Git Bash Installation Guide

Step 1: Before proceeding make sure you have FIRST installed GitHub Desktop. If not, you will have to do a few additional setup tasks that are listed at the end of this document. Note: using Git Bash is optional in ELECTENG 209.

Step 2: When working with Git, you will require a text editor. Though Notepad can be used, it is recommended to use a much more capable editor like Visual Studio Code. So, let's download Visual Studio Code as shown in Fig. 1.

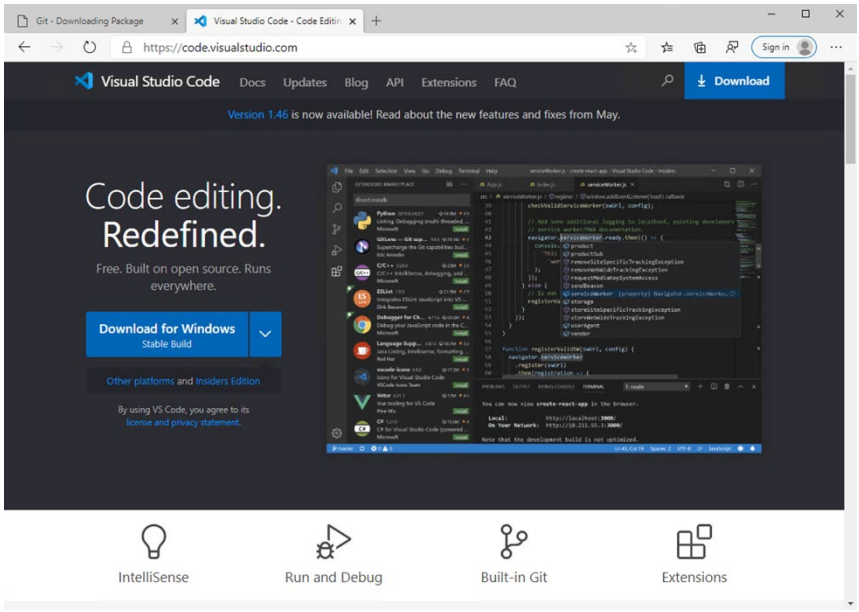

Fig. 1: Step 2

Step 3: Run the installer. This will launch the "License Agreement" and prompt you to accept it as shown in Fig. 2(a). Tick the checkbox next to "I accept the agreement" and click "Next" to proceed.

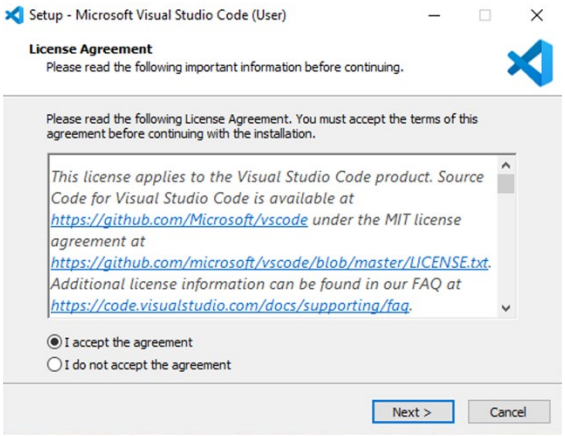

Fig. 2(a): Step 3

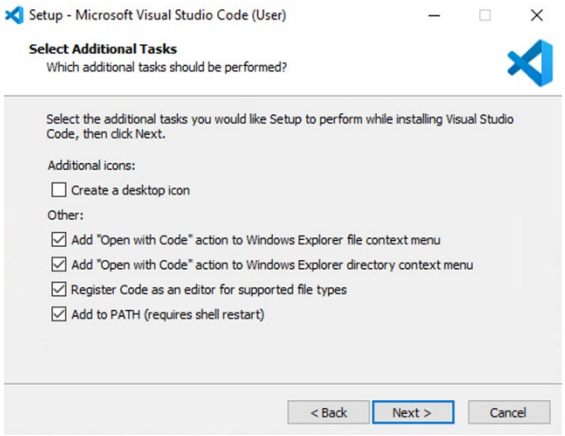

Fig. 2(b): Step 4

Step 4: You will be given a few setup options. As shown in Fig. 2(b), you may select all options listed under “Other:” by ticking the checkboxes next to each item. Then click “Next” to proceed.

Step 5: As shown in Fig. 3(a), click “Install” to install Visual Studio Code.

Step 6: As shown in Fig. 3(b), ones the installation is complete, run Visual Studio Code to confirm it has been installed correctly.

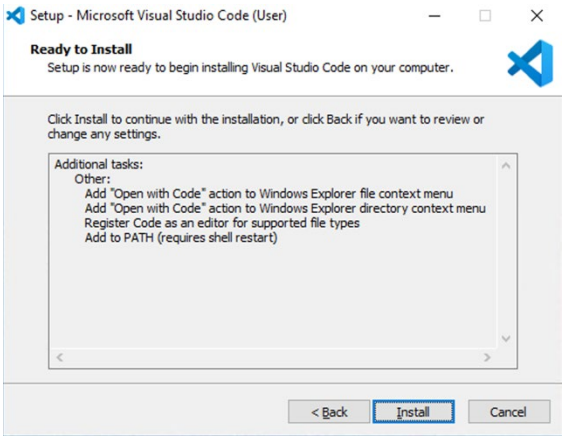

Fig. 3(a): Step 5

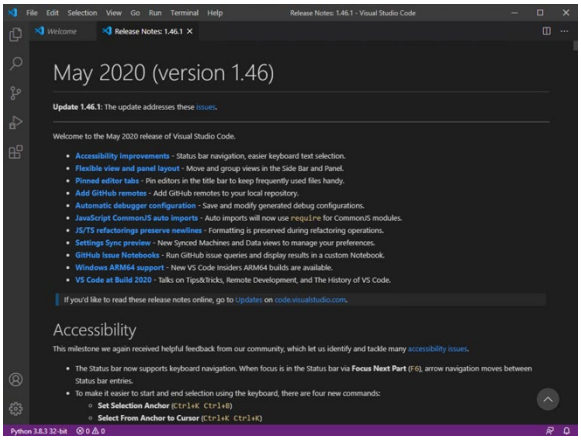

Fig. 3(b): Step 6

Step 7: Next let’s install Git Bash. First download Git using the link as shown in Fig. 4. If installing on a Mac (or Linux) you may consider using a package manager like Homebrew.

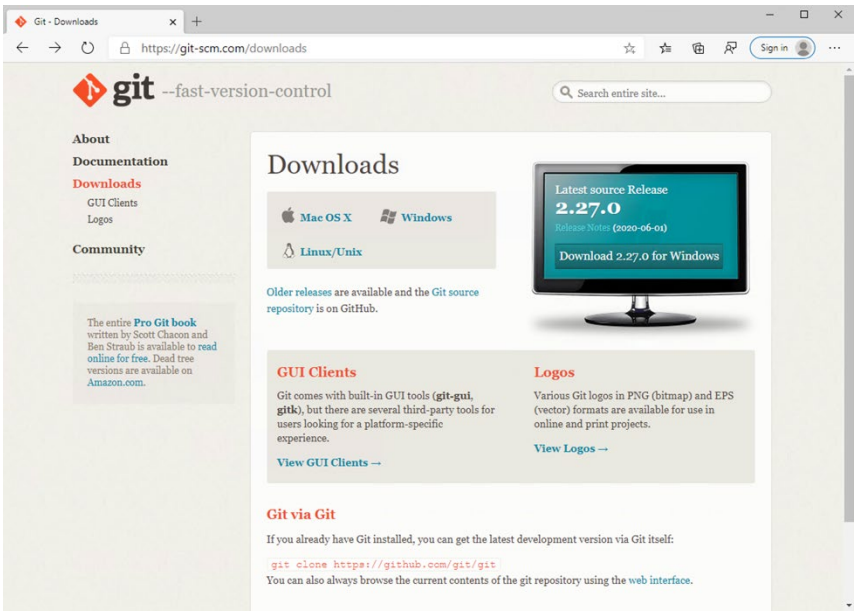

Fig. 4: Step 7

Step 8: Run the Git installer. You will be prompted license information as shown in Fig. 5(a). Click “Next” to proceed.

Step 9: You will be prompted to select the installation directory as shown in Fig. 5(b). Keep the default location and click “Next” to proceed.

Fig. 5(a): Step 8

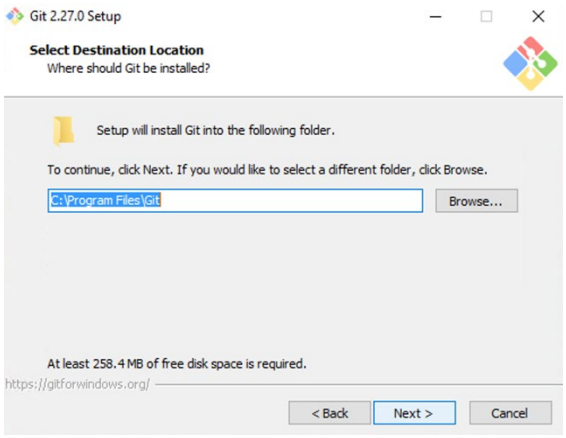

Fig. 5(b): Step 9

Step 10: You will be prompted to select the components that will be installed. As shown in Fig. 6(a), use the default settings. In addition to the default settings, you may decide to enable daily update checks as indicated in Fig. 6(a). Click “Next” to proceed.

Step 11: You will be prompted to select a start menu folder. As shown in Fig. 6(b), use the default settings. Click “Next” to proceed.

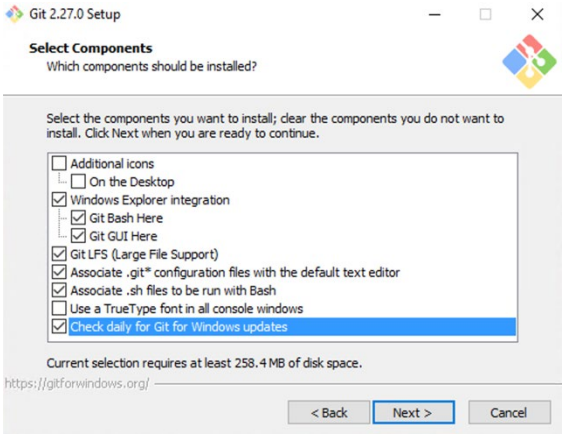

Fig. 6(a): Step 10

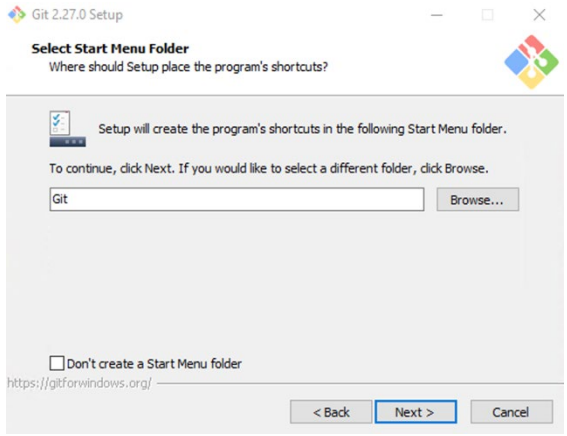

Fig. 6(b): Step 11

Step 12: You will be prompted to select the default editor used by Git as shown in Fig. 7(a). If you installed Visual Studio Code as per Steps 2-6, then select “Use Visual Studio Code as Git’s default editor”. Alternatively, you may select your preferred editor from the list. Click “Next” to proceed.

Step 13: You will be prompted the window shown in Fig. 7(b). As shown in Fig. 7(b), use the default recommended setting. Click “Next” to proceed.

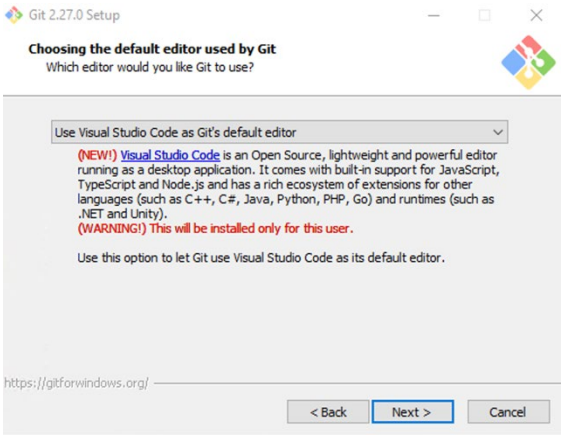

Fig. 7(a): Step 12

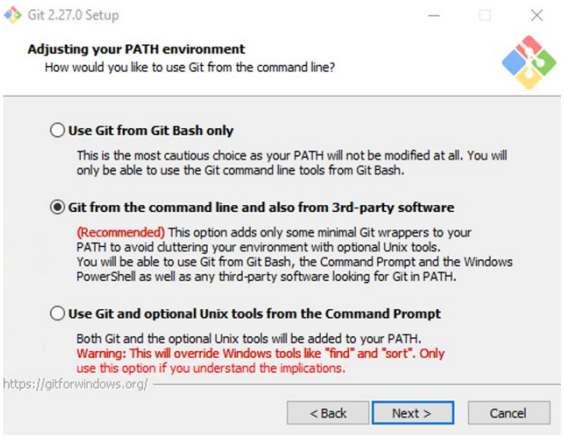

Fig. 7(b): Step 13

Step 14: You will be prompted the window shown in Fig. 8(a). As shown in Fig. 8(a), use the default OpenSSL setting. Click “Next” to proceed.

Step 15: You will be prompted asking the configuration for line ending conversion (i.e. Line Feed and Carriage Return). As shown in Fig. 8(b), use the default setting. Click “Next” to proceed.

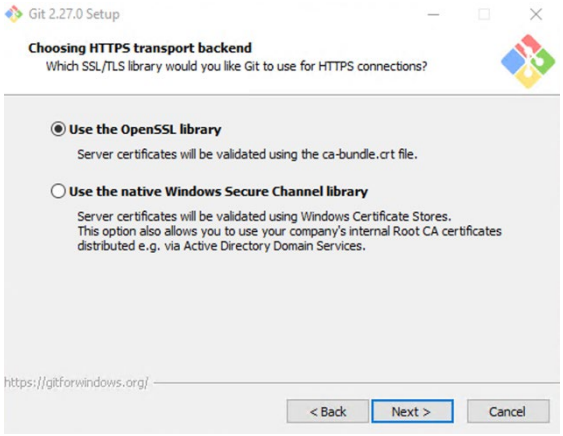

Fig. 8(a): Step 14

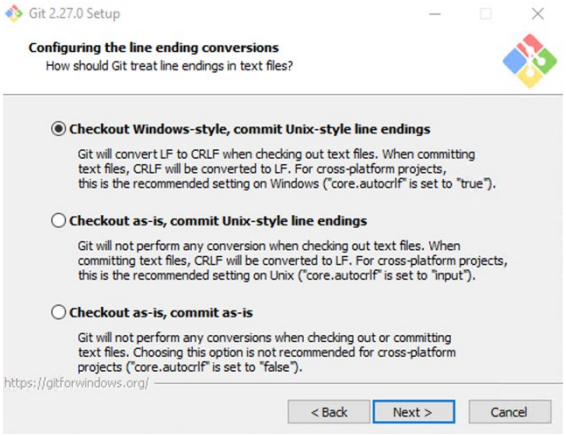

Fig. 8(b): Step 15

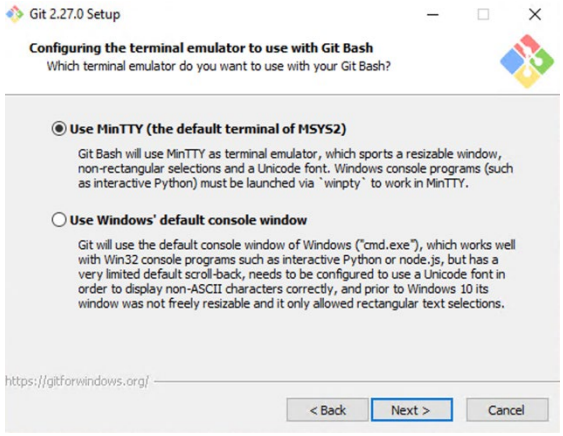

Fig. 9(a): Step 16

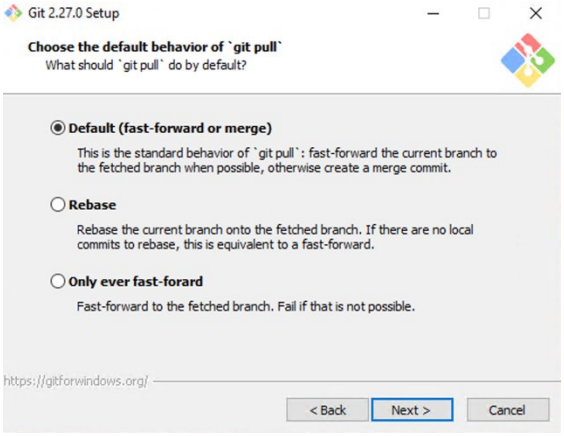

Fig. 9(b): Step 17

Step 16: You will be prompted asking the terminal emulator configuration. As shown in Fig. 9(a), use the default MinTTY setting. Click “Next” to proceed.

Step 17: You will be prompted asking the default behavior of ‘git pull’. As shown in Fig. 9(b), use the default setting (i.e. fast-forward or merge). Click “Next” to proceed.

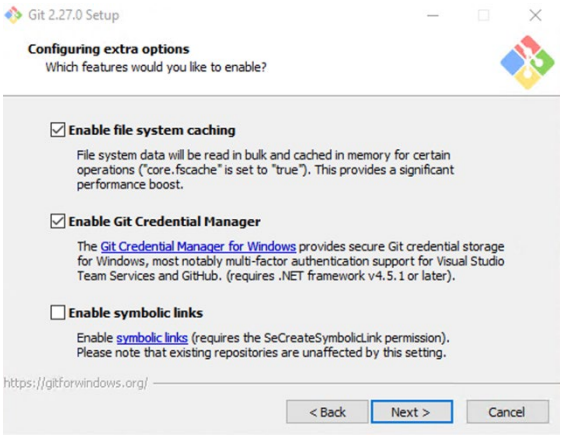

Fig. 10(a): Step 18

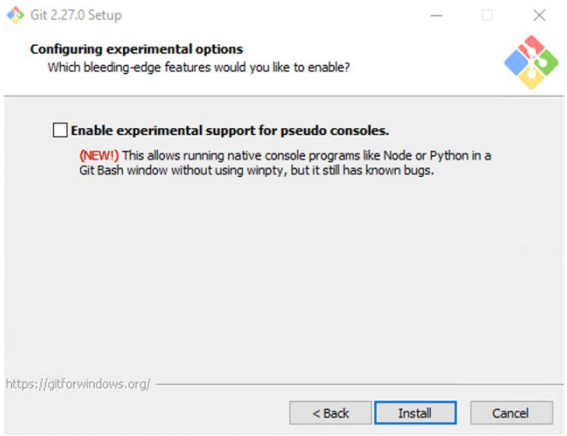

Fig. 10(b): Step 19

Step 18: You will be asked about a few extra configuration options. As shown in Fig. 10(a), keep the default options enabled. Click “Next” to proceed.

Step 19: You will be asked if you like to enable experimental features. As shown in Fig. 10(b), do not enable these options. Click “Install” to install Git.

Step 20: Ones the installation is complete the window shown in Fig. 11 will appear. Untick both options and click “Next” to complete the installation.

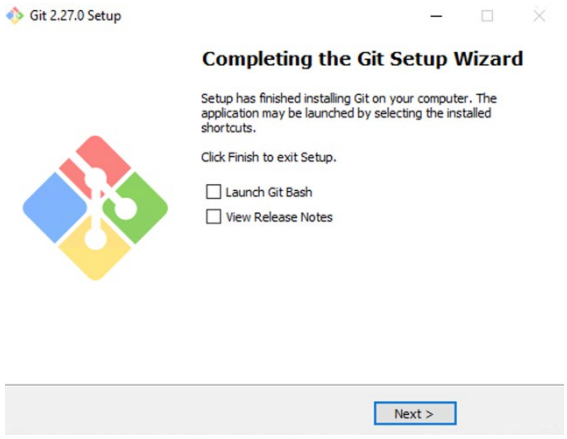

Fig. 11: Step 20

Step 21: Now let’s check if Git Bash has been installed and setup correctly. As shown in Fig. 12, from start menu open Git Bash.

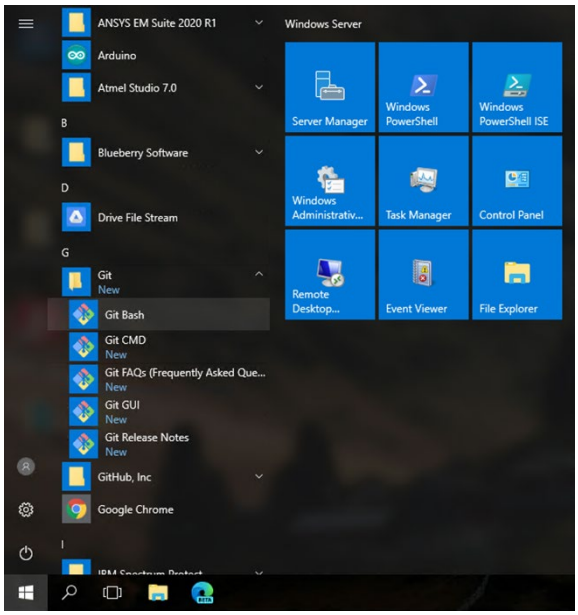

Fig. 12: Step 21

Step 21: As mentioned in Step 1, if you had installed GitHub Desktop, then it would have created a global 'git config' file with information like your user name and email. You can check these settings by entering the command below in Git Bash. When you run this command in Git Bash, if you had correctly setup GitHub Desktop prior to installing Git Bash, you will see user.name and user.email fields populated with your information. An example is shown in Fig. 13(a). If the global 'git config' file is not correctly setup you will see something similar to Fig. 13(b).

```
git config --global --list
```

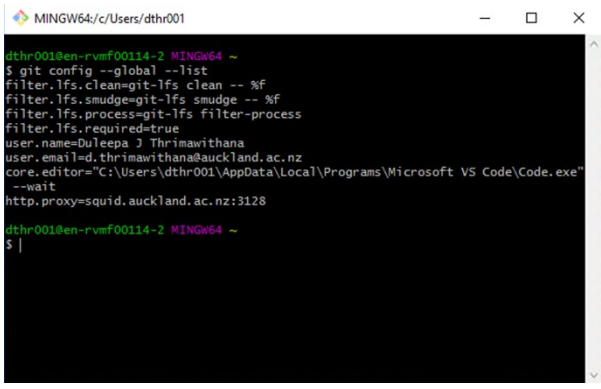

Fig. 13(a): Step 21 – Correct Setup

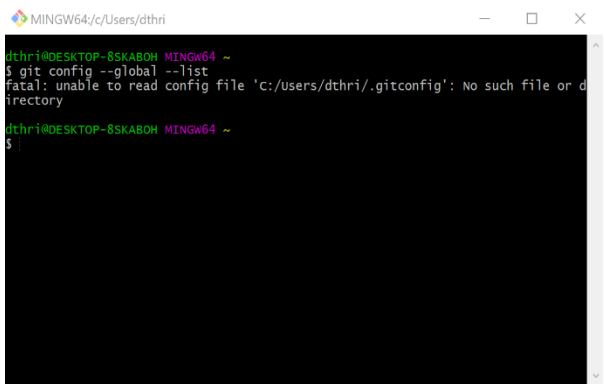

Fig. 13(b): Step 21 – Incorrect Setup

Step 22: If the global 'git config' file is not setup correctly, or you wish to update the details, run the two commands given below on Git Bash as shown in Fig. 14(a). This is a onetime setup and you do not have to run these two commands again unless you like to change your information. The 'git config' file (named '.gitconfig') is stored in the root of your local profile (i.e. C:\Users\YourUserFolder\)

```
git config --global user.name "FirstName LastName"
```

```
git config --global user.email "abcd001@aucklanduni.ac.nz"
```

Step 23: To check if Git Bash has been setup correctly, lets clone a repository. Cloning refers to copying a repository from for example GitHub. For this purpose we have setup a public repository, which can be viewed using the link https://github.com/ee209-2020class/RepoForTesting. As shown in Fig. 14(b), in Git Bash run the command given below to clone (copy) this repository to your local computer.

git clone https://github.com/ee209-2020class/RepoForTesting.git

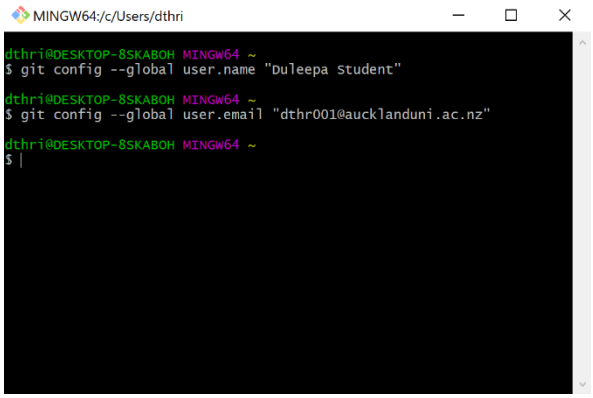

Fig. 14(a): Step 22

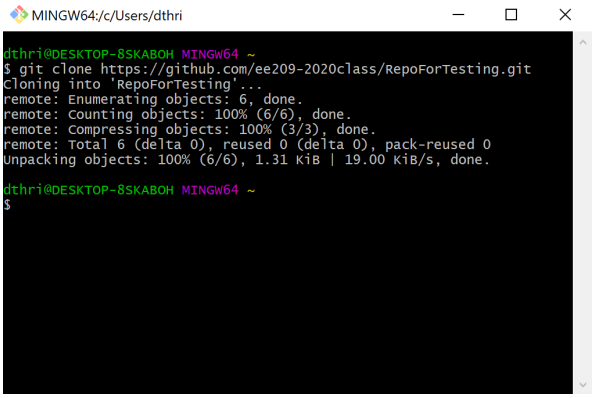

Fig. 14(b): Step 23

Step 24: Open the 'Readme.md' file that is in the repository you just cloned using Visual Studio Code. The repository you cloned will be by default stored in a folder named 'RepoForTesting' and it is located in the root of your local profile. For example, in my case, as shown in Fig. 15(a), it is stored at C:\Users\dthri\ . As shown in Fig. 15(b), add the text "- This line is added for testing" and save.

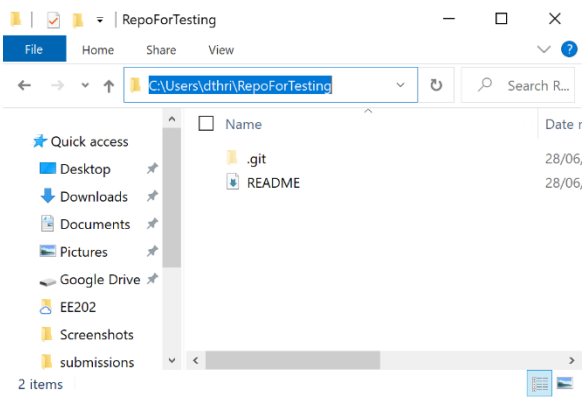

Fig. 15(a): Step 24 – File Location

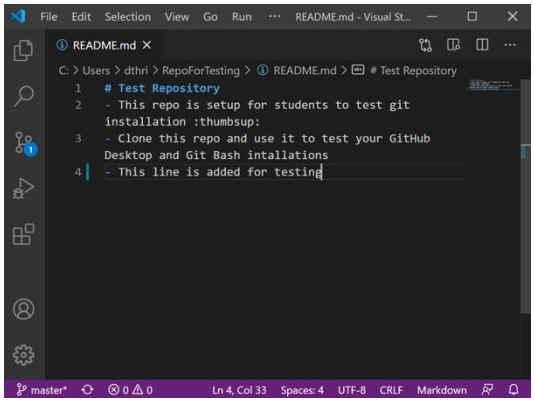

Fig. 14(b): Step 24 – Edited File in VSC

Step 25: Now go back to Git Bash and type the two commands below. First command changes the working folder to the folder 'RepoForTesting'. The second command will tell you modifications made to your local Git repository. Since you have changed the 'Readme.md' file by adding a line and saving, it will say that this file has been modified as shown in Fig. 16(a).

cd RepoForTesting

git config --global user.email "abcd001@aucklanduni.ac.nz"

Step 26: Lets commit all the changes you made to repository by running the command given below. '-a' in the command tells to commit changes to all files. '-m' in the command let you specify a commit message, which in this case is 'Modified the file'. The commit message tells what changes you have made in the files committed. This step is shown in Fig. 16(b). If this works, you have completed setting up Git Bash. During the workshop we will learn how to store your login credentials so that you can push (send) changes made to your local repository to GitHub.

git commit -a -m 'Modified the file'

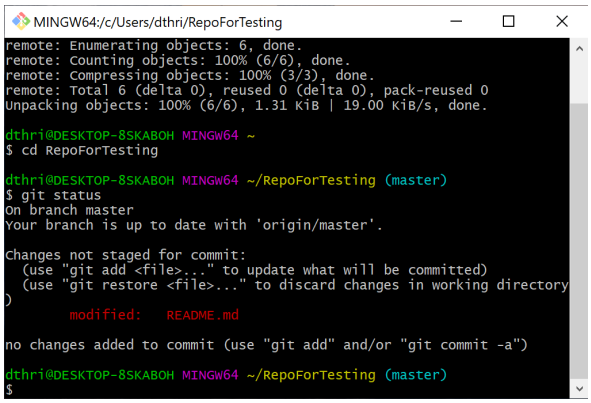

Fig. 16(a): Step 25

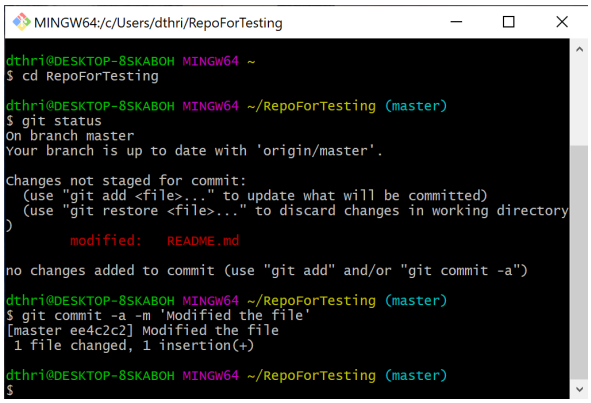

Fig. 16(b): Step 26