

## Git Exercises

Use a text editor, e.g., latex, word, libreoffice writer, etc, and write short and precise answers supported by inline or block code as well as screen shots when needed.  
Your final file must be in pdf format and must be delivered to your GitHub repository.

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

Use git clone, and the result is shown below:

```
See 'git help git' for an overview of the system.
(base) shanxu@Shans-MacBook-Pro ~ % git clone https://github.com/BI-DS/GRA-4152
Cloning into 'GRA-4152'...
[remote] Enumerating objects: 106, done.
[remote] Counting objects: 100% (58/58), done.
[remote] Compressing objects: 100% (48/48), done.
[remote] Total 106 (delta 20), reused 46 (delta 10), pack-reused 48 (from 1)
Receiving objects: 100% (106/106), 25.14 KiB | 1.09 MiB/s, done.
Resolving deltas: 100% (29/29), done.
(base) shanxu@Shans-MacBook-Pro ~ % cd GRA-4152
```

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

Use git log -graph

```
(base) shanxu@Shans-MacBook-Pro ~ % cd GRA-4152
(base) shanxu@Shans-MacBook-Pro GRA-4152 % git log --graph --oneline --all --decorate
* 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
* 0f6036b adding 1 asnyc exercise from lecture 1
* ae45ae7 removing git_exercise.py file
* 84ed53d adding file for git exercise in lecture 2
* 37ac00f adding some py files for lecture 1
* 0fb7842 adding instructions for UML
* 20b8851 initial commit
```

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

Use git log -1, and the last modify time is Aug 31 2022

```
((base) shanxu@Shans-MacBook-Pro 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
```

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

```
[base] shanxu@Shans-MacBook-Pro GRA-4152 % git blame README.md
^2b8851 (rogelioandrade 2022-08-19 08:48:04 +0200 1) # GRA-4152
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 2) This repository contains different materials used
t the course, e.g. examples shown in lectures, suggested solutions for homework, problems discussed in tut
sions, etc. You should follow this repository frequently, as different materials will become available as
th
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 3)
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 4) ## Packages
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 5) Unified Modeling Language (UML) is a tool to visua
design, or architecture, of (complex) software systems. Just like classes in OOP. We can generate UML dia
`Python` classes using the library `pylint`, which uses `graphviz` to generate `png` or `pdf` files show
rchitecture of a given class.
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 6) ```bash
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 7) pip install pylint
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 8) sudo apt install graphviz
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 9) pyreverse -o png <your code>.py
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 10)
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 11)
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 12) ## Honor Code
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 13) You are free to form study groups and may discuss
in groups. However, each student must write down the solutions and code from scratch independently and mu
tand the solution well enough. It is a honor code violation to copy, refer to, or look at written or code
s from a previous year or solutions posted online (inspired by the Stanford Honor Code).
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 14)
```

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git show
commit dc730e88e9e4ad1cf94c0511b50d4d3135eb7383 (HEAD -> master, origin/master, origin/HEAD)
Author: A1910329 <rogelio.mancisidor@gmail.com>
Date:   Wed Oct 26 12:48:02 2022 +0200

    adding colab notebooks

diff --git a/Lecture-10/parallel_computing.ipynb b/Lecture-10/parallel_computing.ipynb
new file mode 100644
index 000000..bd37a23
--- /dev/null
+++ b/Lecture-10/parallel_computing.ipynb
@@ -0,0 +1 @@

```

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.

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % touch .gitignore
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % pwd
/Users/shanxu/GRA-4152
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % vim .gitignore
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % touch foo.py
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % vim.gitignore
zsh: command not found: vim.gitignore
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % cd .gitignore
cd: not a directory: .gitignore
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % pwd
/Users/shanxu/GRA-4152
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % ls -al
total 16
drwxr-xr-x 16 shanxu staff 512 Sep 9 15:20 .
drwxr-xr-x+ 45 shanxu staff 1440 Sep 9 15:20 ..
drwxr-xr-x 12 shanxu staff 384 Sep 9 14:32 .git
-rw-r--r-- 1 shanxu staff 240 Sep 9 15:18 .gitignore
drwxr-xr-x 14 shanxu staff 448 Sep 9 15:18 GRA-4152
drwxr-xr-x 5 shanxu staff 160 Sep 9 14:32 Lecture-1
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-10
drwxr-xr-x 4 shanxu staff 128 Sep 9 14:32 Lecture-2
drwxr-xr-x 6 shanxu staff 192 Sep 9 14:32 Lecture-4
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-5
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-6
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-7
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-9
drwxr-xr-x 6 shanxu staff 192 Sep 9 14:32 Lecture_3
-rw-r--r-- 1 shanxu staff 1114 Sep 9 14:32 README.md
-rw-r--r-- 1 shanxu staff 0 Sep 9 15:20 foo.py
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % vim .gitignore
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

---

```
Book/
*.pdf
uml_diagrams
Lecture-1/*.pdf
Others/
*/__pycache__
*.log
Project
Lecture-9/mlp.py
Lecture-9/approx_pi.py
Lecture-10/average_solution.py
Lecture-10/get_ratios_solution.py
Lecture-10/list_solution.py
Lecture-10/primes_solution.py
foo.py
```

~

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

1. What happens when you type git stash?

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git stash
Saved working directory and index state WIP on master: dc730e8 adding colab note
books
(base) shanxu@Shans-MacBook-Pro GRA-4152 % ]
```

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

Show all the log in a pretty simple page

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git log --all --oneline
acffcbe (refs/stash) WIP on master: dc730e8 adding colab notebooks
36f06e9 index on master: dc730e8 adding colab notebooks
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
0f6036b adding 1 asnyc exercise from lecture 1
ae45ae7 removing git_exercise.py file
84ed53d adding file for git exercise in lecture 2
37ac00f adding some py files for lecture 1
0fb7842 adding instructions for UML
20b8851 initial commit
(base) shanxu@Shans-MacBook-Pro GRA-4152 % ]
```

3. Run git stash pop to undo what you did with git stash. In what

scenario might this be useful?

It is useful when you want to save some work temporally , and keep the directory or stage clean. For example, when I do some code workload and I do not want to commit now, so I can use git stash to save this change and do not affect directories.

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git stash pop
On branch master
Your branch is up to date with 'origin/master'.

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:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GRA-4152/

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (acffcbe7a9ac529ce3069de1b4418d11d82d5da4)
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

#### 4. List your current stashes and delete them with git stash drop

```
<stash_id>
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git stash list
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

Type git stash list and show nothing, because I used git slash pop. Git slash pop leads the change is deleted from the git list, thus git list shows nothing.

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

```
(base) shanxu@Shans-MacBook-Pro GRA-4152 % ls -al
total 16
drwxr-xr-x 16 shanxu staff 512 Sep 9 18:46 .
drwxr-xr-x+ 45 shanxu staff 1440 Sep 9 15:26 ..
drwxr-xr-x 13 shanxu staff 416 Sep 9 18:46 .git
-rw-r--r-- 1 shanxu staff 248 Sep 9 18:46 .gitignore
drwxr-xr-x 14 shanxu staff 448 Sep 9 15:18 GRA-4152
drwxr-xr-x 5 shanxu staff 160 Sep 9 14:32 Lecture-1
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-10
drwxr-xr-x 4 shanxu staff 128 Sep 9 14:32 Lecture-2
drwxr-xr-x 6 shanxu staff 192 Sep 9 14:32 Lecture-4
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-5
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-6
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-7
drwxr-xr-x 7 shanxu staff 224 Sep 9 14:32 Lecture-9
drwxr-xr-x 6 shanxu staff 192 Sep 9 14:32 Lecture_3
-rw-r--r-- 1 shanxu staff 1114 Sep 9 14:32 README.md
-rw-r--r-- 1 shanxu staff 0 Sep 9 15:20 foo.py
(base) shanxu@Shans-MacBook-Pro GRA-4152 % vim .gitignore
```

```
(base) shanxu@Shans-MacBook-Pro GRA-4152 % git stash
Saved working directory and index state WIP on master: dc730e8 adding colab notebooks
(base) shanxu@Shans-MacBook-Pro GRA-4152 % git stash list
stash@{0}: WIP on master: dc730e8 adding colab notebooks
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

Firstly, I want to see all the files in the directory of GRA4152, and decide to add a note in .gitignore file. Later on, I use git stash to save this change and use git stash list to show the change.

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

```
(base) shanxu@Shans-MacBook-Pro GRA-4152 % git stash pop
On branch master
Your branch is up to date with 'origin/master'.

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:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GRA-4152/

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (5603685f3b68c1861ababcb2be588db400a720ce)
(base) shanxu@Shans-MacBook-Pro GRA-4152 % vim .gitignore
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

After I type the git stash pop, and open the file I changed before. I find the file is not changed.

The reason for this phenomenon is that nobody modify .gitignore file. No contradiction happens.

5. Create a new branch in your class repository (<https://github.com/SXXXXXXX/GRA4152>) and call it my\_test\_branch.

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % pwd
/Users/shanxu/GRA-4152
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git checkout -b my_test_branch
fatal: A branch named 'my_test_branch' already exists.
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git push origin my_test_branch
Username for 'https://github.com': s2411256
>Password for 'https://s2411256@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/BI-DS/GRA-4152/'
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % pwd
/Users/shanxu/GRA-4152
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git push origin my_test_branch
Username for 'https://github.com': s2411256
>Password for 'https://s2411256@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/BI-DS/GRA-4152/'
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

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

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git branch
  master
* my_test_branch
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git checkout master
M   .gitignore
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git branch
* master
  my_test_branch
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git checkout my_test_branch
M   .gitignore
Switched to branch 'my_test_branch'
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git branch
  master
* my_test_branch
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```

b. Add a comment or a line in any file in the branch my\_test\_branch. Then add such a change and commit.

```

(base) shanxu@Shans-MacBook-Pro GRA-4152 % git branch
  master
* my_test_branch
(base) shanxu@Shans-MacBook-Pro GRA-4152 % touch git_exercise_solutions.txt
(base) shanxu@Shans-MacBook-Pro GRA-4152 % ls -al
total 16
drwxr-xr-x  17 shanxu  staff   544 Sep  9 20:00 .
drwxr-xr-x+ 45 shanxu  staff  1440 Sep  9 19:13 ..
drwxr-xr-x  13 shanxu  staff   416 Sep  9 19:53 .git
-rw-r--r--  1 shanxu  staff   273 Sep  9 19:12 .gitignore
drwxr-xr-x  14 shanxu  staff   448 Sep  9 15:18 GRA-4152
drwxr-xr-x  5 shanxu  staff   160 Sep  9 14:32 Lecture-1
drwxr-xr-x  7 shanxu  staff   224 Sep  9 14:32 Lecture-10
drwxr-xr-x  4 shanxu  staff   128 Sep  9 14:32 Lecture-2
drwxr-xr-x  6 shanxu  staff   192 Sep  9 14:32 Lecture-4
drwxr-xr-x  7 shanxu  staff   224 Sep  9 14:32 Lecture-5
drwxr-xr-x  7 shanxu  staff   224 Sep  9 14:32 Lecture-6
drwxr-xr-x  7 shanxu  staff   224 Sep  9 14:32 Lecture-7
drwxr-xr-x  7 shanxu  staff   224 Sep  9 14:32 Lecture-9
drwxr-xr-x  6 shanxu  staff   192 Sep  9 14:32 Lecture_3
-rw-r--r--  1 shanxu  staff   1114 Sep  9 14:32 README.md
-rw-r--r--  1 shanxu  staff     0 Sep  9 15:20 foo.py
-rw-r--r--  1 shanxu  staff     0 Sep  9 20:00 git_exercise_solutions.txt
(base) shanxu@Shans-MacBook-Pro GRA-4152 % git add git_exercise_solutions.txt

```

c. Finally, merge my\_test\_branch into master.

```

(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % git branch
  master
* my_test_branch
(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % ls -al
total 0
drwxr-xr-x  3 shanxu  staff    96 Sep  9 20:16 .
drwxr-xr-x  18 shanxu  staff   576 Sep  9 20:14 ..
drwxr-xr-x  14 shanxu  staff   448 Sep  9 20:16 GRA-4152
(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % git checkout master
M      .gitignore
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % git merge my_test_branch
Updating dc730e8..5d2fc01
Fast-forward
 git_exercise_solutions.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 git_exercise_solutions.txt
(base) shanxu@Shans-MacBook-Pro GRA-4152-copy %

```

6. 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.

```

[(base) shanxu@Shans-MacBook-Pro GRA-4152 % pwd
/Users/shanxu/GRA-4152
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % mkdir GRA-4152-copy
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % cd GRA-4152-copy
[(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % pwd
/Users/shanxu/GRA-4152/GRA-4152-copy
(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % git clone https://github.com/s2411256/GRA-4152.git

Cloning into 'GRA-4152'...
remote: Enumerating objects: 106, done.
remote: Counting objects: 100% (58/58), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 106 (delta 20), reused 46 (delta 10), pack-reused 48 (from 1)
Receiving objects: 100% (106/106), 25.14 KiB | 3.59 MiB/s, done.
Resolving deltas: 100% (29/29), done.
[(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % git branch
  master
* my_test_branch
[(base) shanxu@Shans-MacBook-Pro GRA-4152-copy % ls -al
total 0
drwxr-xr-x  3 shanxu  staff   96 Sep  9 20:16 .
drwxr-xr-x  18 shanxu  staff  576 Sep  9 20:14 ..
drwxr-xr-x  14 shanxu  staff  448 Sep  9 20:16 GRA-4152
(base) shanxu@Shans-MacBook-Pro GRA-4152-copy %

```

ChatGPT can make mistakes. Check important info.

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

```

[(base) shanxu@Shans-MacBook-Pro GRA-4152 % touch s2411256.txt
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % ls -al
total 16
drwxr-xr-x  15 shanxu  staff  480 Sep  9 20:59 .
drwxr-xr-x   3 shanxu  staff   96 Sep  9 20:16 ..
drwxr-xr-x  12 shanxu  staff  384 Sep  9 20:57 .git
-rw-r--r--   1 shanxu  staff  240 Sep  9 20:16 .gitignore
drwxr-xr-x   5 shanxu  staff  160 Sep  9 20:16 Lecture-1
drwxr-xr-x   7 shanxu  staff  224 Sep  9 20:16 Lecture-10
drwxr-xr-x   4 shanxu  staff  128 Sep  9 20:16 Lecture-2
drwxr-xr-x   6 shanxu  staff  192 Sep  9 20:16 Lecture-4
drwxr-xr-x   7 shanxu  staff  224 Sep  9 20:16 Lecture-5
drwxr-xr-x   7 shanxu  staff  224 Sep  9 20:16 Lecture-6
drwxr-xr-x   7 shanxu  staff  224 Sep  9 20:16 Lecture-7
drwxr-xr-x   7 shanxu  staff  224 Sep  9 20:16 Lecture-9
drwxr-xr-x   6 shanxu  staff  192 Sep  9 20:16 Lecture_3
-rw-r--r--   1 shanxu  staff  1114 Sep  9 20:16 README.md
-rw-r--r--   1 shanxu  staff     0 Sep  9 20:59 s2411256.txt
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git add s2411256.txt
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git commit -m"student id is 2411256"
[student-id 9f57787] student id is 2411256
  Committer: Shan Xu <shanxu@Shans-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 s2411256.txt
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git branch -
  master
* student-id
  remotes/origin/HEAD -> origin/master
  remotes/origin/master

```

```
[(base) shanxu@Shans-MacBook-Pro GRA-4152 % git push origin student-id
Username for 'https://github.com': s2411256
[Password for 'https://s2411256@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'student-id' on GitHub by visiting:
remote:     https://github.com/s2411256/GRA-4152/pull/new/student-id
remote:
To https://github.com/s2411256/GRA-4152.git
 * [new branch]      student-id -> student-id
(base) shanxu@Shans-MacBook-Pro GRA-4152 %
```