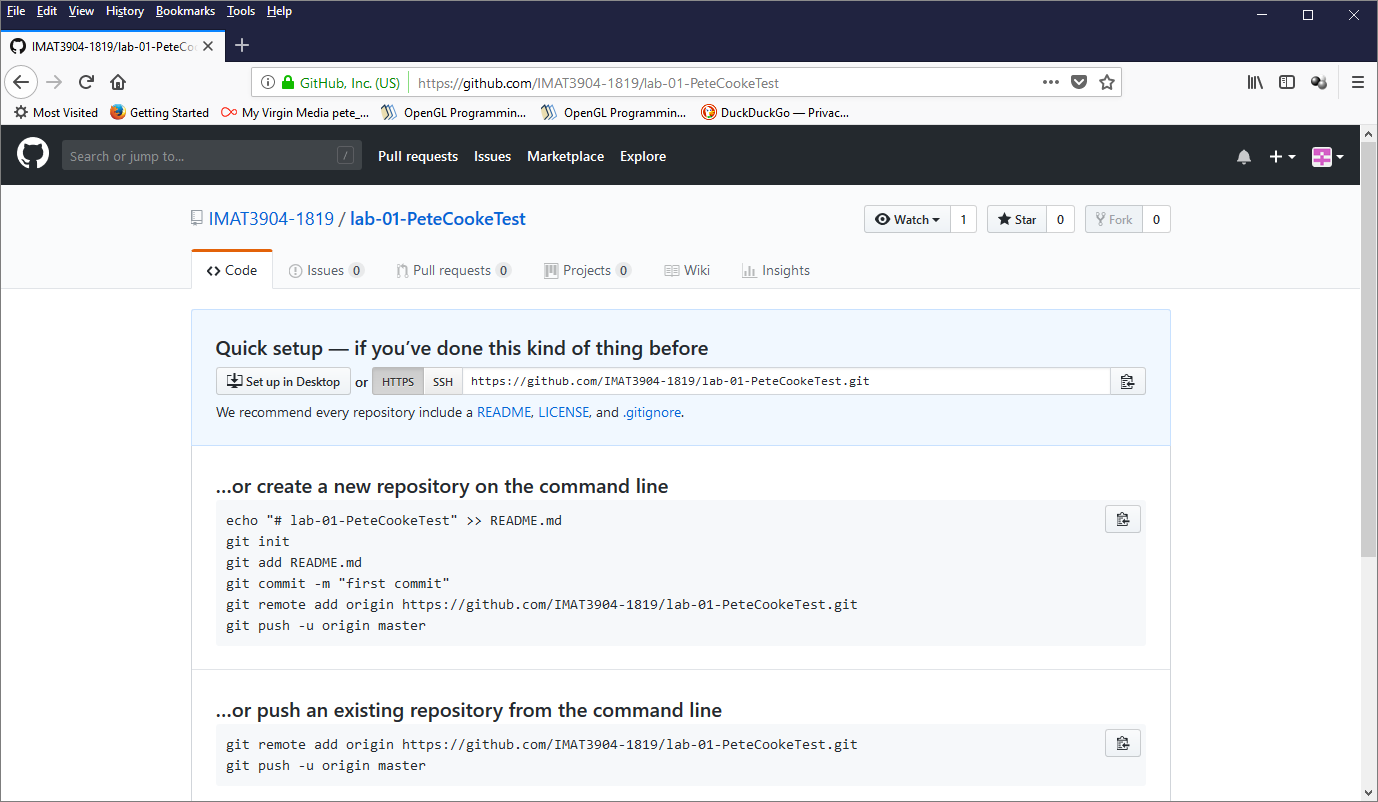


Connect your local repository with the remote github repo. Click on the Team Explorer tab (right hand side on screenshot below). Click the drop-down box that says Home and click Sync.

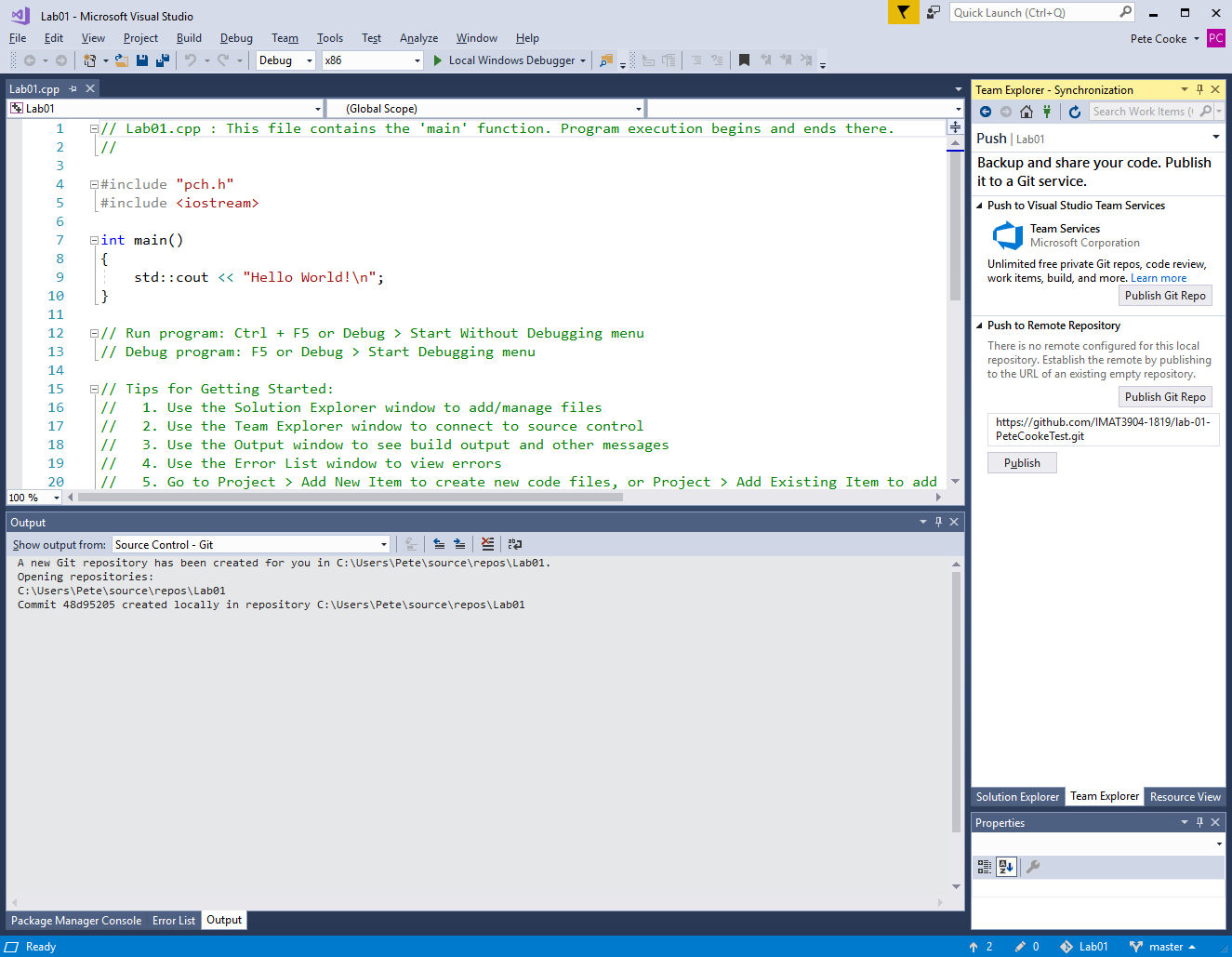




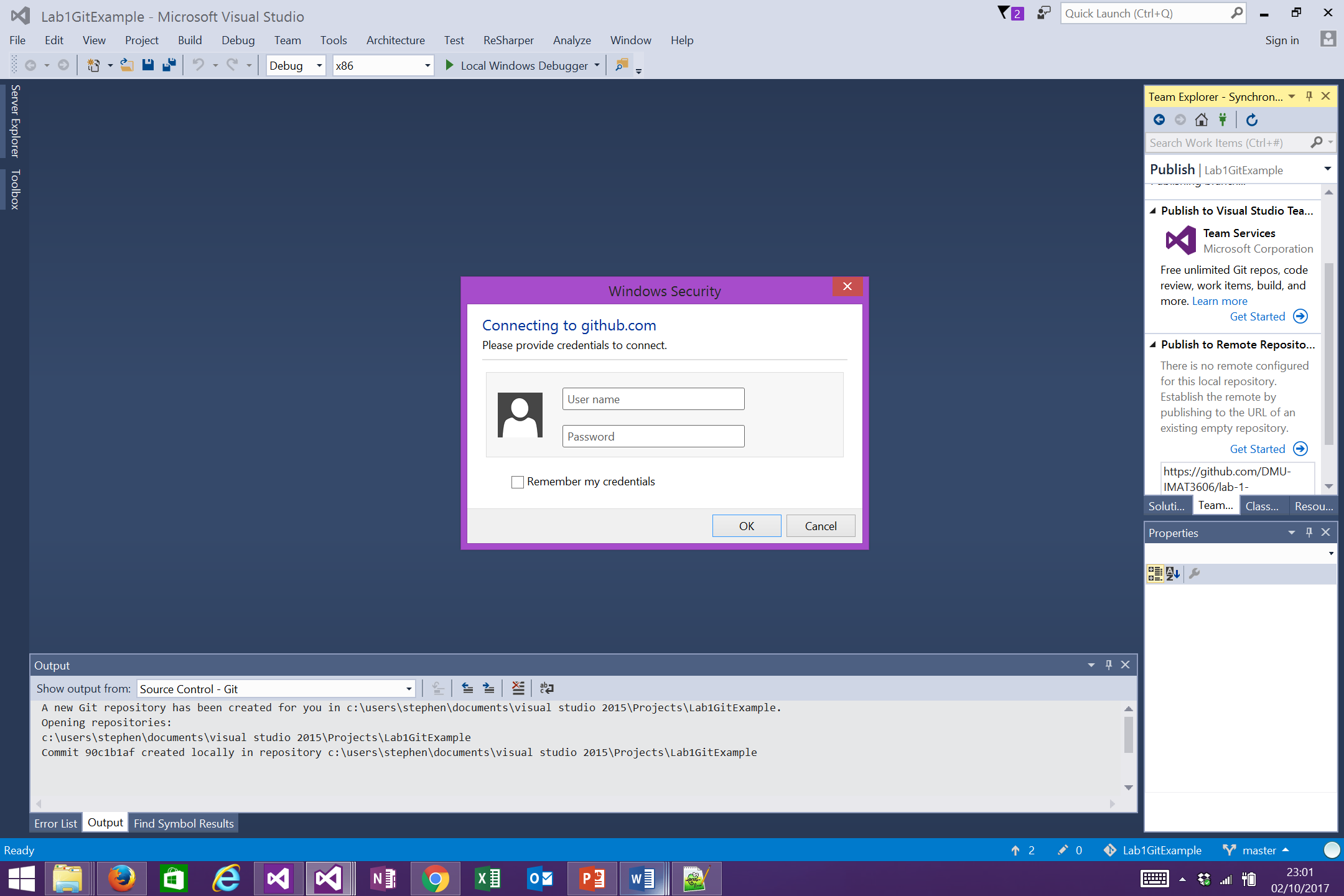
Your repository will have a web address which we can now link with visual studio. Get a copy of the link (ending in .git)



Paste it in the ‘Publish to Remote Repository’ section and click Publish



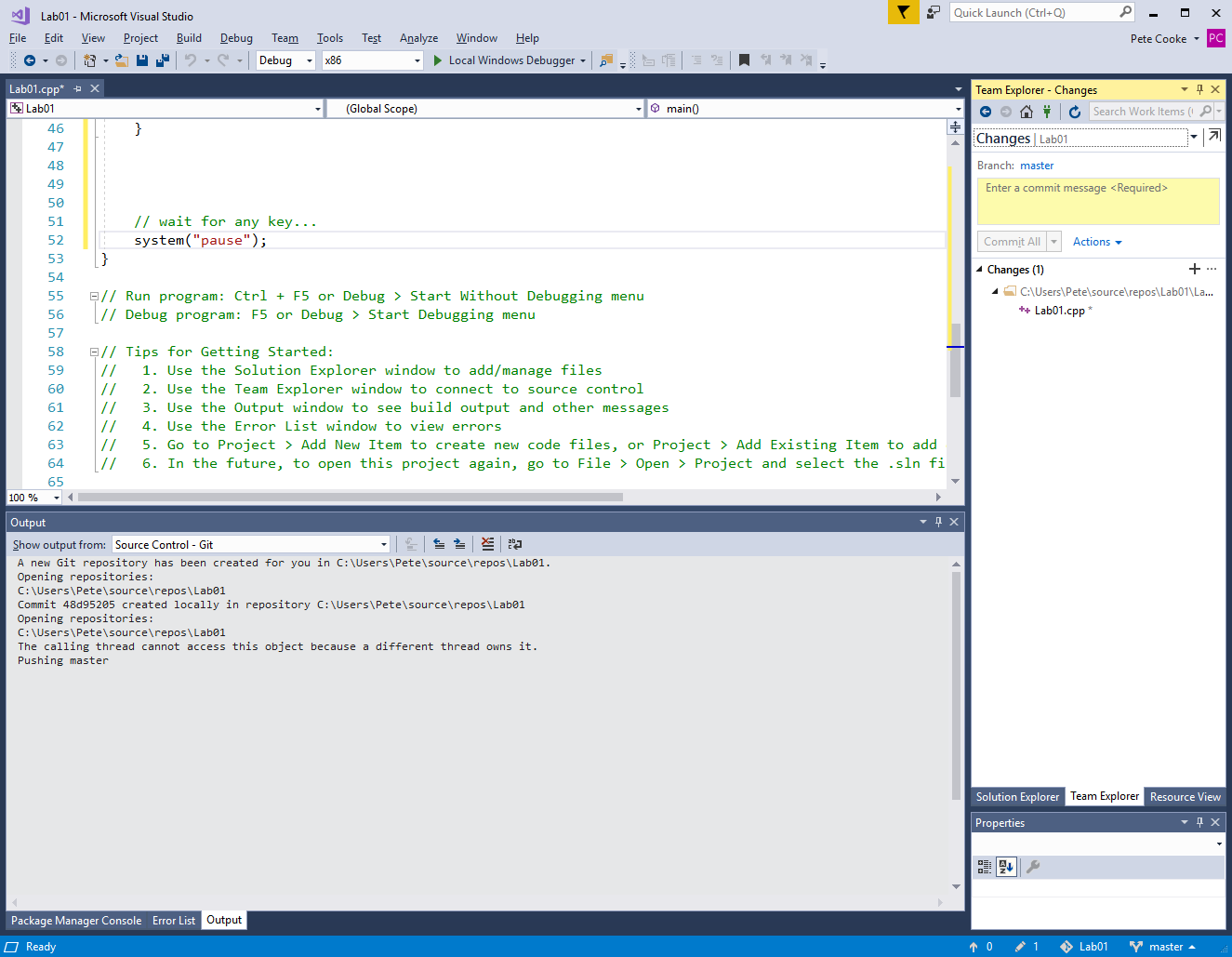
It will ask for your details that you signed up with to github.



Click the Solution explorer tab and add a new source file. Refresh your basic abilities with C++ by creating a simple console-based game.

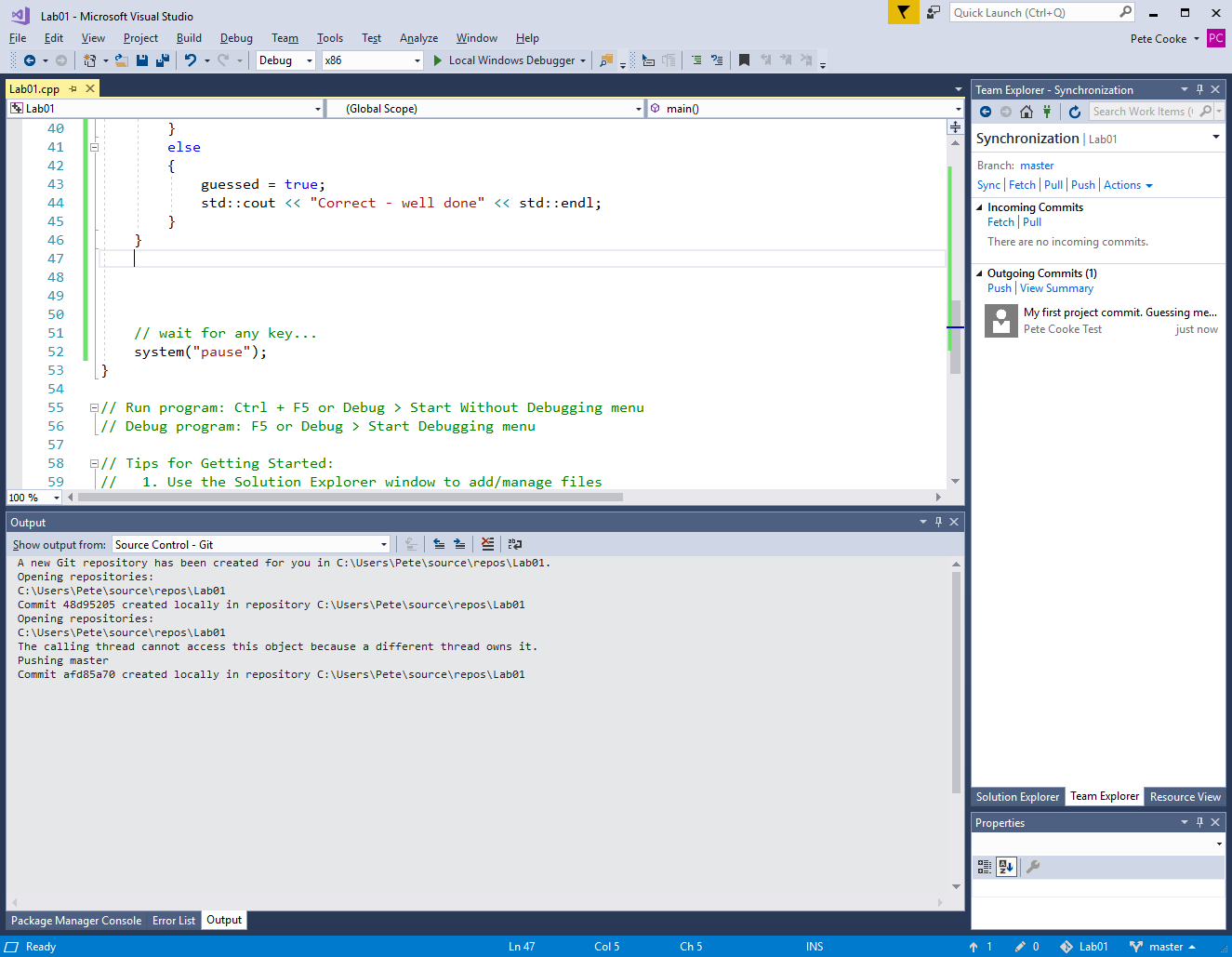
* Let the computer select a random number between 1 and 100 with the player having to guess the number in 5 guesses.
* The player’s questions are specific guesses in the form of numbers between 1 and 100 but answers are in the form of higher/lower or yes if the guess is correct.
* It is good practice to plan out the flow of your code via pseudo code and not worry about the details initially.
* Use the rand() function for the number choice. Check if it is actually random every time you compile the program (what is the role of the srand() function?)

Once you have completed (or partially completed) the task. Open up the team explorer tab again and click ‘Changes’ from the drop-down box. All the files that have been changed will be shown.

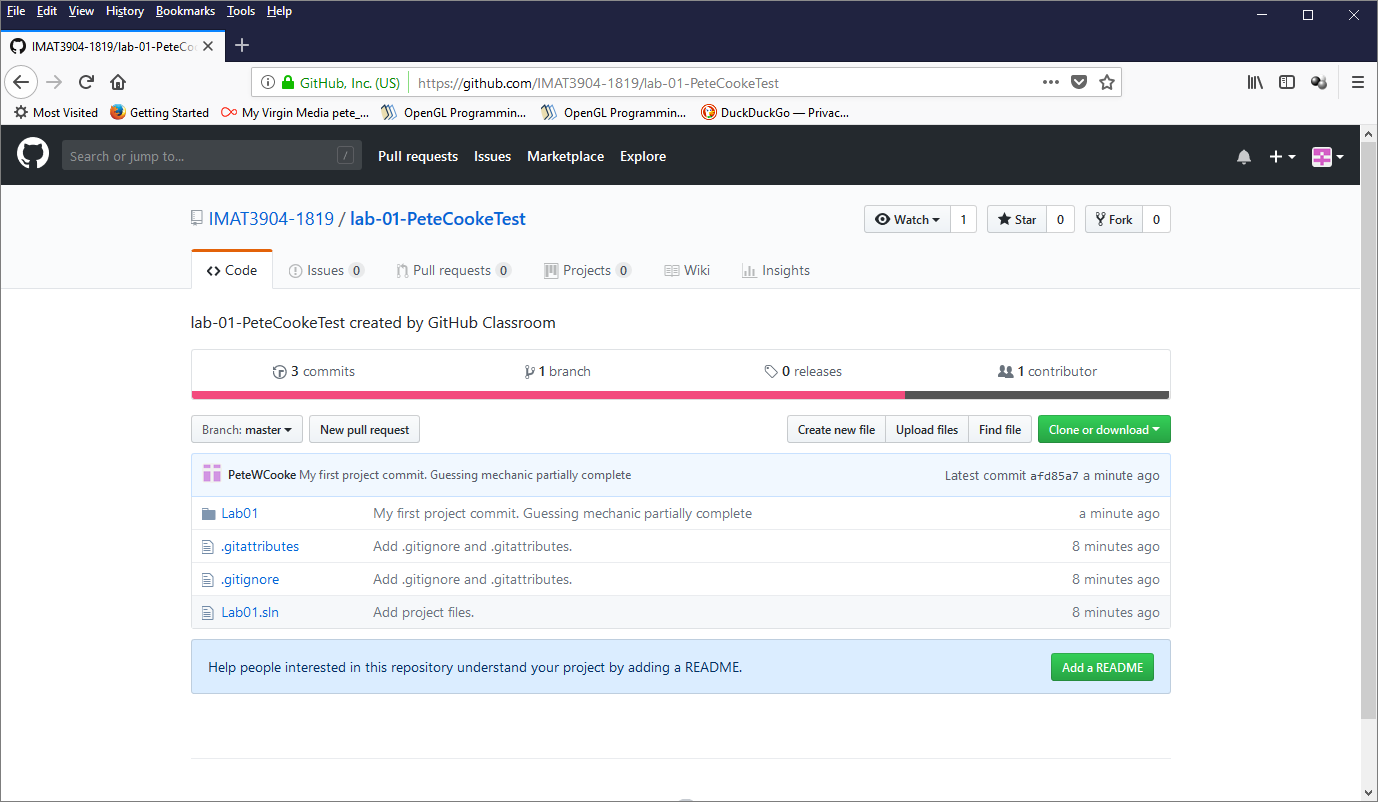


We want to commit these changes to the local repository first. Enter a suitable message e.g. “My first project commit. Guessing mechanic partially complete.” And click ‘Commit All’. This effectively stages a suitable stopping point in our workflow that we can come back to at any point, but does not change the online repository which may be shared by many people.

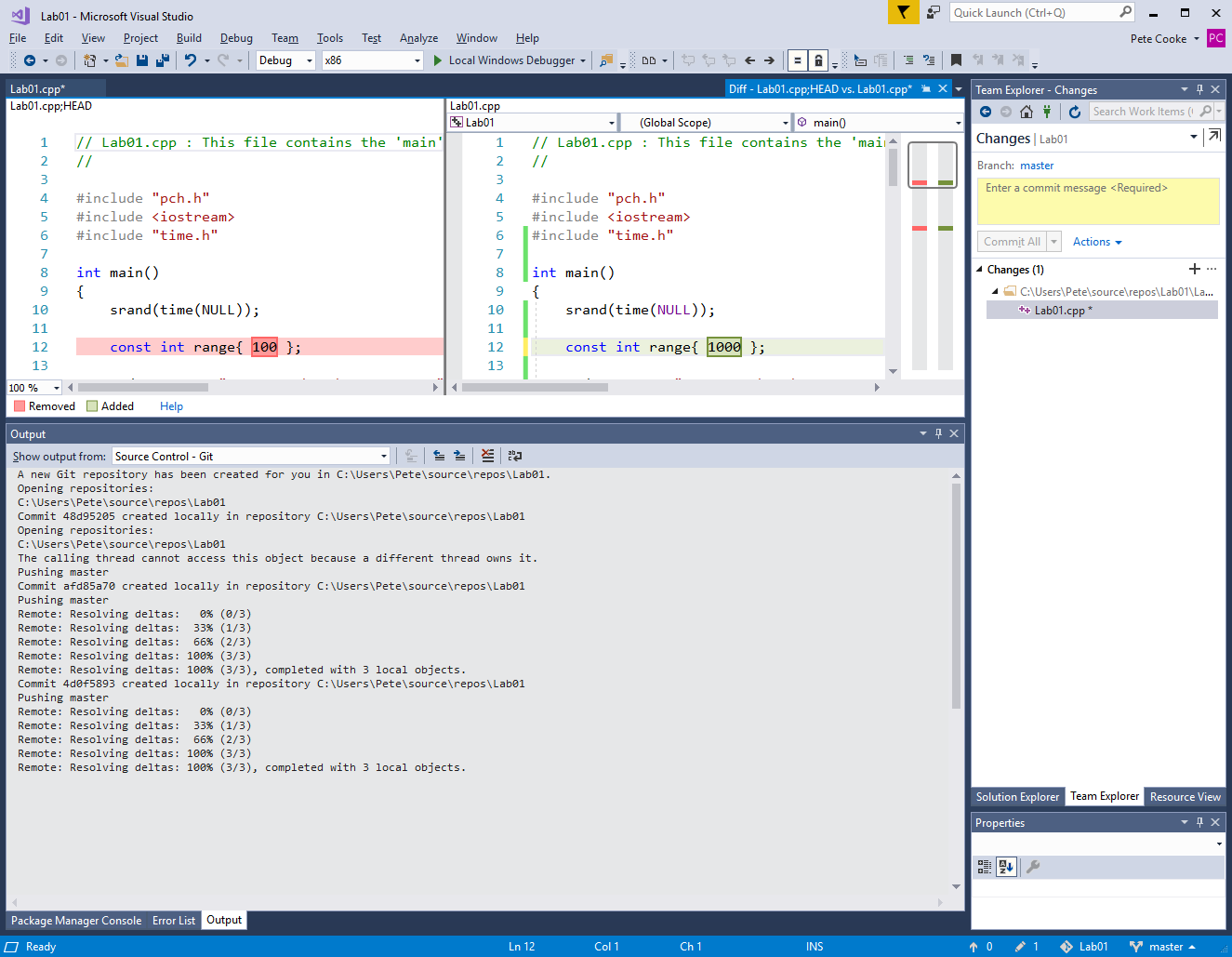
Once you are satisfied with your work, you can upload it by ‘Pushing’ it to the server. You can use the Sync drop down menu to do this.



Online you should now be able to see the changes you have just made…

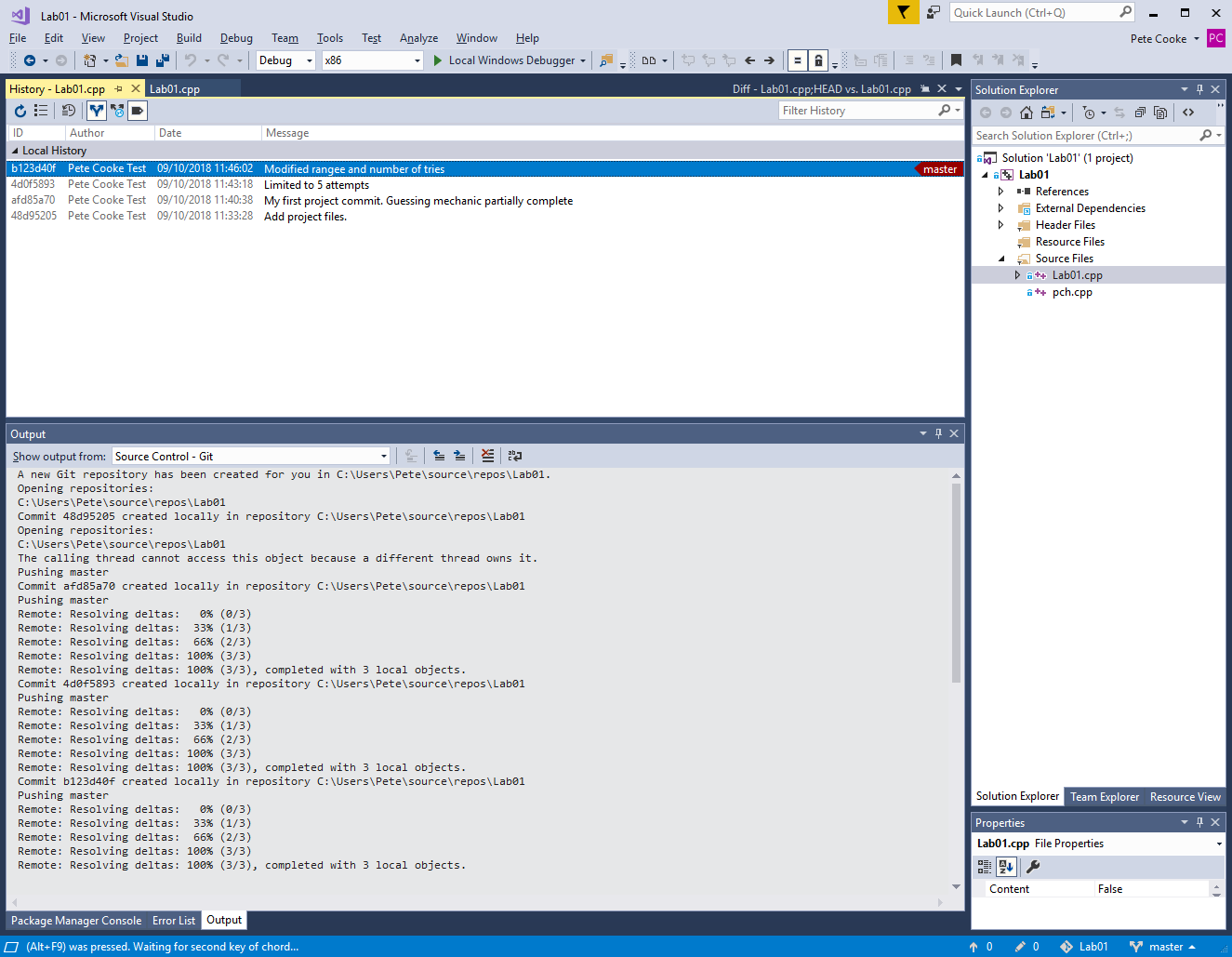


Change the number of guesses allowed to 8 and the randomly generated number to between 1 and 1000. Right click on your main cpp file n Team Explorer and click ‘Compare with unmodified’. This will give you a nice visual highlight of the changes.



Go ahead and commit your changes and push them to server as before.

Right-click your main cpp file in Solution Explorer and click View History. This will bring up a list of your commits. You can double click each one to view what the file was like at that stage of development. The changes are no good so let’s perform a hard reset back to the previous commit by right-clicking on it and choosing reset. The hard reset option deletes the changes permanently.



We have now performed some of the basic options for git, and you will get better at it the more you use it. For more visual studio git tutorials such as for making branches to try new code ideas, you can look at <https://docs.microsoft.com/en-us/vsts/git/tutorial/branches?tabs=visual-studio>

Alternatively you can try the github application installed on the university machines or even the git commands themselves via git bash command prompt. You can try them all and see which one works best for you.