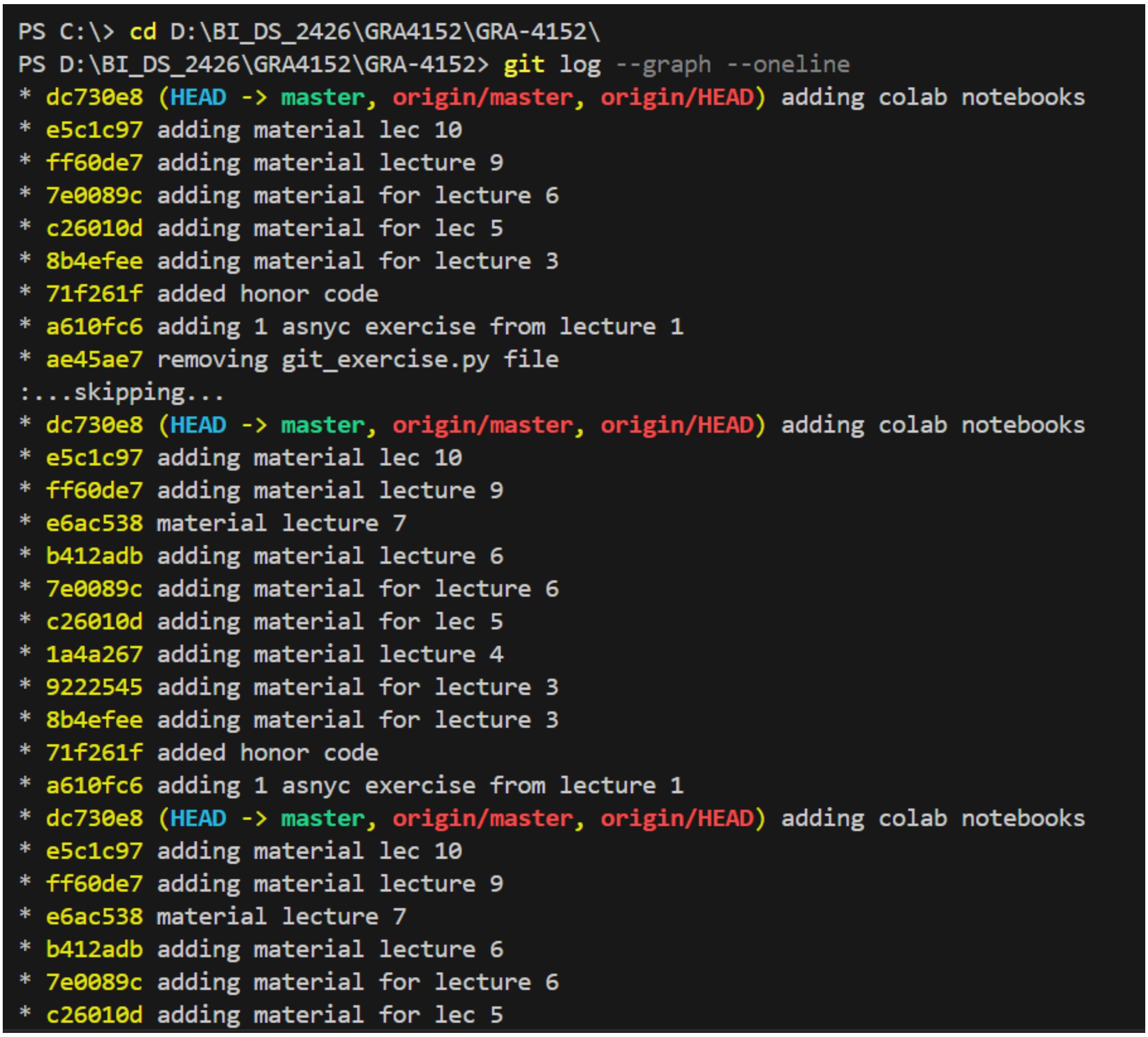
Git Exercises

1. Clone the repository for the class: https://github.com/BI-DS/GRA-4152

git clone https://github.com/BI-DS/GRA-4152.git

a. Explore the version history by visualizing it as a graph.

git log --graph --oneline

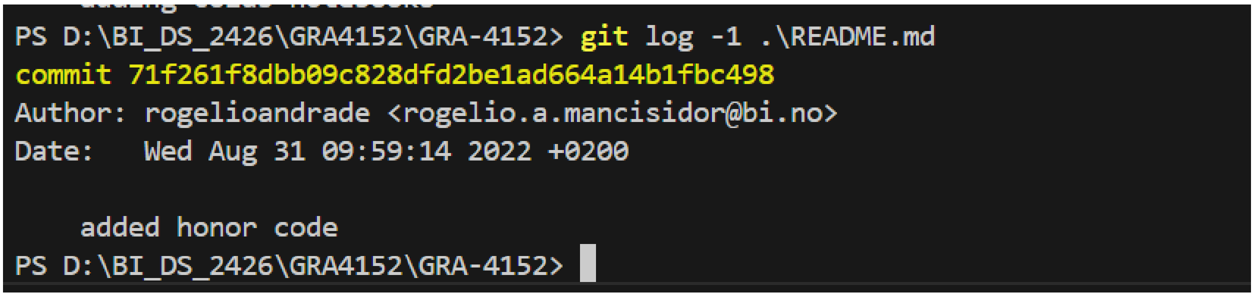


```
PS C:\> cd D:\BI_DS_2426\GRA4152\GRA-4152\
PS D:\BI_DS_2426\GRA4152\GRA-4152> git log --graph --oneline
* dc730e8 (HEAD -> master, origin/master, origin/HEAD) adding colab notebooks
* e5c1c97 adding material lec 10
* ff60de7 adding material lecture 9
* 7e0089c adding material for lecture 6
* c26010d adding material for lec 5
* 8b4efee adding material for lecture 3
* 71f261f added honor code
* a610fc6 adding 1 asnyc exercise from lecture 1
* ae45ae7 removing git_exercise.py file
:...skipping...
* dc730e8 (HEAD -> master, origin/master, origin/HEAD) adding colab notebooks
* e5c1c97 adding material lec 10
* ff60de7 adding material lecture 9
* e6ac538 material lecture 7
* b412adb adding material lecture 6
* 7e0089c adding material for lecture 6
* c26010d adding material for lec 5
* 1a4a267 adding material lecture 4
* 9222545 adding material for lecture 3
* 8b4efee adding material for lecture 3
* 71f261f added honor code
* a610fc6 adding 1 asnyc exercise from lecture 1
* dc730e8 (HEAD -> master, origin/master, origin/HEAD) adding colab notebooks
* e5c1c97 adding material lec 10
* ff60de7 adding material lecture 9
* e6ac538 material lecture 7
* b412adb adding material lecture 6
* 7e0089c adding material for lecture 6
* c26010d adding material for lec 5
```

b. When was the last time README.md was modified?

git log -1 README.md

The last modify was Wednesday 31/08/2022

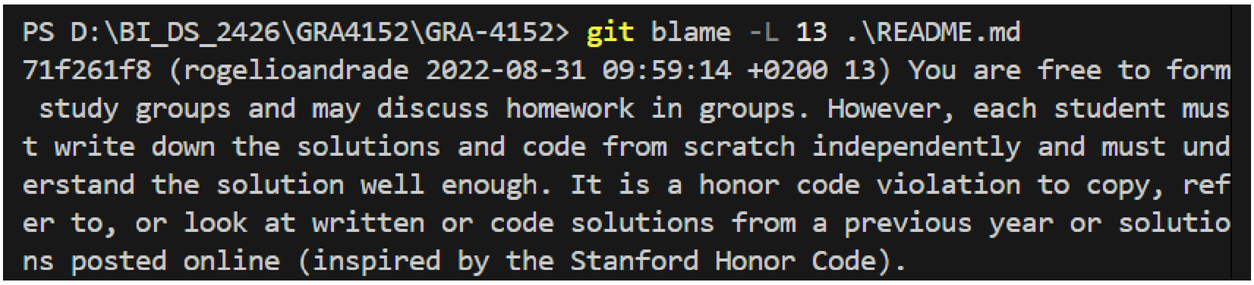


```
PS D:\BI_DS_2426\GRA4152\GRA-4152> git log -1 .\README.md
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code
PS D:\BI_DS_2426\GRA4152\GRA-4152>
```

c. What was the commit message associated with the last modification to the README.md? (Hint: use git blame and git show)

git blame -L 13 README.md

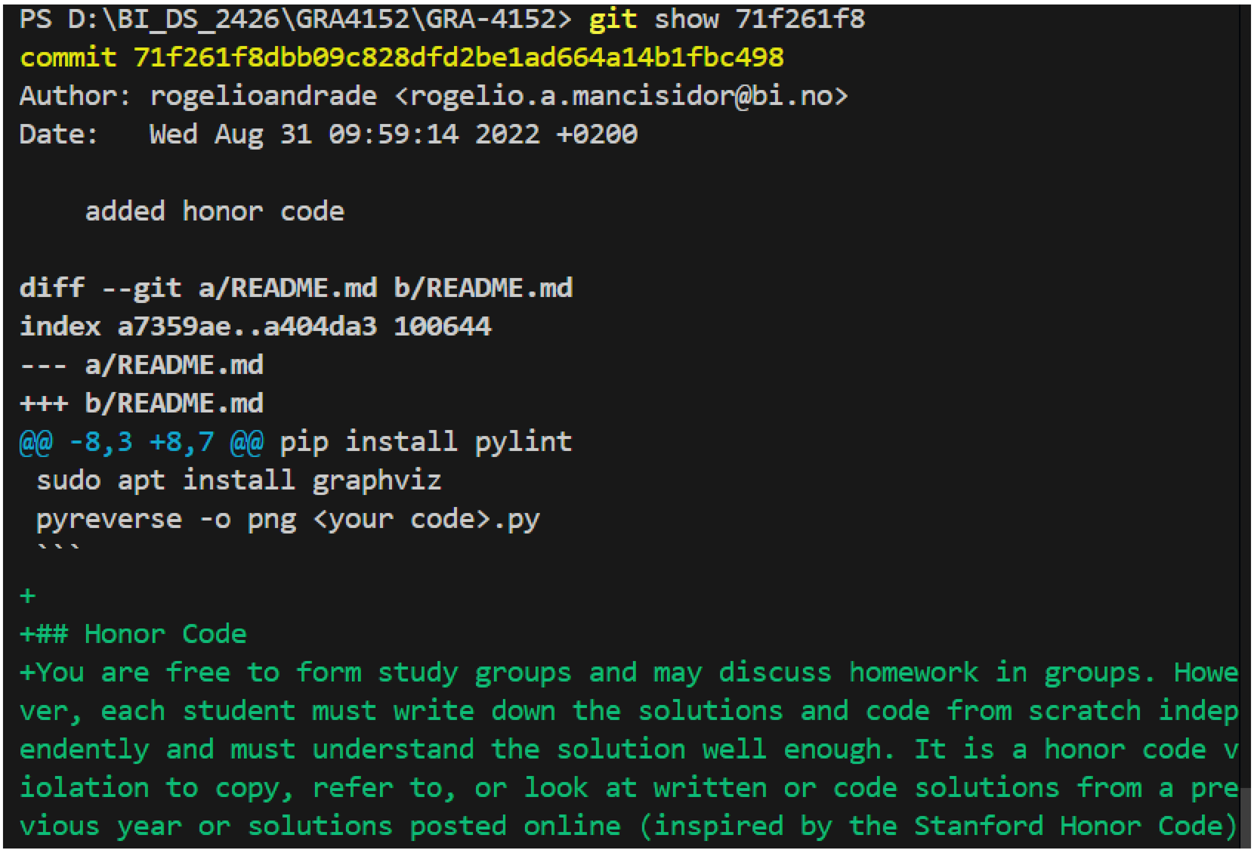
The message was below (I actually try git blame first then I saw there was 13 lines so I add -L 13 to get the last message)



```
PS D:\BI_DS_2426\GRA4152\GRA-4152> git blame -L 13 .\README.md
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 13) You are free to form
 study groups and may discuss homework in groups. However, each student mus
t write down the solutions and code from scratch independently and must und
erstand the solution well enough. It is a honor code violation to copy, ref
er to, or look at written or code solutions from a previous year or solutio
ns posted online (inspired by the Stanford Honor Code).
```

Another option is to use git show

git show 71f261f8

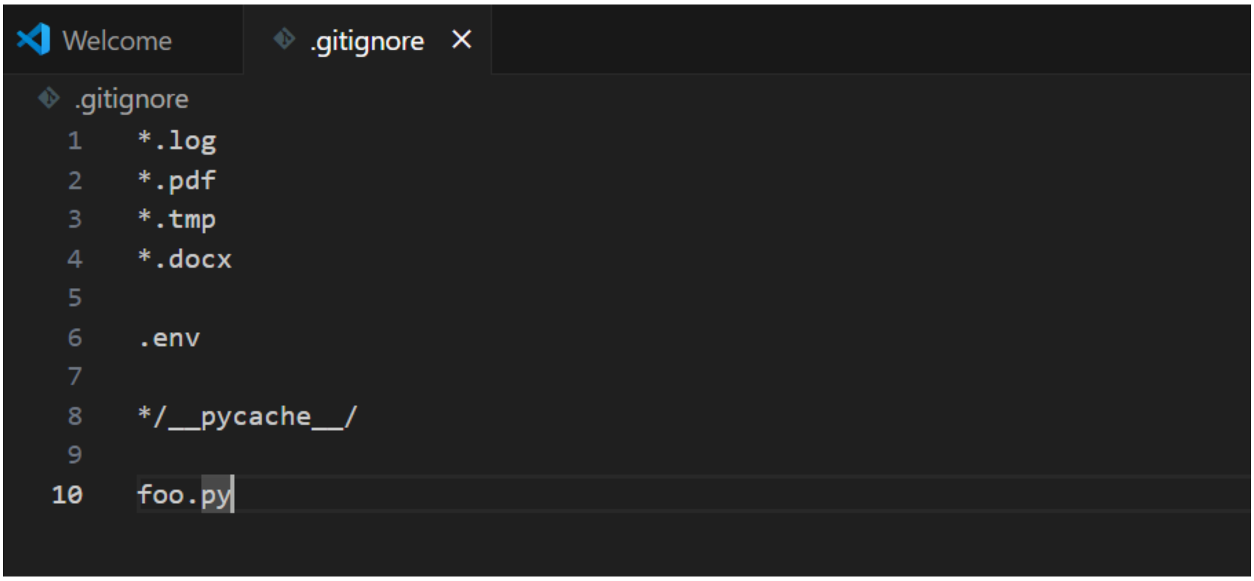


```
PS D:\BI_DS_2426\GRA4152\GRA-4152> git show 71f261f8
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code

diff --git a/README.md b/README.md
index a7359ae..a404da3 100644
--- a/README.md
+++ b/README.md
@@ -8,3 +8,7 @@ pip install pylint
 sudo apt install graphviz
 pyreverse -o png <your code>.py
 ```
+
+### Honor Code
+You are free to form study groups and may discuss homework in groups. Howe
ver, each student must write down the solutions and code from scratch indep
endently and must understand the solution well enough. It is a honor code v
iolation to copy, refer to, or look at written or code solutions from a pre
vious year or solutions posted online (inspired by the Stanford Honor Code)
```
```

2. One common mistake when learning Git is to commit large files that should not be managed by Git or adding sensitive information.

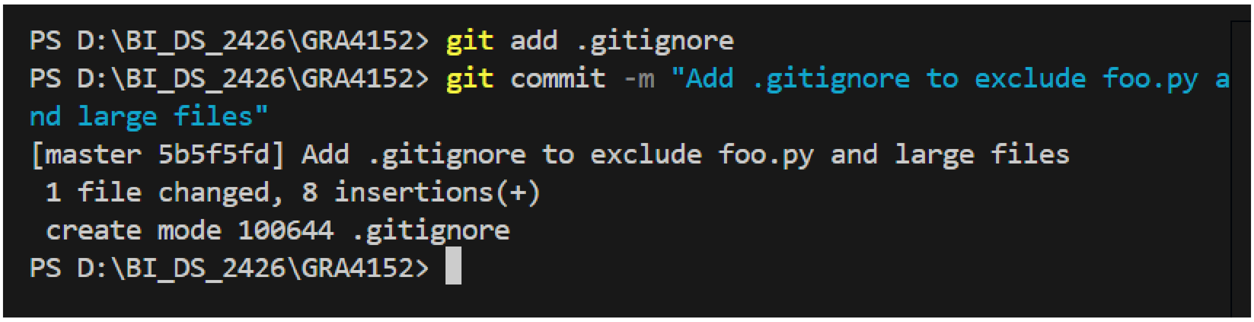
a. Add a .gitignore file to your portfolio code repository (https://github.com/SXXXXXXX/GRA4152) and exclude files and/or folders. You might need to create a foo.py file to be excluded.

Step 1: Create a .gitignore file and exclude files and folders in .gitignore



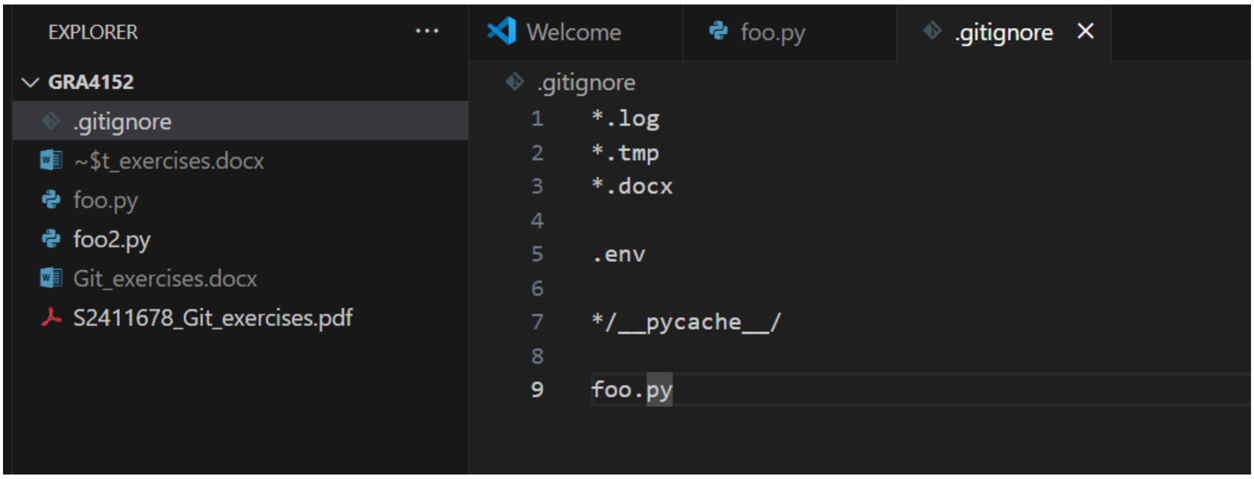
```
*.log
*.pdf
*.tmp
*.docx

.env

*/__pycache__/

foo.py
```

Step 2: Add and commit .gitignore

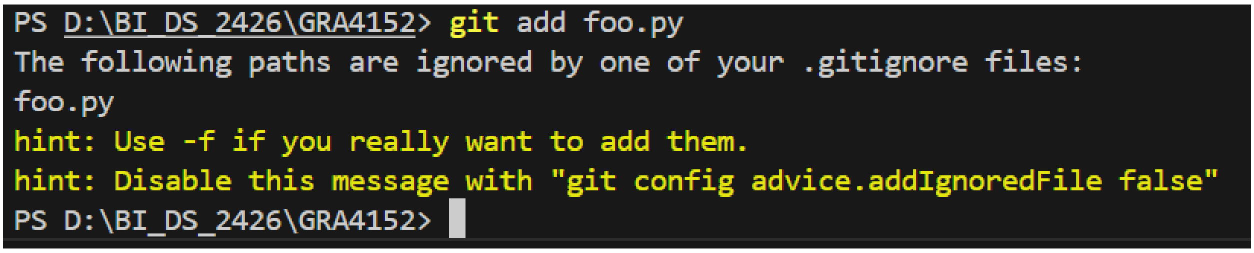


```
PS D:\BI_DS_2426\GRA4152> git add .gitignore
PS D:\BI_DS_2426\GRA4152> git commit -m "Add .gitignore to exclude foo.py a
nd large files"
[master 5b5f5fd] Add .gitignore to exclude foo.py and large files
 1 file changed, 8 insertions(+)
 create mode 100644 .gitignore
PS D:\BI_DS_2426\GRA4152>
```

The foo.py, .docx is not tracked (blur gray)



```
*.log
*.tmp
*.docx

.env

*/__pycache__/

foo.py
```



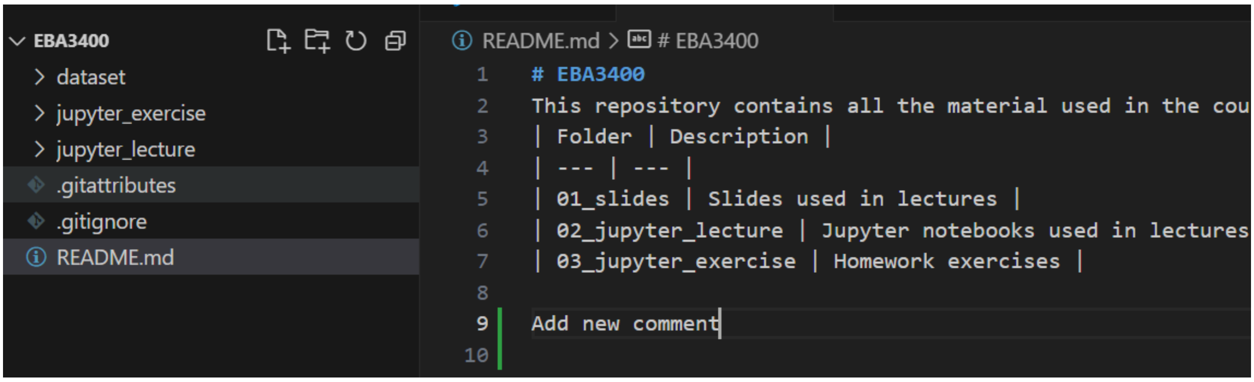
```
PS D:\BI_DS_2426\GRA4152> git add foo.py
The following paths are ignored by one of your .gitignore files:
foo.py
hint: Use -f if you really want to add them.
hint: Disable this message with "git config advice.addIgnoredFile false"
PS D:\BI_DS_2426\GRA4152>
```

As foo.py in .gitignore hence the file cannote be commit to github

3. Clone some repository from GitHub and modify one of its existing files

git clone https://github.com/BI-DS/EBA3400.git

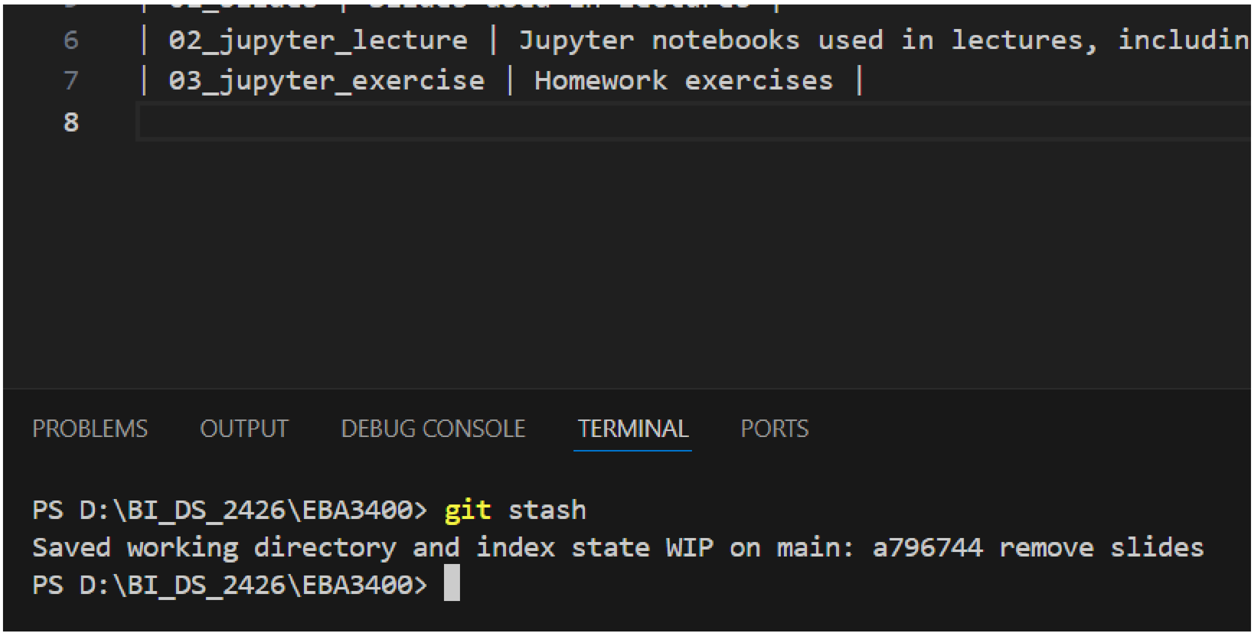
I added a new comment to READ.me



```
# EBA3400
This repository contains all the material used in the cou
| Folder | Description |
| --- | --- |
| 01_slides | Slides used in lectures |
| 02_jupyter_lecture | Jupyter notebooks used in lectures
| 03_jupyter_exercise | Homework exercises |

Add new comment

```

a. What happens when you type git stash?

The line I added was stashed (not shown) on README.md

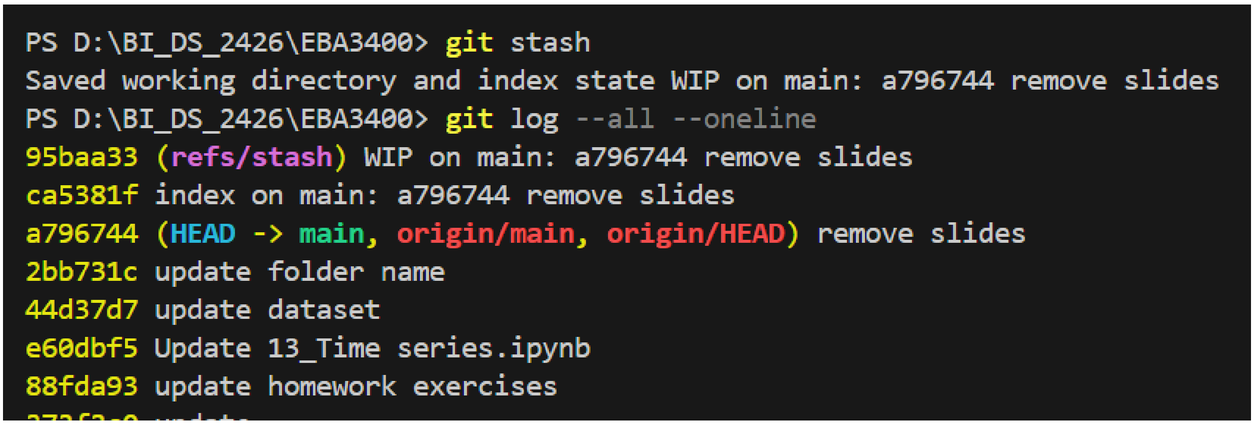


```
| 02_jupyter_lecture | Jupyter notebooks used in lectures, includin
| 03_jupyter_exercise | Homework exercises |

```

```
PS D:\BI_DS_2426\EBA3400> git stash
Saved working directory and index state WIP on main: a796744 remove slides
PS D:\BI_DS_2426\EBA3400>
```

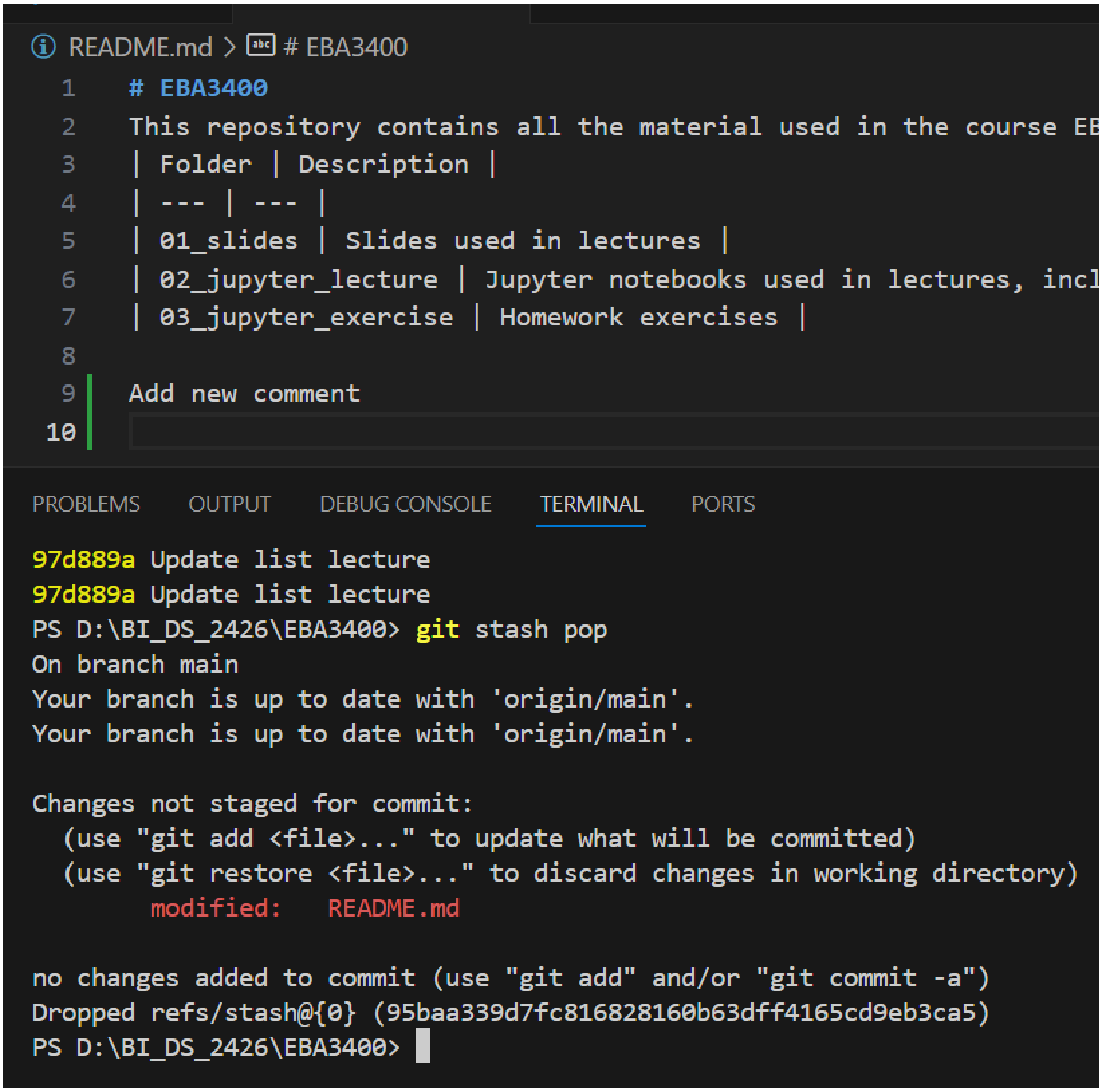
b. What do you see when running git log --all --oneline?

This stash is on main branch



```
PS D:\BI_DS_2426\EBA3400> git stash
Saved working directory and index state WIP on main: a796744 remove slides
PS D:\BI_DS_2426\EBA3400> git log --all --oneline
95baa33 (refs/stash) WIP on main: a796744 remove slides
ca5381f index on main: a796744 remove slides
a796744 (HEAD -> main, origin/main, origin/HEAD) remove slides
2bb731c update folder name
44d37d7 update dataset
e60dbf5 Update 13_Time series.ipynb
88fda93 update homework exercises
```

c. Run git stash pop to undo what you did with git stash. In what scenario might this be useful?



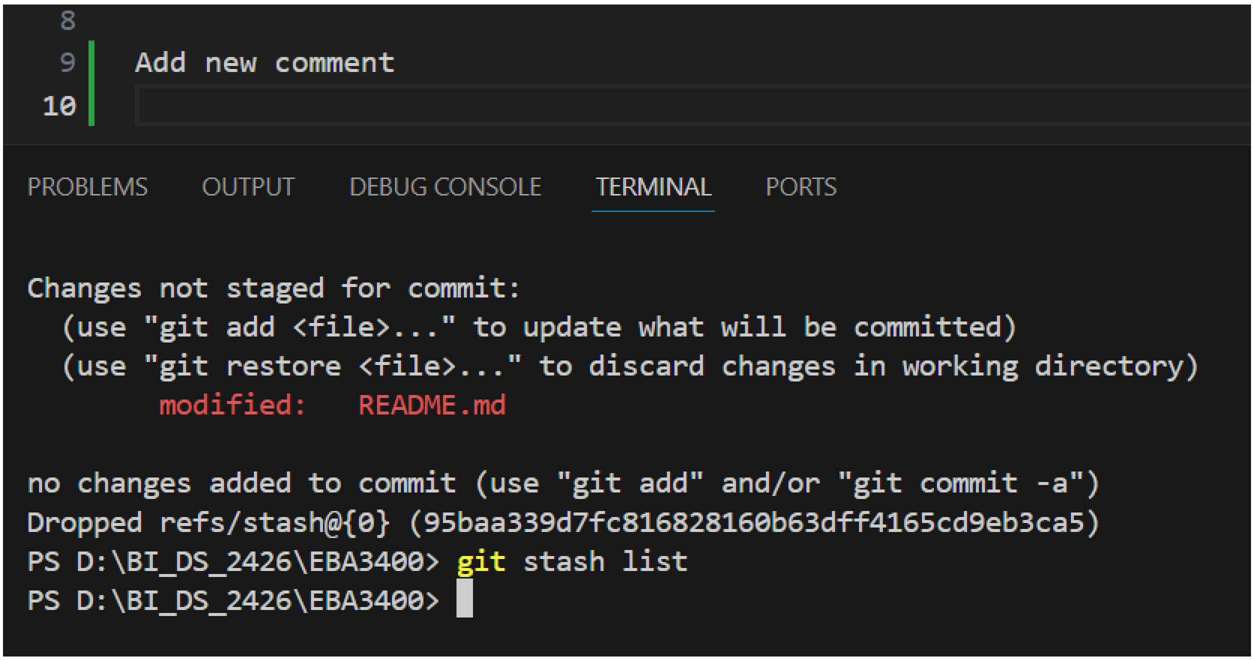
```
# EBA3400
This repository contains all the material used in the course EB
| Folder | Description |
| --- | --- |
| 01_slides | Slides used in lectures |
| 02_jupyter_lecture | Jupyter notebooks used in lectures, incl
| 03_jupyter_exercise | Homework exercises |

Add new comment

```

```
97d889a Update list lecture
97d889a Update list lecture
PS D:\BI_DS_2426\EBA3400> git stash pop
On branch main
Your branch is up to date with 'origin/main'.
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (95baa339d7fc816828160b63dff4165cd9eb3ca5)
PS D:\BI_DS_2426\EBA3400> 
```

This would be useful if I want to reapply changes to my working copy that I temporarely shelved to work on something else.

d. List your current stashes and delete them with git stash drop <stash_id>

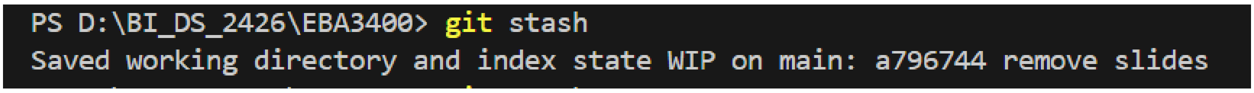
git stash list



```
Add new comment
```

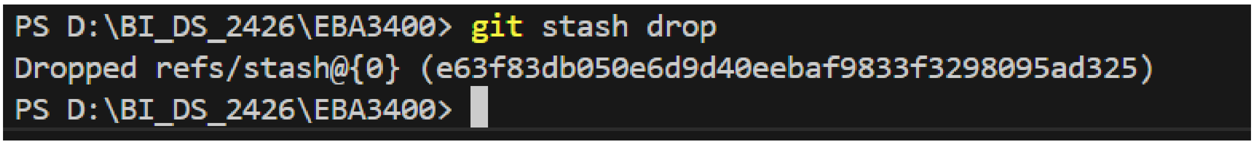
```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (95baa339d7fc816828160b63dff4165cd9eb3ca5)
PS D:\BI_DS_2426\EBA3400> git stash list
PS D:\BI_DS_2426\EBA3400>
```

There is no stash now so I need to make some changes and stash



```
PS D:\BI_DS_2426\EBA3400> git stash
Saved working directory and index state WIP on main: a796744 remove slides
```

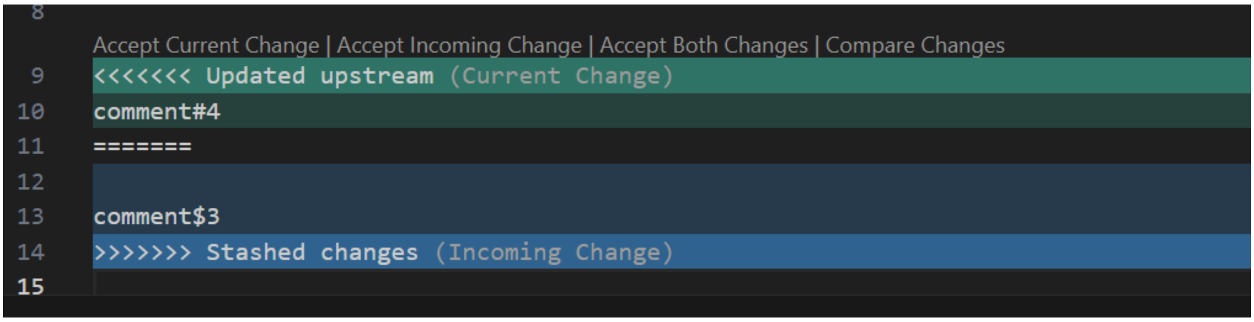
git stash drop



```
PS D:\BI_DS_2426\EBA3400> git stash drop
Dropped refs/stash@{0} (e63f83db050e6d9d40eebaf9833f3298095ad325)
PS D:\BI_DS_2426\EBA3400>
```

Now, modify a file and stash changes. Make a new modification to the same file, but this time commit those changes.

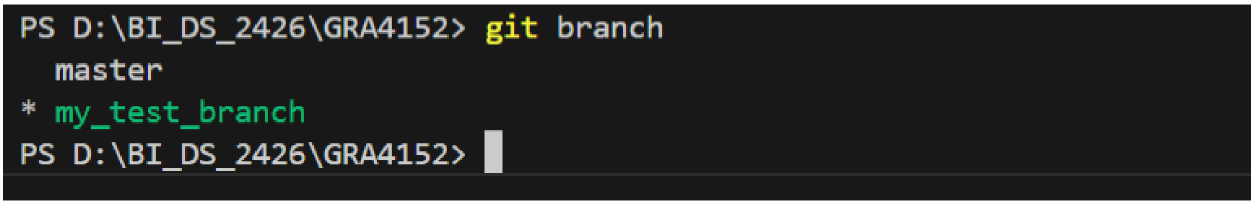
a. What happens if you type git stash pop and open the file that you have modified? What do you see in the file?


Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
<<<<<<< Updated upstream (Current Change)
comment#4
=======

comment$3
>>>>>>> Stashed changes (Incoming Change)
```

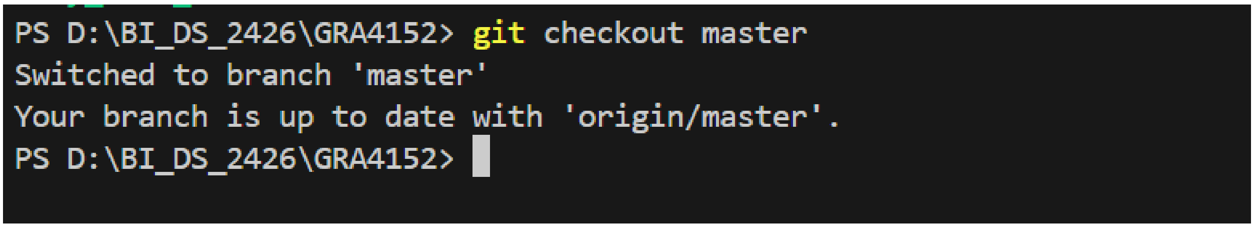
There is conflict that need to be resolved

4. Create a new branch in your class repository (https://github.com/SXXXXXXX/GRA4152) **and call it my_test_branch.**

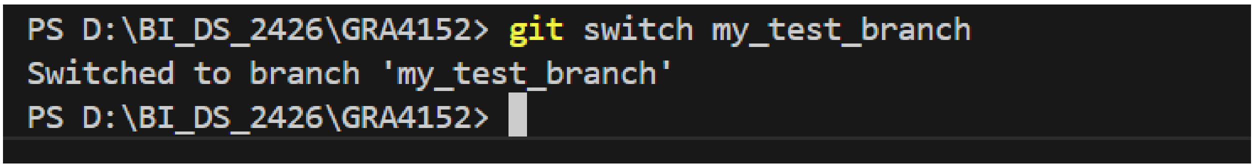


```
PS D:\BI_DS_2426\GRA4152> git branch
  master
* my_test_branch
PS D:\BI_DS_2426\GRA4152> 
```

a. Explore both branches, by switching back and forth.



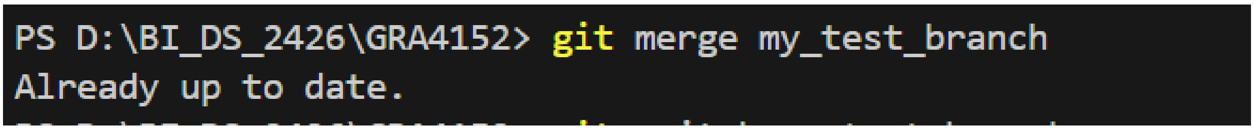
```
PS D:\BI_DS_2426\GRA4152> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
PS D:\BI_DS_2426\GRA4152> 
```



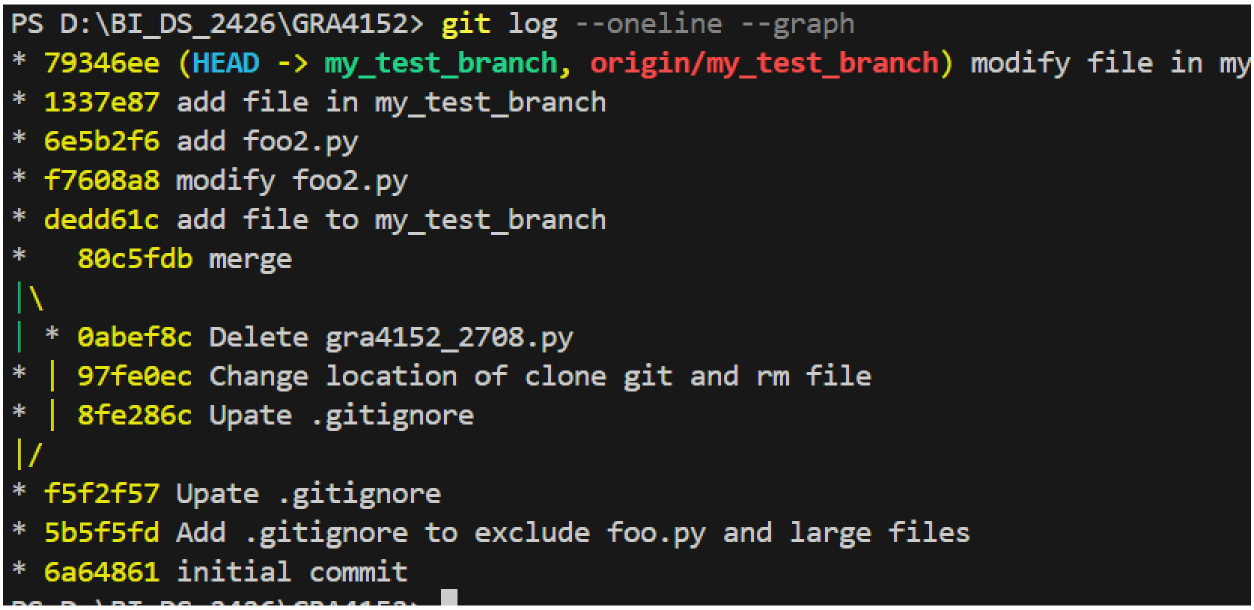
```
PS D:\BI_DS_2426\GRA4152> git switch my_test_branch
Switched to branch 'my_test_branch'
PS D:\BI_DS_2426\GRA4152> 
```

b. Add a comment or a line in any file in the branch my_test_branch. Then add such a change and commit.

c. Finally, merge my_test_branch into master



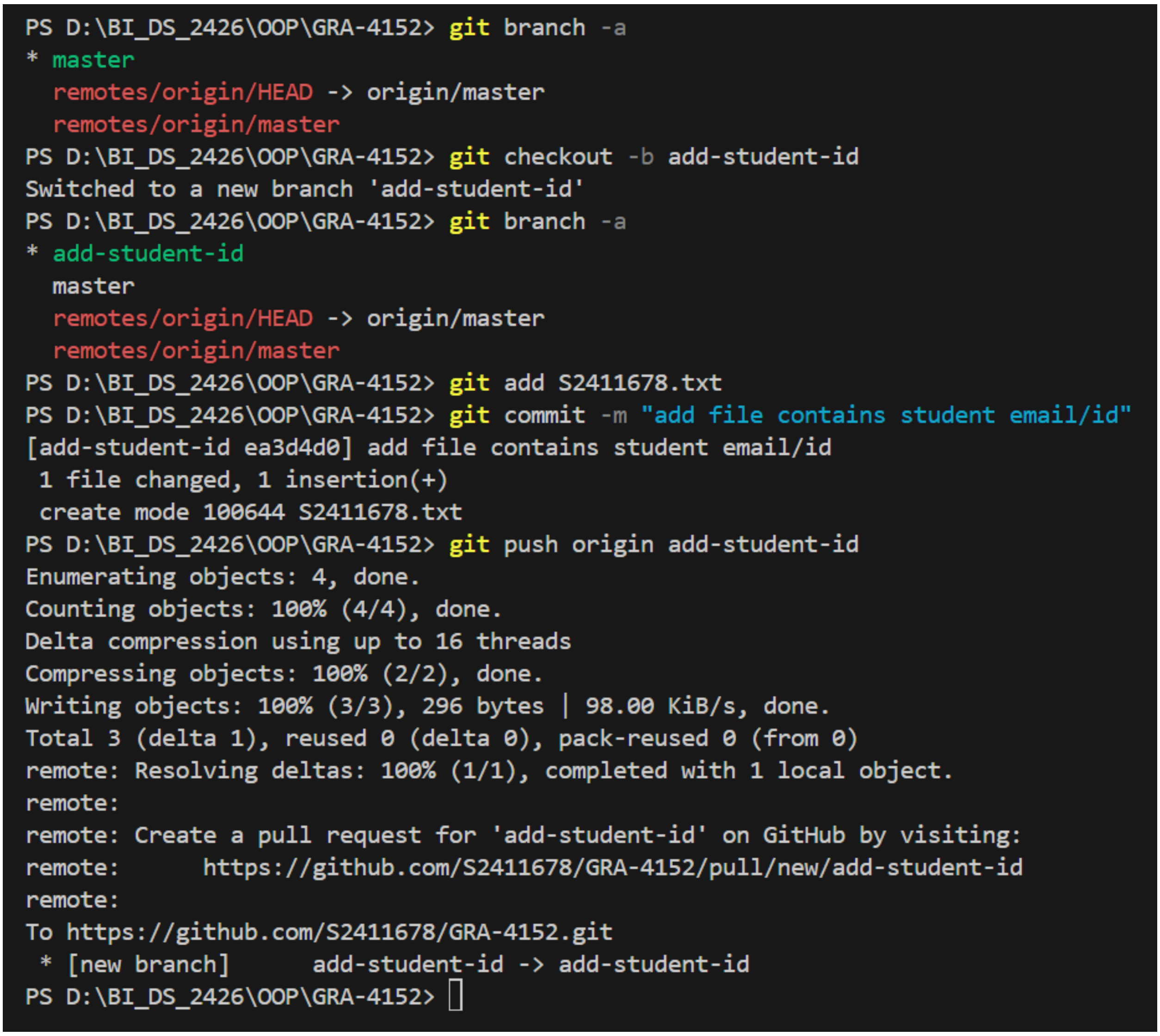
```
PS D:\BI_DS_2426\GRA4152> git merge my_test_branch
Already up to date.
```



```
PS D:\BI_DS_2426\GRA4152> git log --oneline --graph
* 79346ee (HEAD -> my_test_branch, origin/my_test_branch) modify file in my
* 1337e87 add file in my_test_branch
* 6e5b2f6 add foo2.py
* f7608a8 modify foo2.py
* dedd61c add file to my_test_branch
*   80c5fdb merge
|\
| * 0abef8c Delete gra4152_2708.py
* | 97fe0ec Change location of clone git and rm file
* | 8fe286c Upate .gitignore
|/
* f5f2f57 Upate .gitignore
* 5b5f5fd Add .gitignore to exclude foo.py and large files
* 6a64861 initial commit
```

5. Fork the class repository (https://github.com/BI-DS/GRA-4152)**. After you have a copy of the class repository, clone it to your local computer. In that way you can make local changes to the existing class repository**.

a. Send me a pull request to add a text file with your student id, i.e. SXXXXXXX.txt



```
PS D:\BI_DS_2426\OOP\GRA-4152> git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
PS D:\BI_DS_2426\OOP\GRA-4152> git checkout -b add-student-id
Switched to a new branch 'add-student-id'
PS D:\BI_DS_2426\OOP\GRA-4152> git branch -a
* add-student-id
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
PS D:\BI_DS_2426\OOP\GRA-4152> git add S2411678.txt
PS D:\BI_DS_2426\OOP\GRA-4152> git commit -m "add file contains student email/id"
[add-student-id ea3d4d0] add file contains student email/id
 1 file changed, 1 insertion(+)
 create mode 100644 S2411678.txt
PS D:\BI_DS_2426\OOP\GRA-4152> git push origin add-student-id
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 98.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'add-student-id' on GitHub by visiting:
remote:      https://github.com/S2411678/GRA-4152/pull/new/add-student-id
remote:
To https://github.com/S2411678/GRA-4152.git
 * [new branch]      add-student-id -> add-student-id
PS D:\BI_DS_2426\OOP\GRA-4152>
```