ECSE 321 Introduction to Software Engineering

Hands-on Tutorials

McGill University

Table of Contents

- link: HTML version
- PDF version

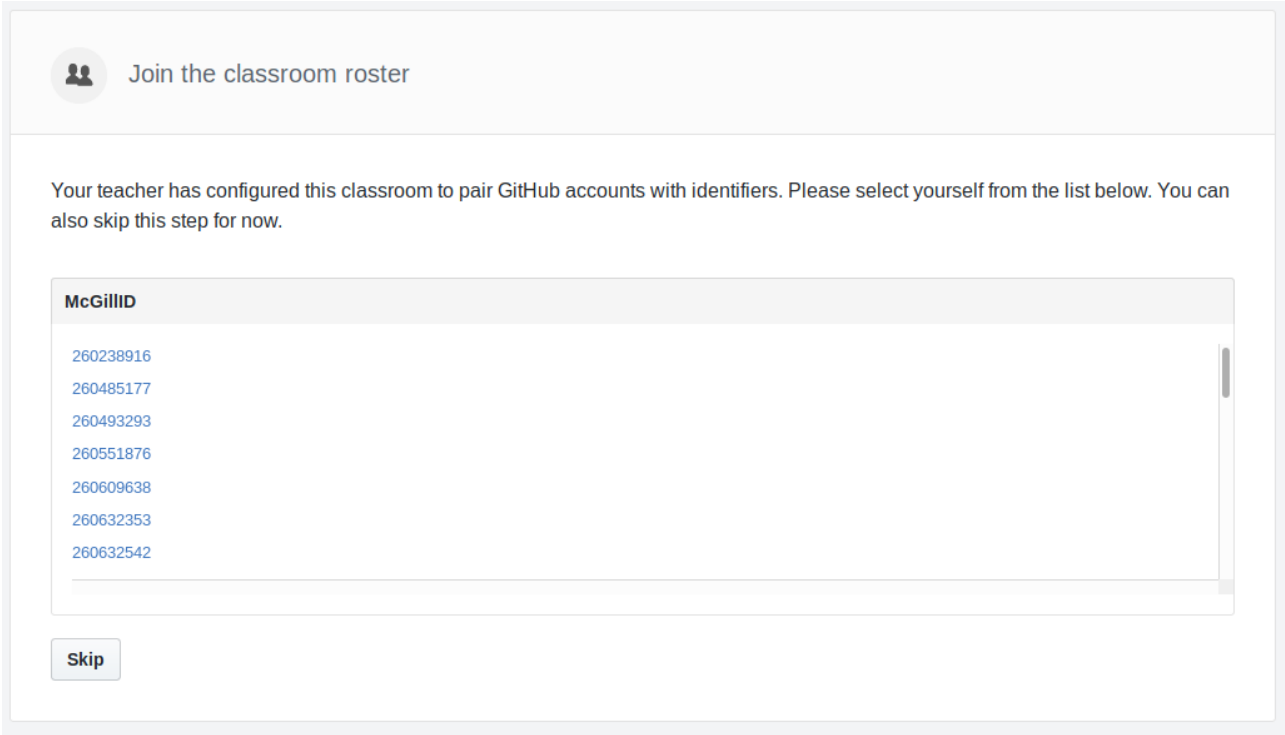
Sections of the tutorial will continuously be published at this web page.

1. Preliminaries

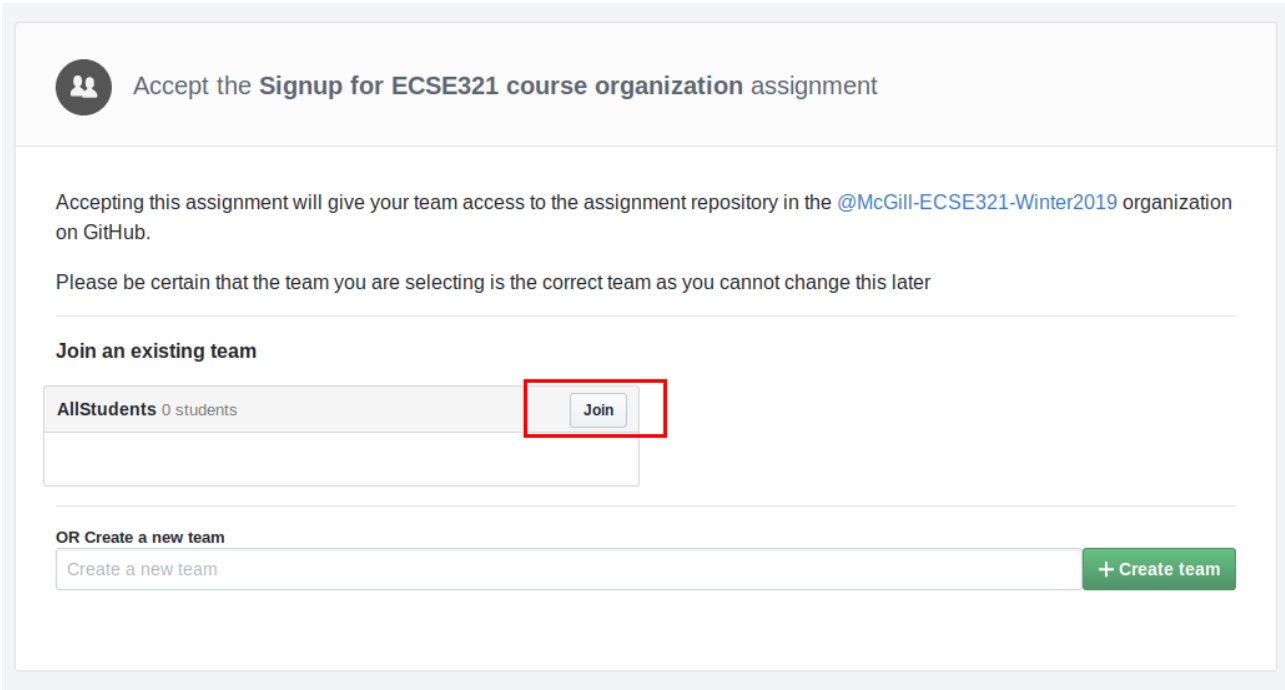
1.1. Getting Started

Steps for signing up for GitHub classroom:

1. Log in/Register on GitHub.
2. Open link https://classroom.github.com/g/o9gWNZis
3. Select your McGill ID from the list



4. Join team *All students*

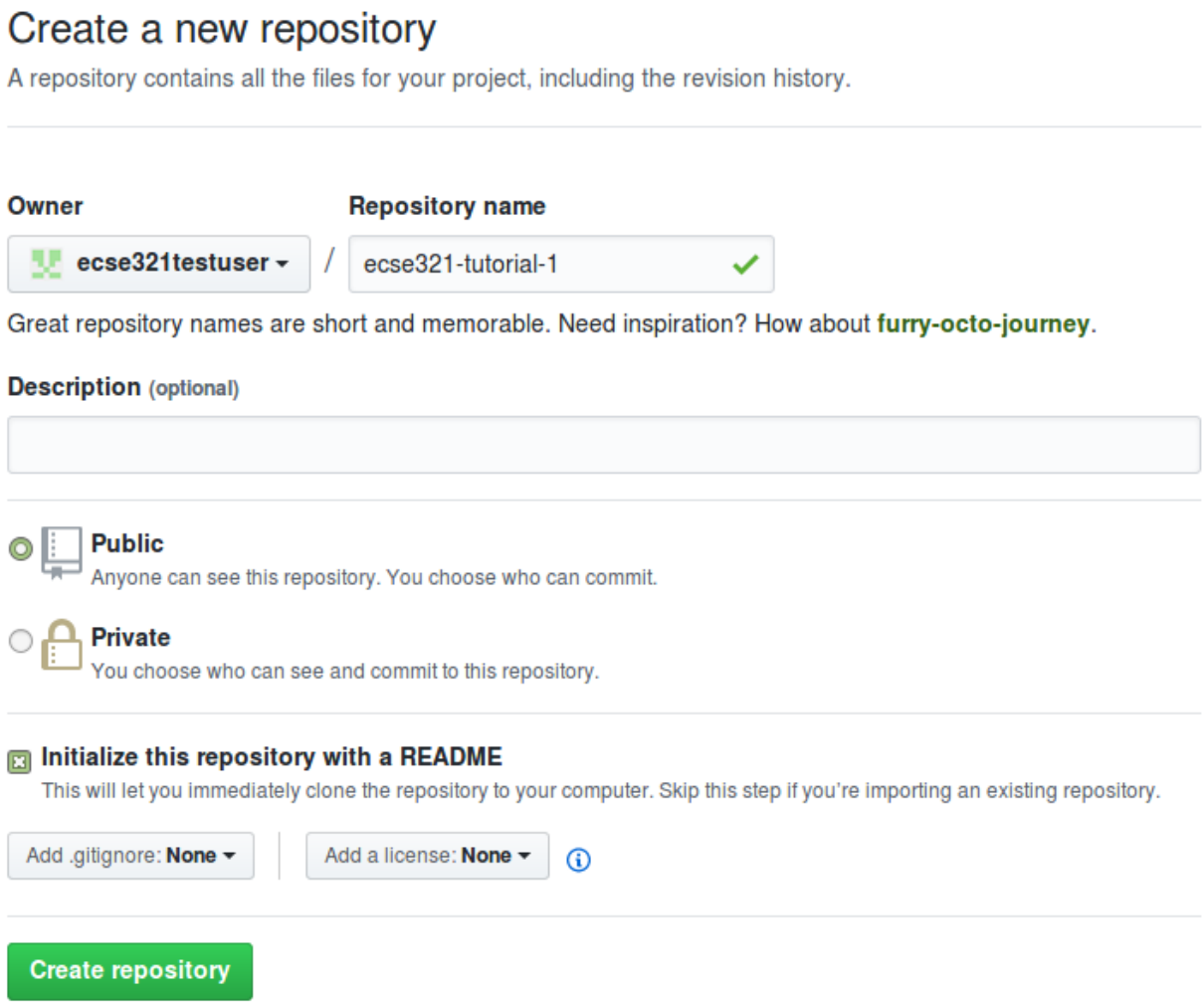


1.2. Project Management Tools for Agile Development

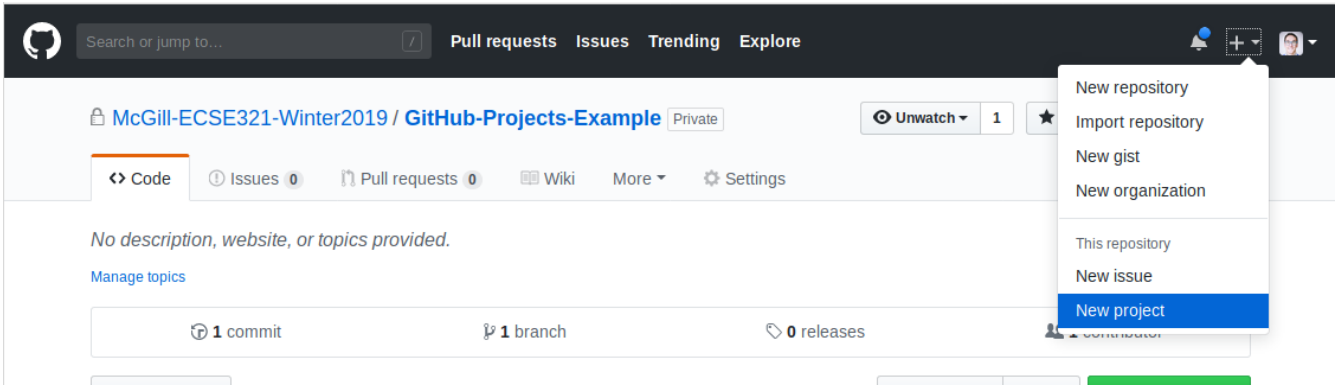
1.2.1. GitHub Projects

First, we create a new repository under everyone's own account to demonstrate the basic features of "GitHub Projects".

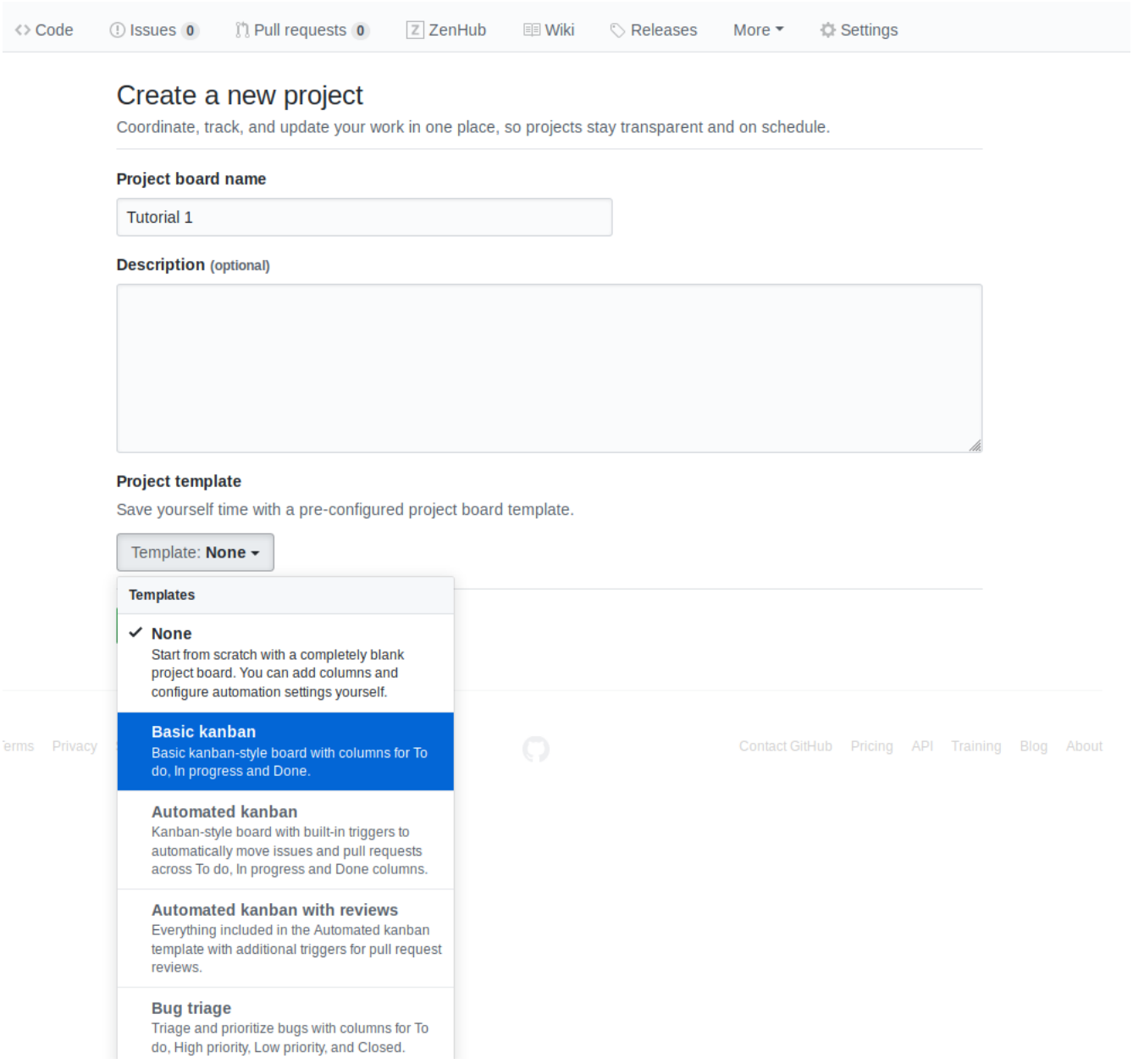
1. Visit https://github.com/ then click on *New repository* (green buttom on the right).
2. Set your user as the owner of the repository.
3. Give a name for the repository (e.g., ecse321-tutorial-1), leave it *public*, then check *Initialize this repository with a README*. Click on *Create repository* afterwards. At this point the remote repository is ready to use.



Once the repository is ready, associate a new GitHub Project and see how their features work. Create a project:

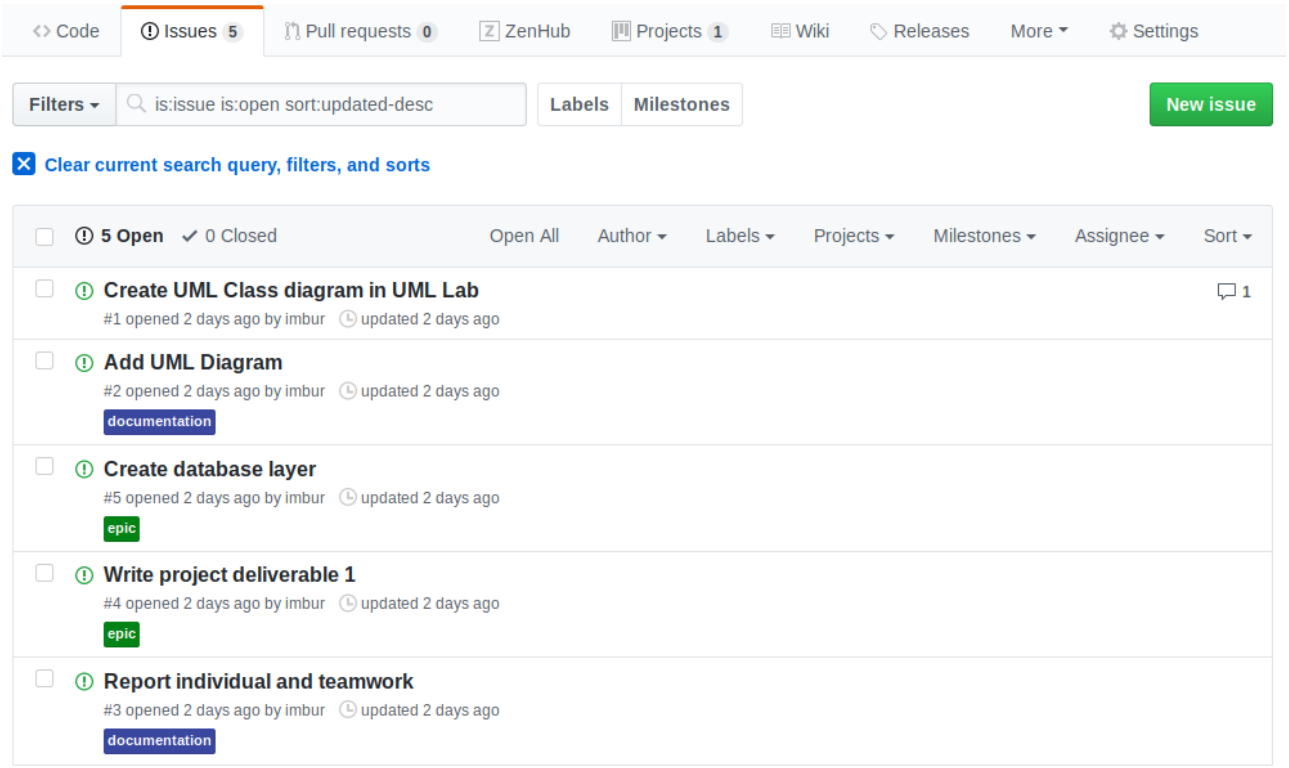


Select Basic Kanban project style:

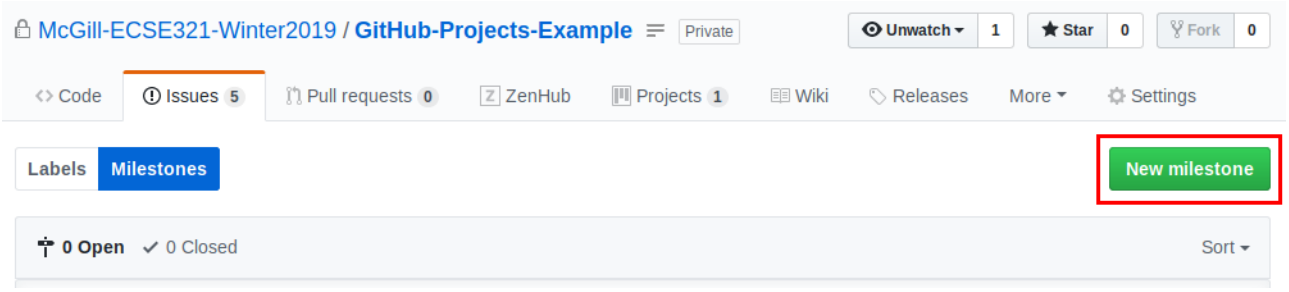


Tasks to complete:

1. Create a few issues to outline the tasks for the first deliverable. Assign them appropriate labels and add yourself as the assignee!



2. Create a milestone for the issues.



3. Create cards from the issues on the project board.
4. See how GitHub track the project progress as you move the cards from the different columns.

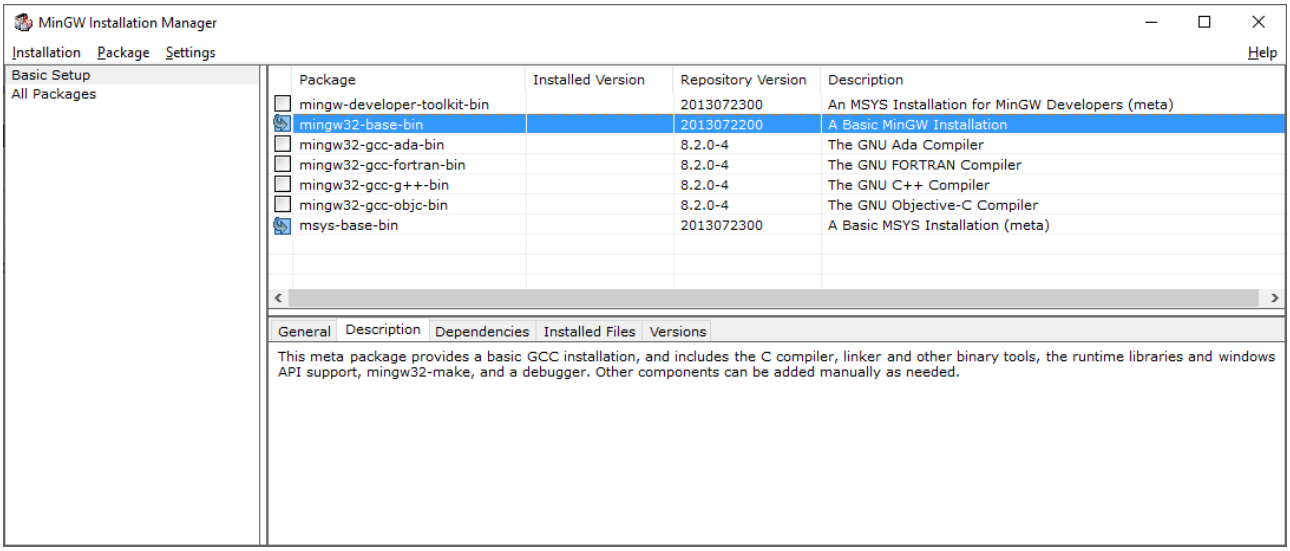
1.3. Command Line Basics

This section shows a few handy terminal commands.

1.3.1. Windows prerequisites

This step can be skipped if you are using MacOS or Linux. However, if you are using Windows, you need to have a terminal that supports the execution of basic Linux commands. Such programs are Git Bash or MinGW, for example. You can find below a few helper steps to get MinGW running on your system.

1. Get the MinGW installer from here
2. Install it to wherever you like, the default installation folder is *C:\MinGW*
3. Once the setup finishes, open the MinGW Installation Manager
4. Select the two packages for installation as shown in the figure below



5. You can open a terminal window by running the executable *C:\MinGW\msys\1.0\bin\bash.exe*

1.3.2. Basic file system operaions

1. Open a terminal, and try the following commands:
   - `pwd`: prints the present working directory
     Example:

     ```
     $ pwd
     /home/ecse321
     ```

   - `ls`: lists the content of a given folder
     Example:

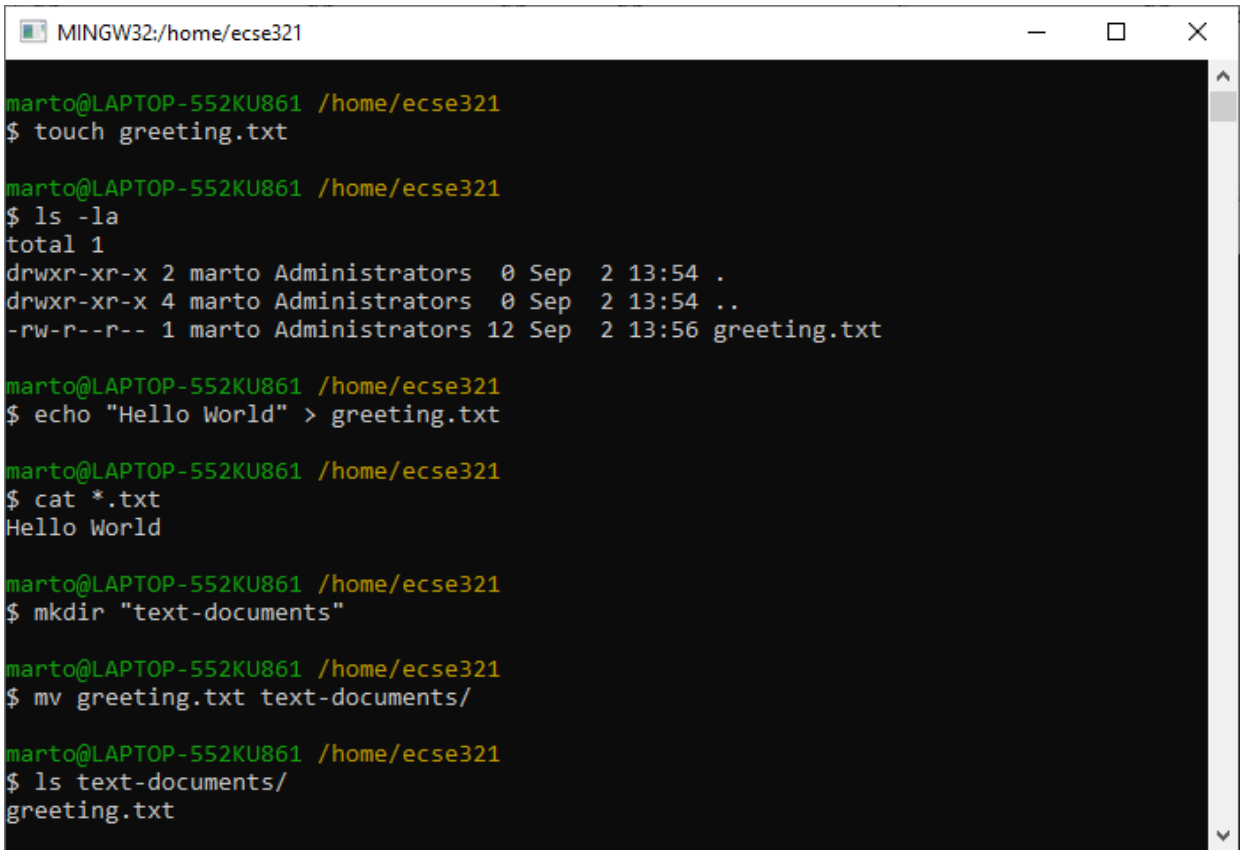
     ```
     $ ls /home
     ecse321 guest-user admin
     ```

- `cd`: navigates the file system
  Example:

```
$ cd ..
$ pwd
/home
$ cd ecse321
$ pwd
/home/ecse321
```

> **NOTE** The following steps will include images that illustrate the commands and their output to prevent easy copy-paste. Sorry! :)

2. Creating files and reading/writing their contents
   - `touch`: creates a file
   - `mkdir`: creates a directory
   - `mv`: moves a file (or directory) from its current location to a target location
   - `echo`: prints a string
   - `cat`: prints the contents of a file
     Example:

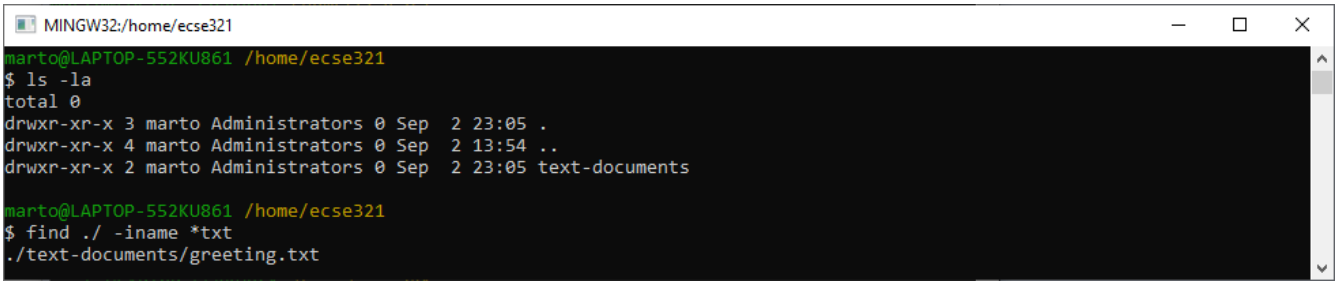


1.3.3. Finding files

The versatile `find` command allows us to find files based on given criteria. Take look at its manual

page with `man find`!

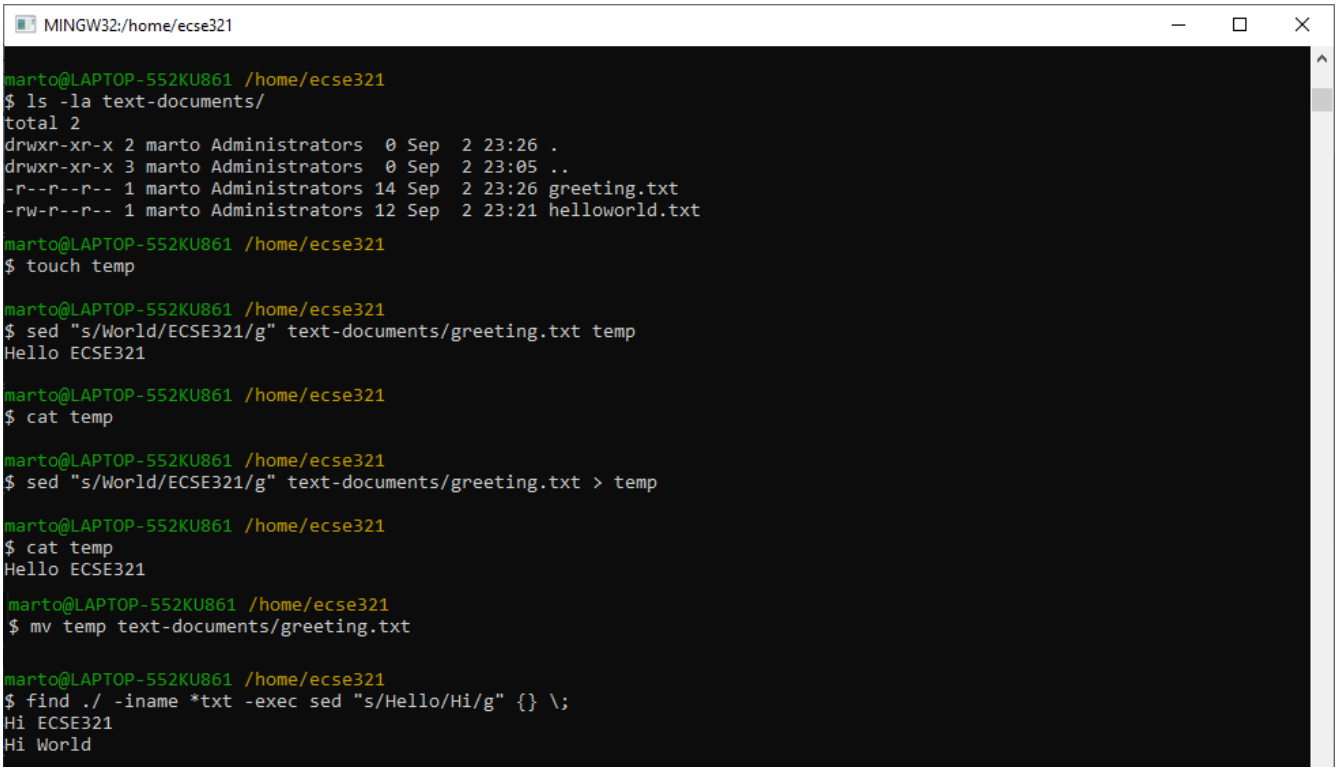
Example:



1.3.4. Batch file operations

- `sed`: stream editor; changes a given string to a replacement

Combining `find` with an additional command (e.g., `sed`) can greatly speed up your repetitive tasks. Example:



1.3.5. Some additional useful commands

- `rm`: removes a file
- `cp -r`: copies a directory recursively with its contents
- `rmdir`: remove an empty directory
- `rm -rf`: force to recursively delete a directory (or file) and all its contents
- `nano`: an easy-to-use text editor (not available by default in MinGW)
- `grep`: finds matches for a string in a given stream of characters
- `ag`: takes a string as argument and searches through the contents of files recursively to find

matches of the given string (this tool is included in the *silversearcher-ag* package)