**Using Git/Github**

Git is a version control system that tracks your changes when working with software.

GitHub is a remote web based repository for storing this software. It is a repository where you can get a free account for storing Git-controlled software. This does not have to be limited to application development either, as Git allows you to use version control for any electronic media, such as documents. GitHub provides a convenient place to store multiple versions of these files. Github isn’t the only remote repository. Bitbucket is another popular one.

Git and Github allow you to collaborate on software development using braches and pull requests to develop your software see <https://guides.github.com/activities/hello-world/> for a quick tutorial. We will use it in a very basic form during this semester but you should develop your expertise in version controlling your software throughout your academic career here.

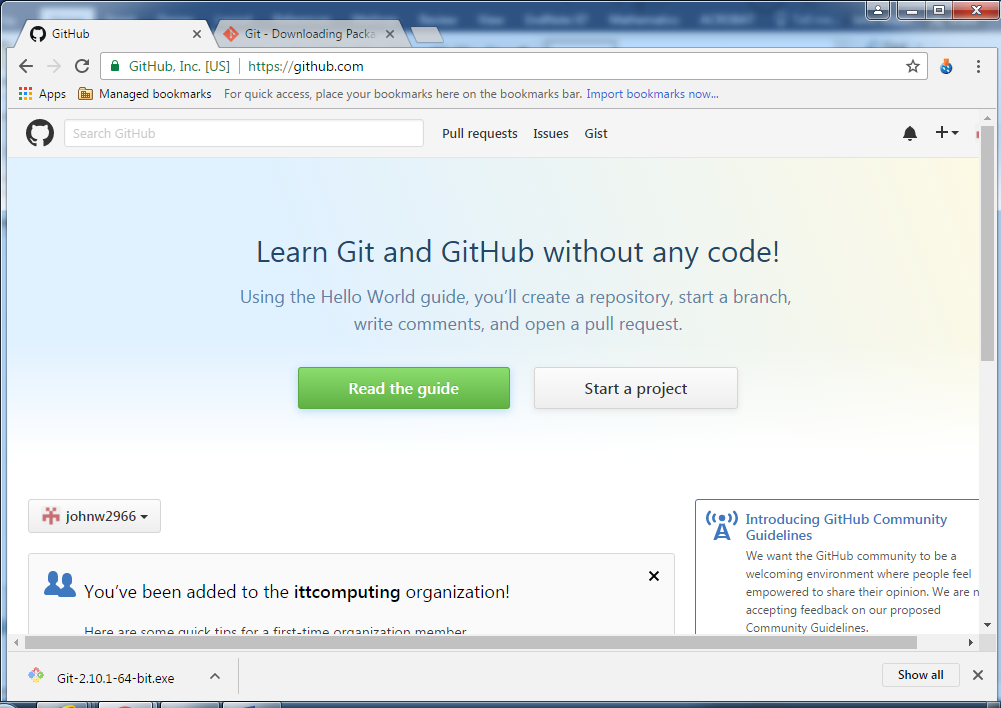
(Investigate Github Student Developer Pack. It will allow you have a free private repository. A repository of you software projects is a good idea, allowing you to keep a portfolio of your work to show to potential employers if you wish)

Git Using Git and GitHub for version control:

**Remote side**

**Step 1** : Create an account on GitHub ( GitHub will host your remote repository for your projects)

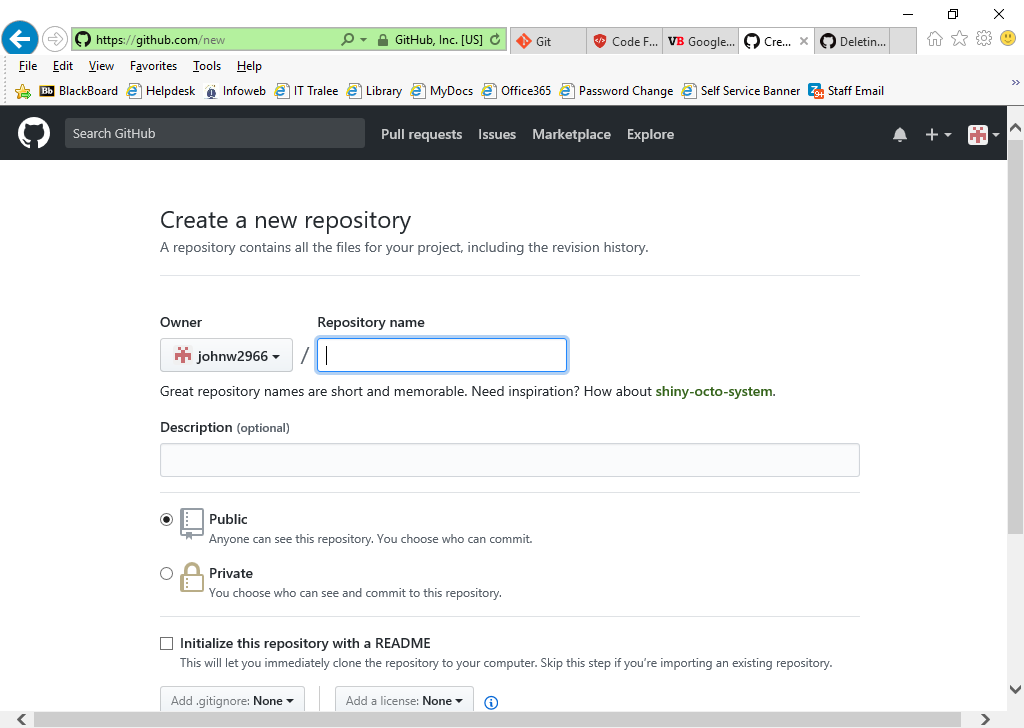
When you open your account you should see:



Click start a project

**Step 2**: Create a new repository:

* Give the repository a name: e.g. OOP-Assign-2017
* Select ‘initialise this repository with a README’ (if available)



Your repository should now look something like this:

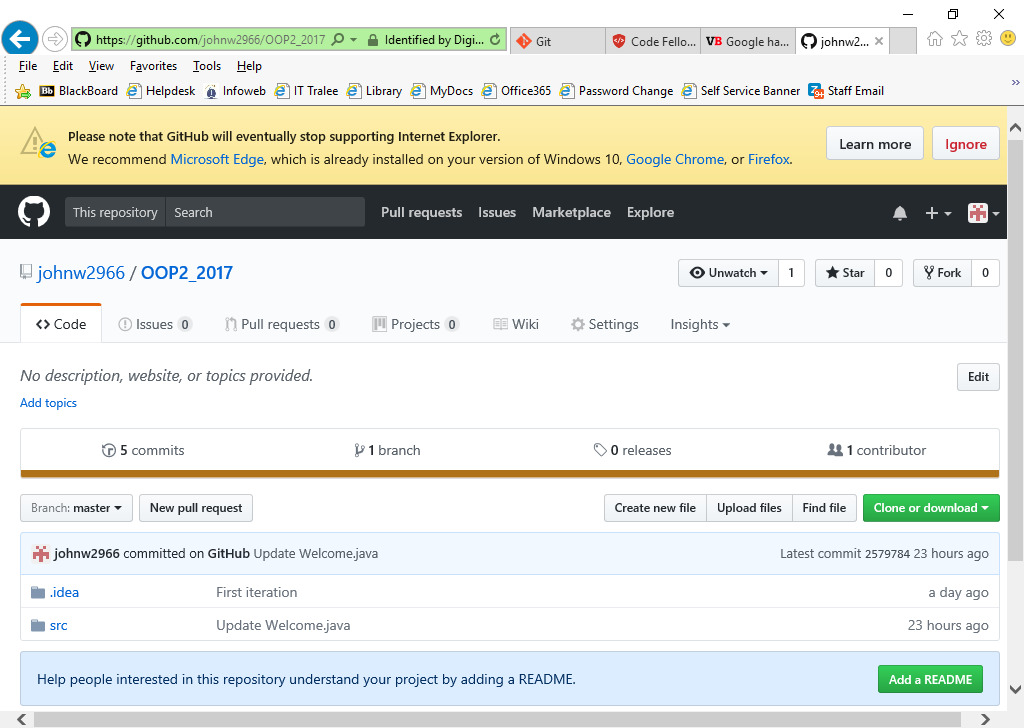


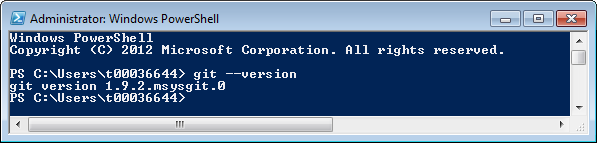
Fig 1: Remote repository

* Copy the URL for download or cloning the repository to the clipboard

**Now for the local side**

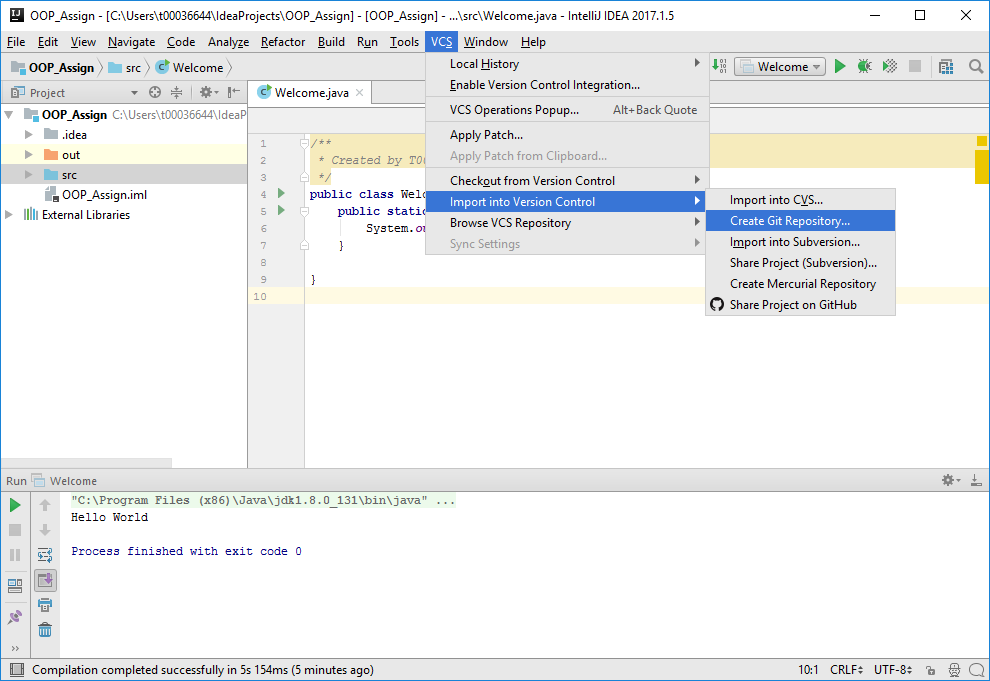
Some IDEs have support for git/github built into them. IntelliJ is one such environment. Below is an example of how we will be utilising IntelliJ to integrate with Git/Github for the moment. Further information can be found at <https://www.jetbrains.com/help/idea/using-git-integration.html> . Also included later in this document is an example of command line instructions to carry out the same command as included in IntelliJ.

**Preliminary Step**: Launch powershell from the command prompt and check that git is installed on your local machine:



**Step 1**

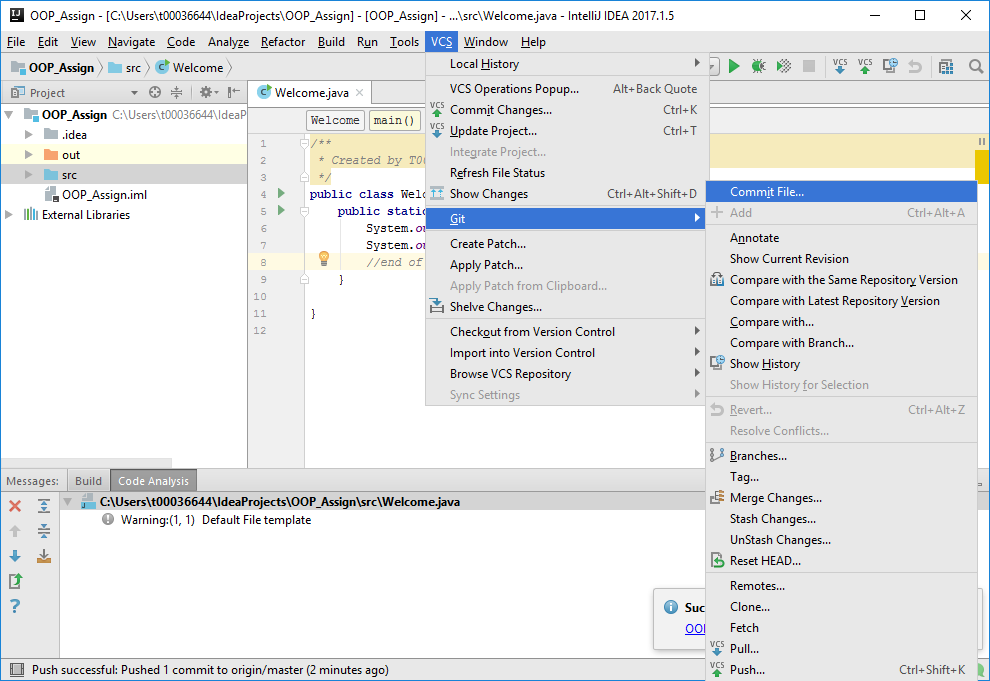
Create a local git repository for your project work. (Note when you push your project work to Github, you will be using the Checkout from Version Control option.)

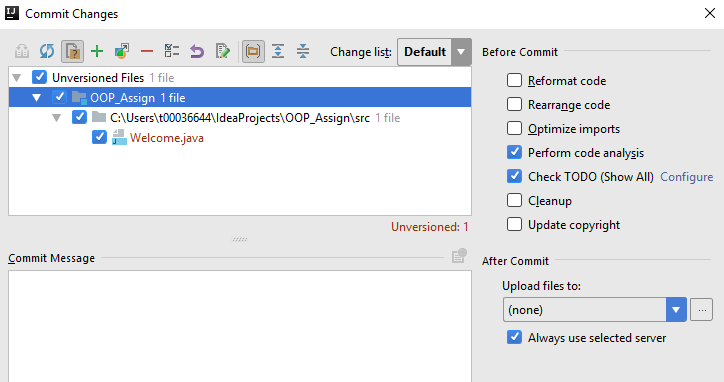


Select the default option for your local repository or create a new folder yourself.( Note this will only store your work locally. You will need to push all your work to the remote repository before leaving the lab).

**Step 2**

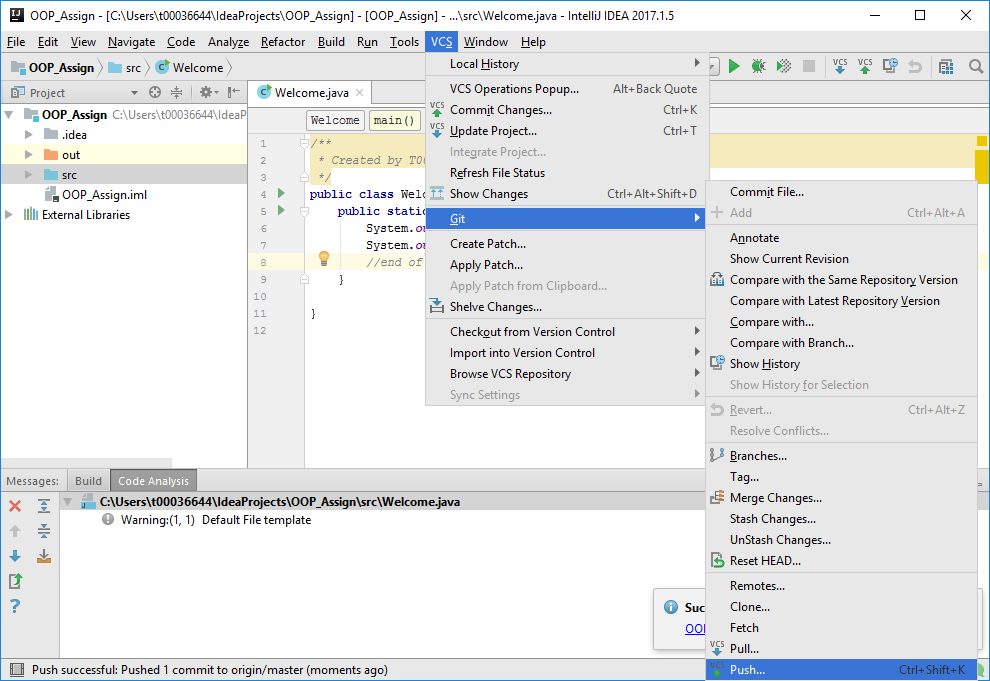
When you have fished developing your code in a session, you will need to ‘commit’ your changes to the local git repository. You will be prompted to insert a commit message as part of this process e.g. ‘Added few file of project with initial code’, ‘Restructured looping structure for client input’, etc.





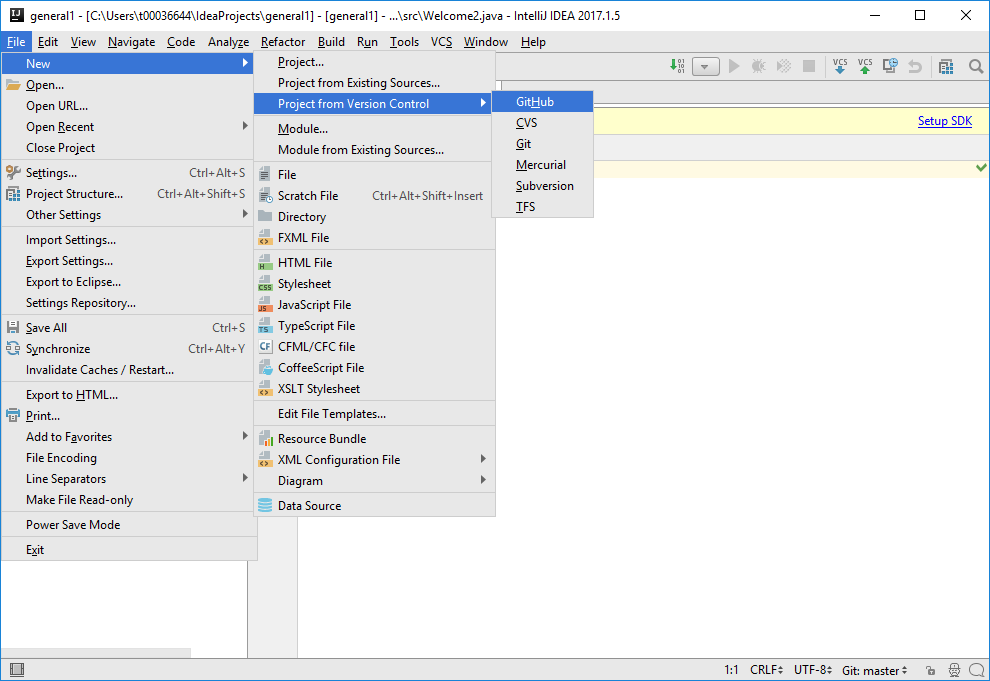
**Step 3**

**We need to push our local changes after the commit stage to update our work on our remote github repository.**

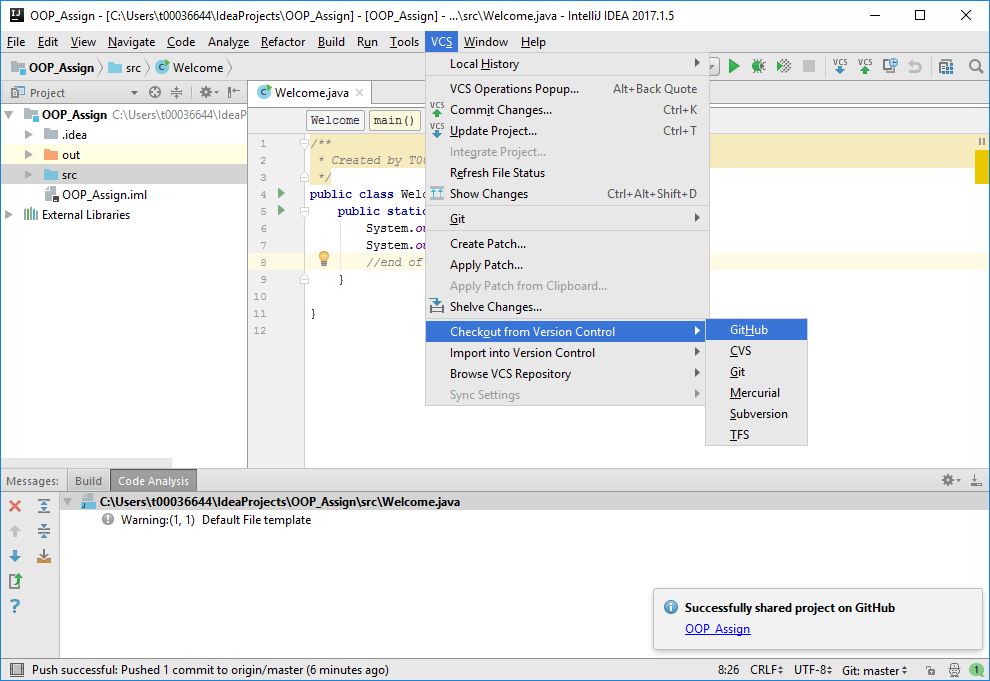


**Step 1 Alternative**

**If you already have a project created in a remote repository and you want to work on it, you will need to check it out from github and ‘clone’ it locally.**

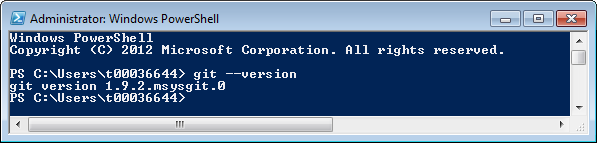


**or**



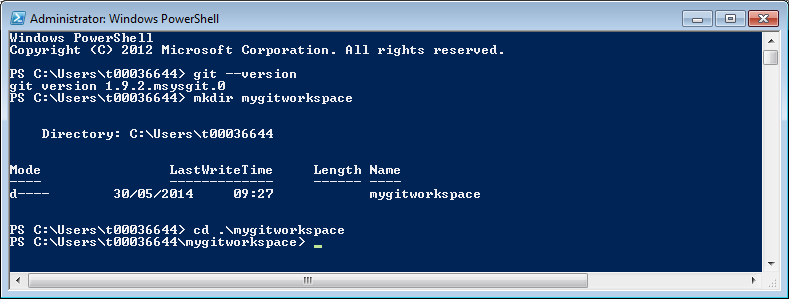
**Optional exercise: Using Command line only**

**Step 1’**: Launch powershell from the command prompt and check that git is installed on your local machine:

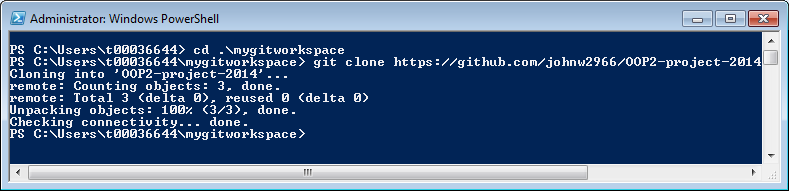


**Step 2’**: We are now going to create a clone of our remote repository. You will need the URL to the remote repository for this. You will find it as indicated in Fig 1 above.(right click to copy url in powershell)

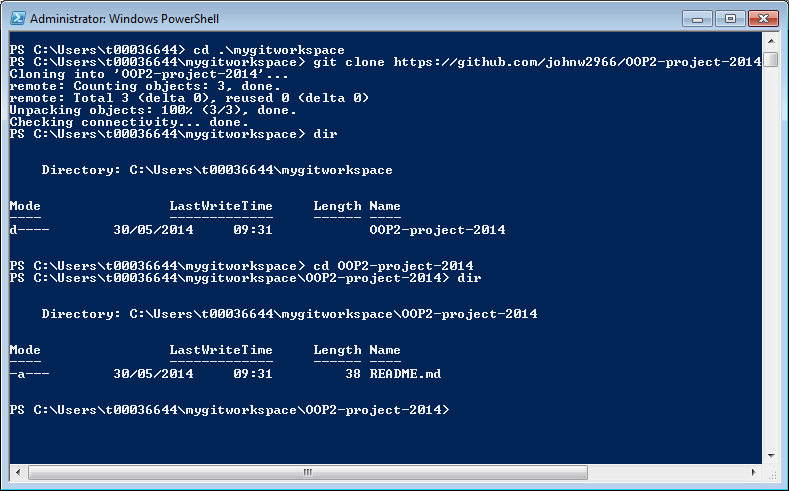
* Make a directory on your local machine (called ‘mygitworkspace’) below.
* Change directory to the new directory. Note: when using cd command in powershell you should enclose a directory name in quotation marks if a space exists in the name.



We make a clone using ‘git clone’…..command; Note when copying URL to the clipboard, you can not use ^v to copy it into powershell. Instead you can just right click with the mouse.



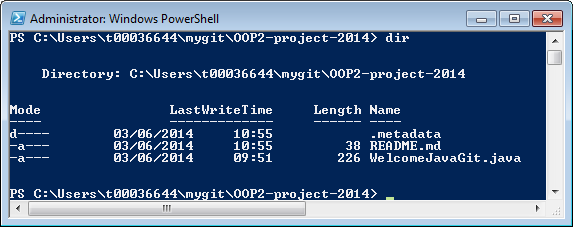
You should now be able to see the local copy of you remote repository:



**Step 3’:** Now that we have a local copy we make changes to our project

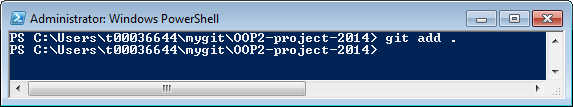
We will be creating a simple java class and submitting this to the remote repository:

After changes you can see that the project has been added to your local git repository.

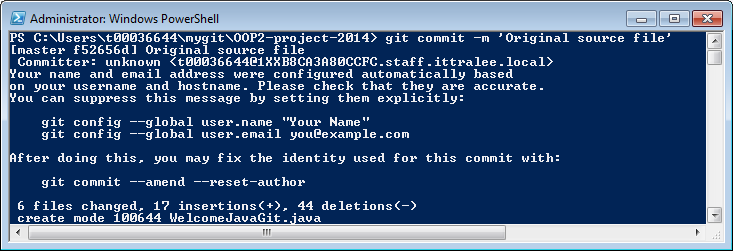


**Step5**: Having created/modified our project we are now ready to push it up to our remote repository on GitHub.

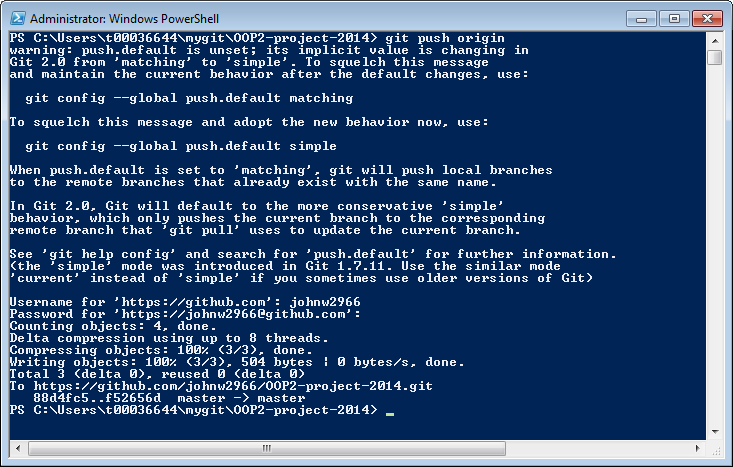
There are a number of steps required…..check status, add changes to staging area, commit changes, push changes back to the origin.



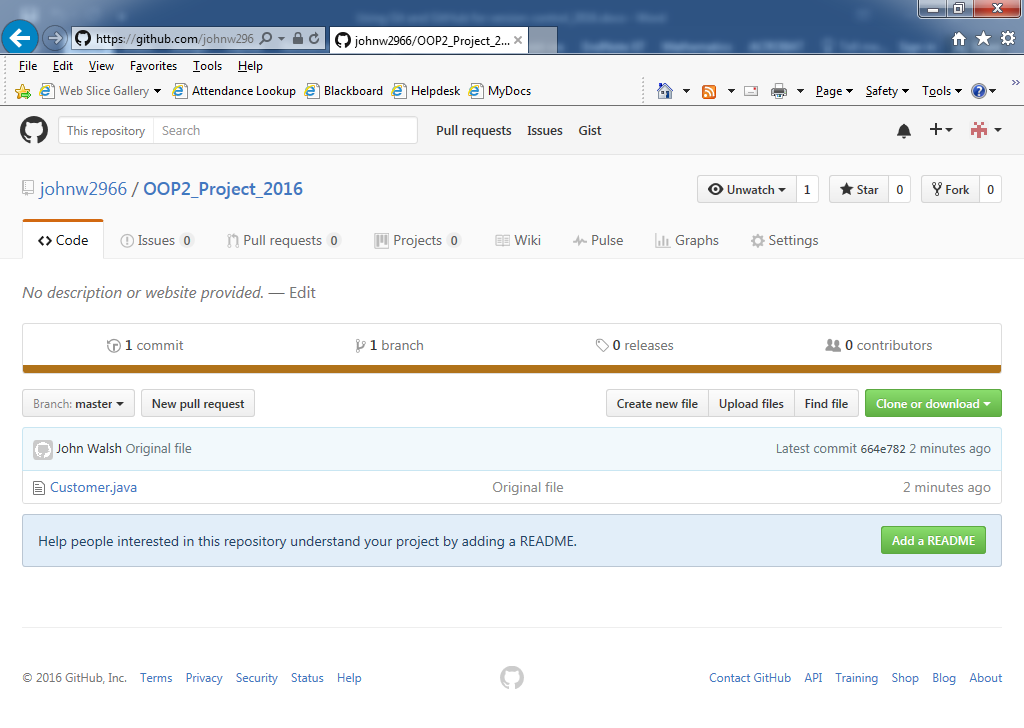
\*\*Write a message with the git commit indicating what changes you have made between this version and the last version. E.g. git commit –m ‘Original source file’



\*\* You should indicate the author of the changes using the git commands as indicated above.



You have now push modification back to the remote repository.



Note : to remove files directories from your repository use the git rm –r directoryname/filename command.

**Step 6**: Changes can be reviewed

