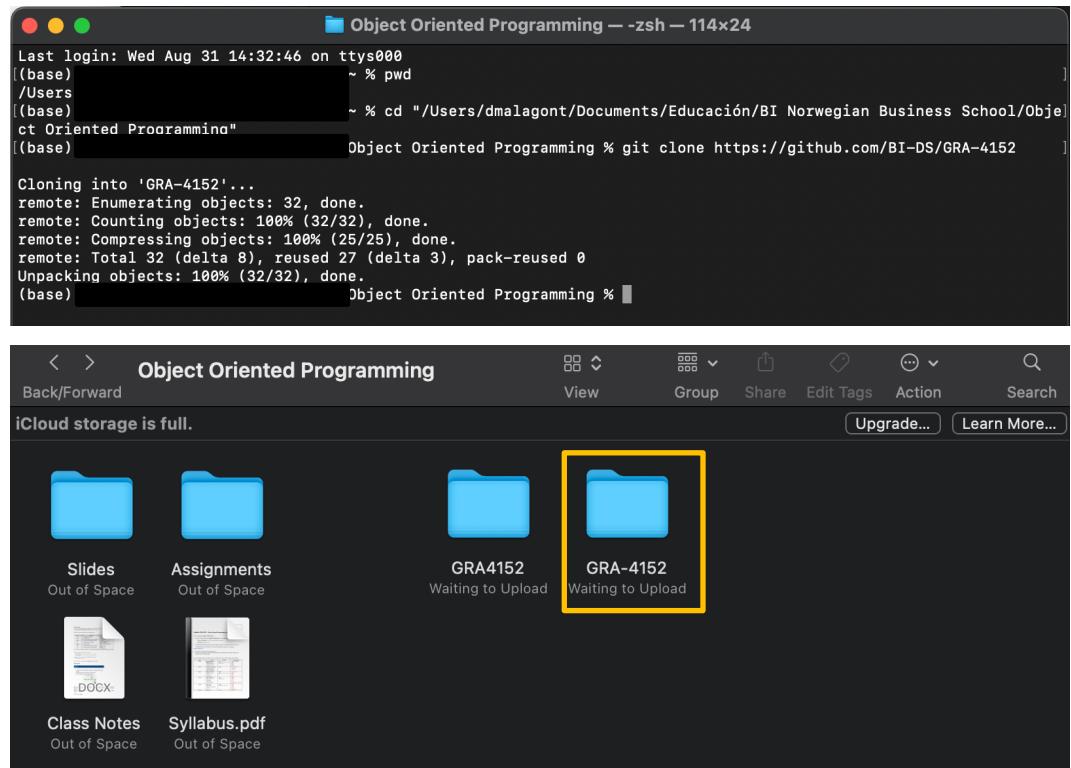


# GIT ASSIGNMENT

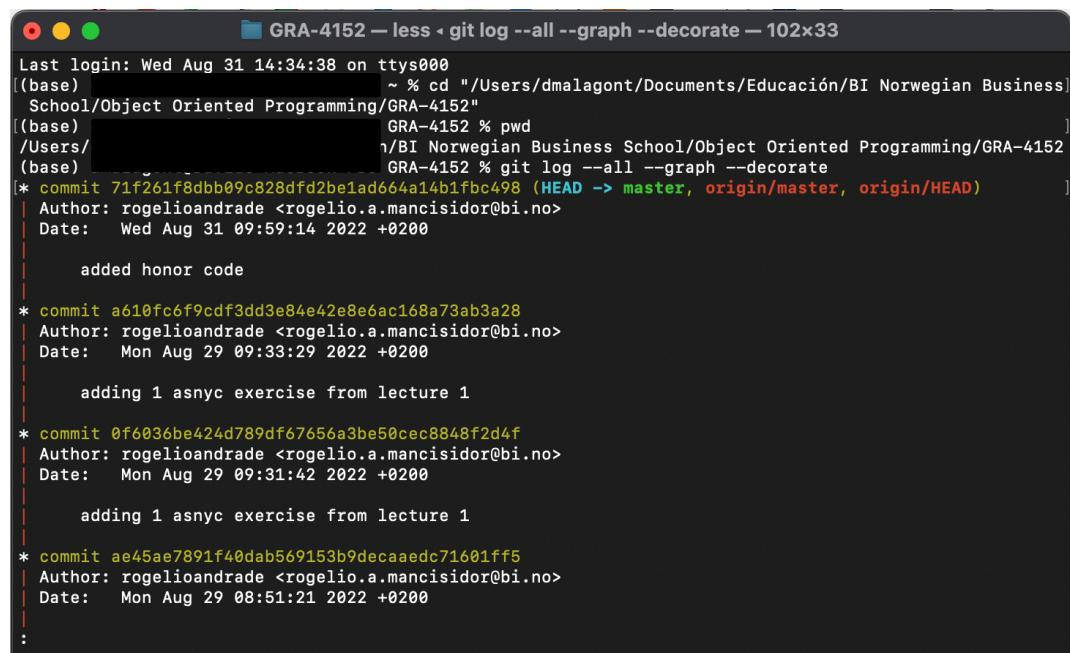
## 1. Clone the repository for the class: <https://github.com/BI-DS/GRA-4152>

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



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

```
git log --all --graph --decorate
```



- b) When was the last time README.md was modified? (Hint: use git log with an argument).**

```
git log --all --graph --decorate README.md
```

```
Last login: Wed Aug 31 14:47:42 on ttys000
(base) ~ % cd "/Users/dmalagont/Documents/Educación/BI Norwegian Business School/Object Oriented Programming/GRA-4152"
(base) GRA-4152 % git log --all --graph --decorate README.md
* commit 71f261f8dbb09c828fd2be1ad664a14b1fbc498 (HEAD -> master, origin/master, origin/HEAD)
| Author: rogelioandrade <rogerio.a.mancisidor@bi.no>
| Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code

* commit 0fb7842d8311144c4d1941b8e9d828059e11c500
| Author: rogelioandrade <rogerio.a.mancisidor@bi.no>
| Date:   Mon Aug 22 08:50:07 2022 +0200

    adding instructions for UML

* commit 20b88515dc1668bed7942359e3ad183f51961f62
| Author: rogelioandrade <rogerio.a.mancisidor@bi.no>
| Date:   Fri Aug 19 08:48:04 2022 +0200

    initial commit
```

The last time the README.md file was modified was on Wednesday August 31 at 09:59:14 when “rogerioandrade” added the honour code.

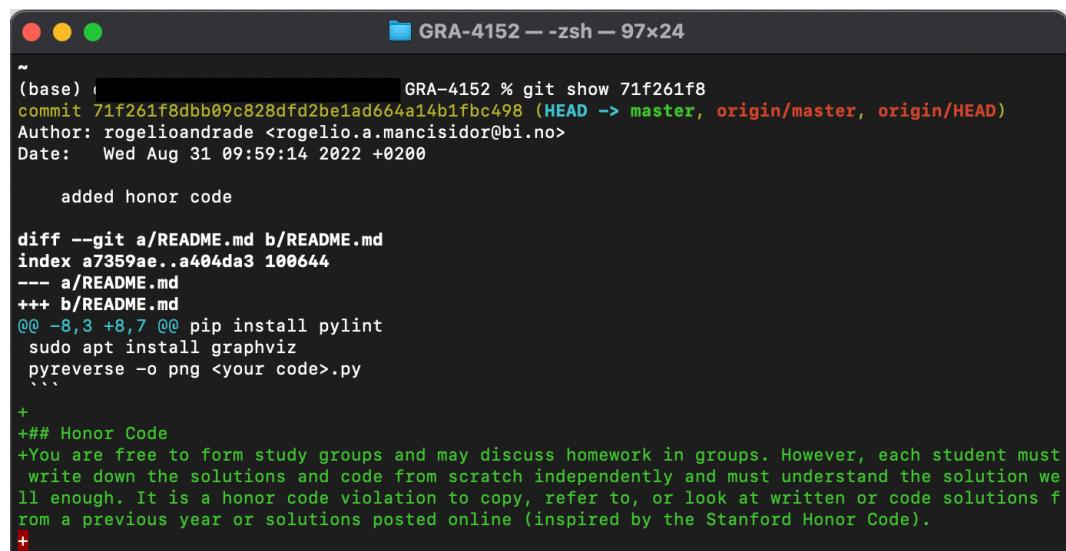
- c) What was the commit message associated with the last modification to the README.md? (Hint: use git blame and git show).**

```
git blame README.md
```

```
^20b8851 (rogerioandrade 2022-08-19 08:48:04 +0200 1) # GRA-4152
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 2) This repository contains different materials used throughout the course, e.g. examples shown in lectures, suggested solutions for homework, problems discussed in tutorial sessions, etc. You should follow this repository frequently, as different materials will become available as we cover them.
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 3)
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 4) ## Packages
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 5) Unified Modeling Language (UML) is a tool to visualize the design, or architecture, of (complex) software systems. Just like classes in OOP. We can generate UML diagrams for 'Python' classes using the library 'pylint', which uses 'graphviz' to generate 'png' or 'pdf' files showing the architecture of a given class.
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 6) ``bash
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 7) pip install pylint
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 8) sudo apt install graphviz
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 9) pyreverse -o png <your code>.py
0fb7842d (rogerioandrade 2022-08-22 08:50:07 +0200 10)
71f261f8 (rogerioandrade 2022-08-31 09:59:14 +0200 11)
71f261f8 (rogerioandrade 2022-08-31 09:59:14 +0200 12) ## Honor Code
71f261f8 (rogerioandrade 2022-08-31 09:59:14 +0200 13) You are free to form study groups and may discuss homework in groups. However, each student must write down the solutions and code from scratch independently and must understand the solution well enough. It is a honor code violation to copy, refer to, or look at written or code solutions from a previous year or solutions posted online (inspired by the Stanford Honor Code).
71f261f8 (rogerioandrade 2022-08-31 09:59:14 +0200 14)
~
```

It is possible to observe that the last hash is 71f261f8. Using this hash, we can look for the commit message:

```
git show 71f261f8
```



GRA-4152 --zsh-- 97x24

```
(base) ~ GRA-4152 % git show 71f261f8
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498 (HEAD -> master, origin/master, origin/HEAD)
Author: rogeliandorade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

        added honor code

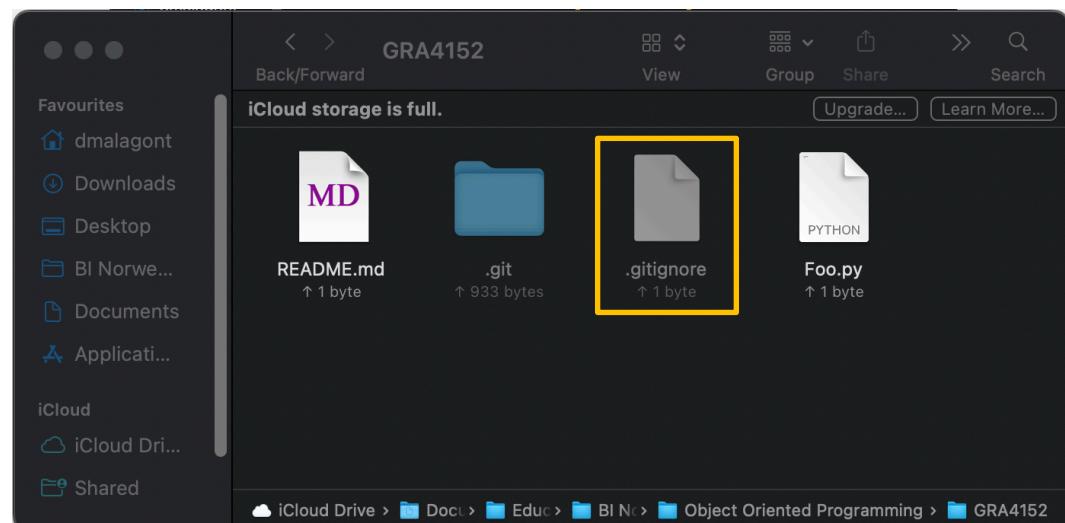
diff --git a/README.md b/README.md
index a7359ae..a404da3 100644
--- a/README.md
+++ b/README.md
@@ -8,3 +8,7 @@ pip install pylint
 sudo apt install graphviz
 pyreverse -o png <your code>.py
 ```

+
+## Honor Code
+You are free to form study groups and may discuss homework in groups. However, each student must
+write down the solutions and code from scratch independently and must understand the solution we
+ll enough. It is a honor code violation to copy, refer to, or look at written or code solutions f
+rom a previous year or solutions posted online (inspired by the Stanford Honor Code).
+
```

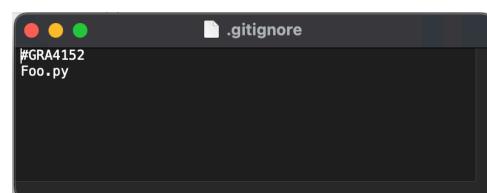
2. One common mistake when learning Git is to commit large files that should not be managed by Git or adding sensitive information. Add a .gitignore file to your portfolio code repository (<https://github.com/SXXXXXXX/GRA4152>) and exclude files and/or folders. You might need to create a Foo.py file to be excluded.

First, I create the .gitignore file in my code repository:

```
echo "#GRA4152" >> .gitignore
```

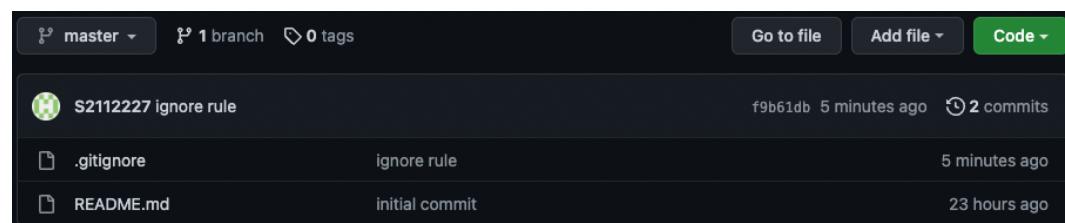


In the .gitignore file I specify that the file Foo.py should be ignored.



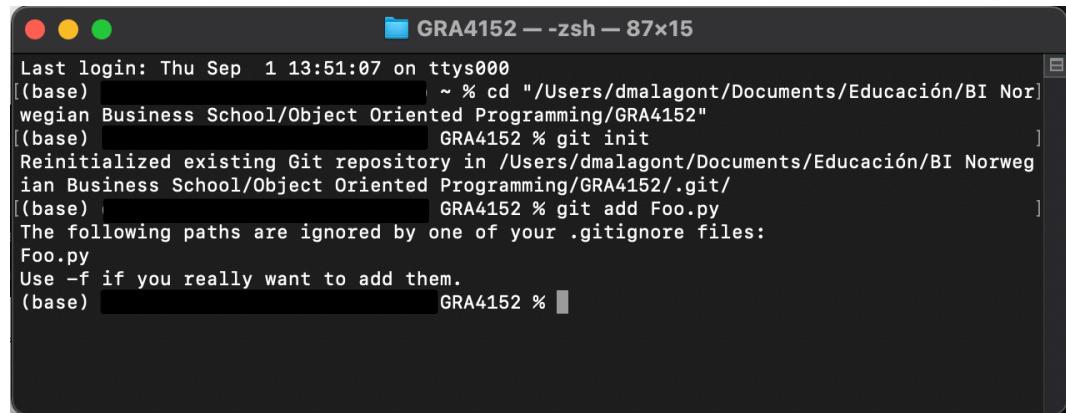
I begin tracking the .gitignore, commit and push it to github.

```
git add .gitignore
git commit -m "ignore rule"
git push
```



Now that the .gitignore file is in github we can test it by trying to track Foo.py:

```
git add Foo.py
```



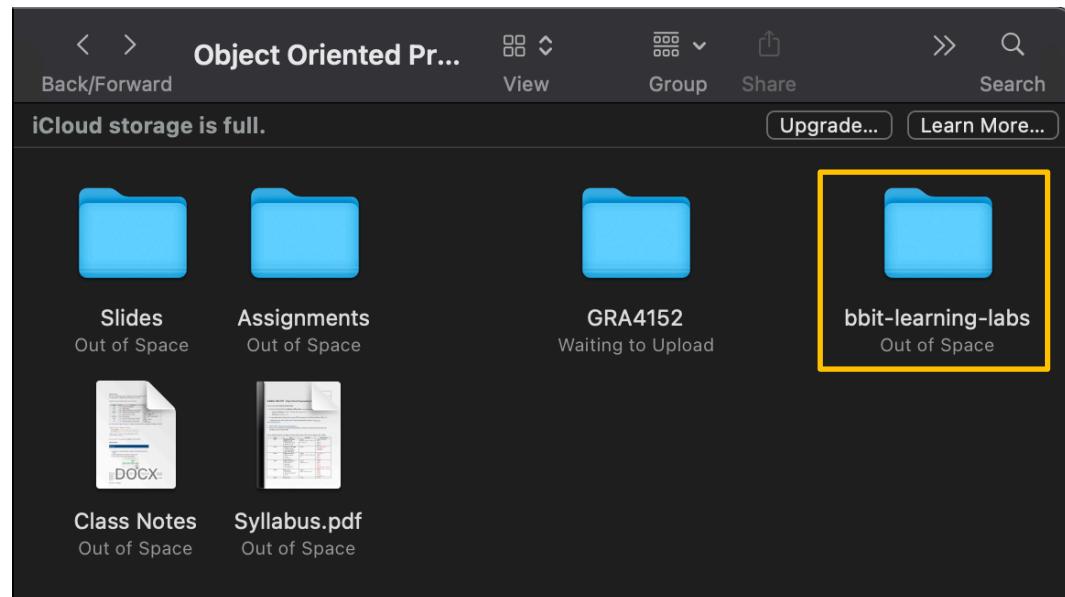
```
Last login: Thu Sep  1 13:51:07 on ttys000
[(base)] ~ % cd "/Users/dmalagont/Documents/Educación/BI Norwegian Business School/Object Oriented Programming/GRA4152"
[(base)] ~ % git init
[GRA4152 % git init]
Reinitialized existing Git repository in /Users/dmalagont/Documents/Educación/BI Norwegian Business School/Object Oriented Programming/GRA4152/.git/
[(base)] ~ % git add Foo.py
[GRA4152 % git add Foo.py]
The following paths are ignored by one of your .gitignore files:
Foo.py
Use -f if you really want to add them.
(base) [GRA4152 %]
```

As we can see the .gitignore file makes that git ignores the Foo.py file.

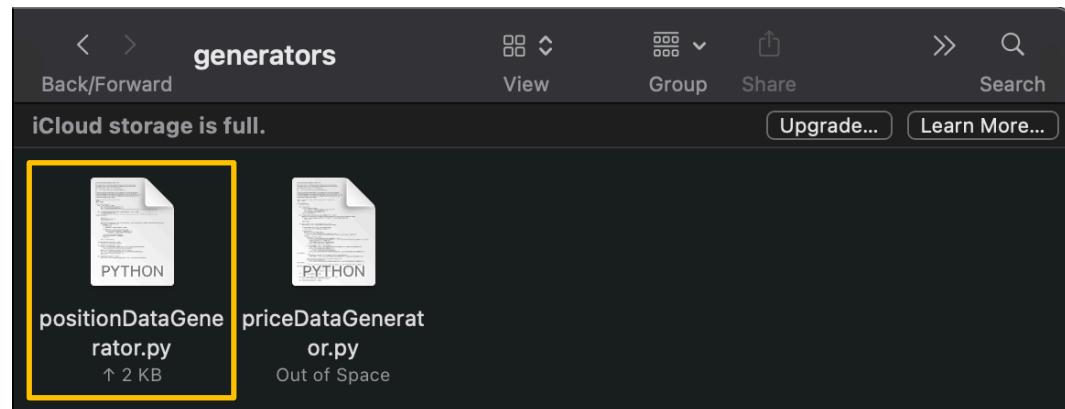
3. a) Clone some repository from GitHub and modify one of its existing files. What happens when you type git stash? What do you see when running git log --all -- oneline? Run git stash pop to undo what you did with git stash. In what scenario might this be useful? List your current stashes and delete them with git stash drop <stash\_id>

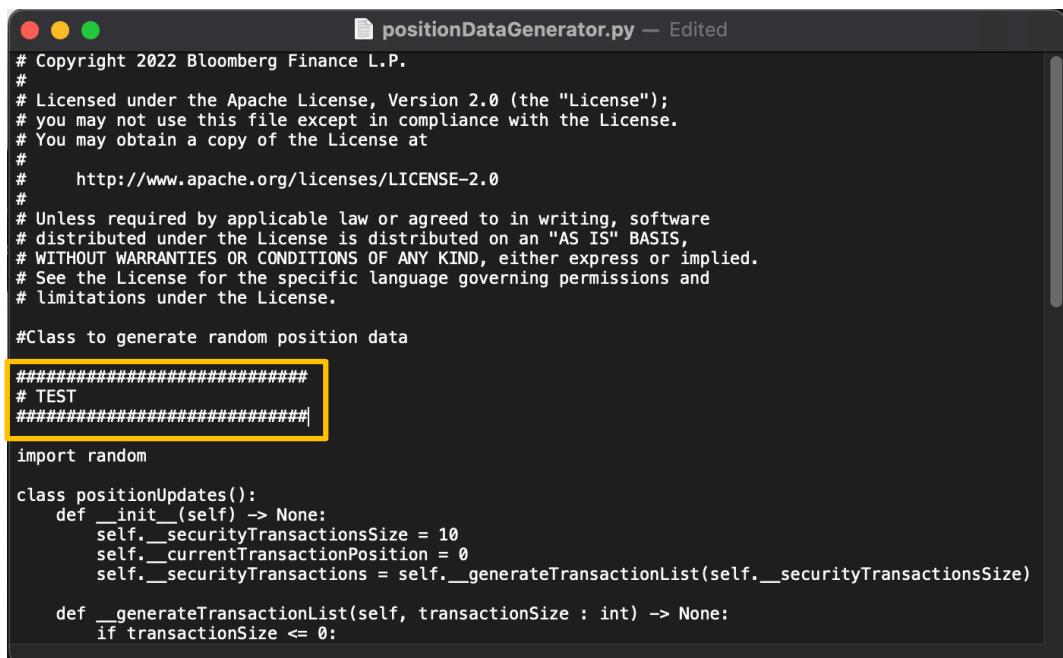
I clone a repository from GitHub:

```
git clone https://github.com/bloomberg/bbit-learning-labs.git
```



I modify the file "positionDataGenerator.py" within the repository:





```

# Copyright 2022 Bloomberg Finance L.P.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#Class to generate random position data

#####
# TEST
#####

import random

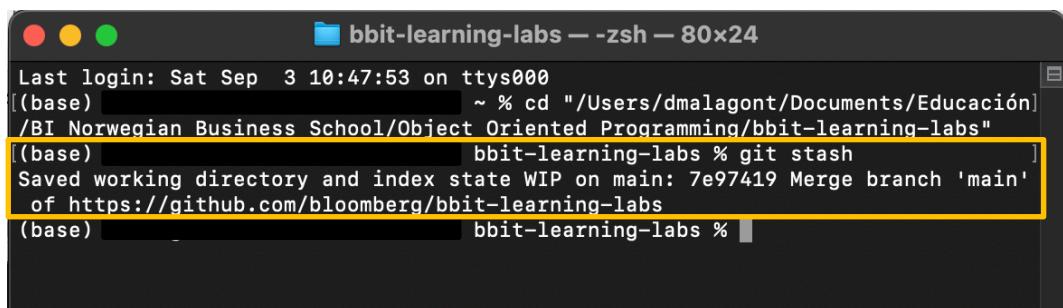
class positionUpdates():
    def __init__(self) -> None:
        self.__securityTransactionsSize = 10
        self.__currentTransactionPosition = 0
        self.__securityTransactions = self.__generateTransactionList(self.__securityTransactionsSize)

    def __generateTransactionList(self, transactionSize : int) -> None:
        if transactionSize <= 0:

```

I type git stash and check the file again:

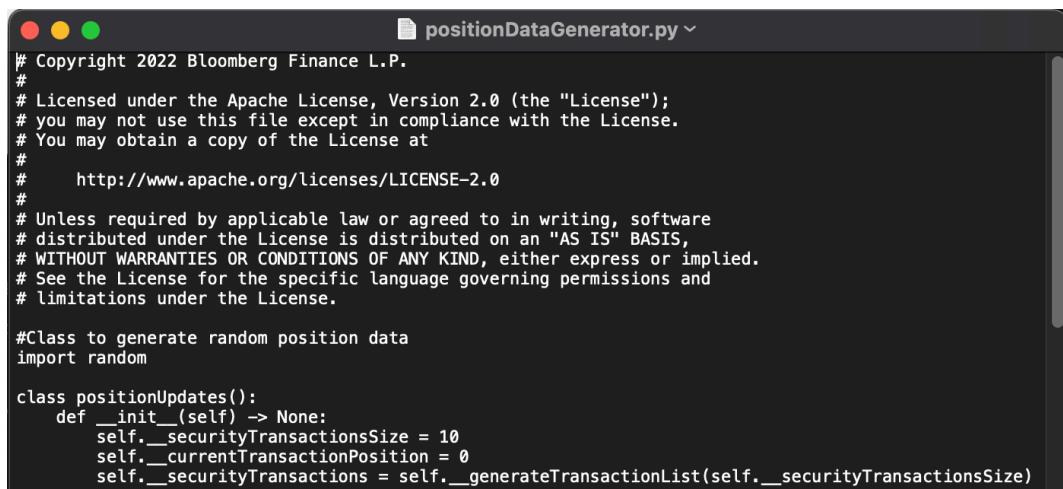
```
git stash
```



```

Last login: Sat Sep  3 10:47:53 on ttys000
[(base)] ~ % cd "/Users/dmalagon/Dокументos/Educación"
[BI Norwegian Business School/Object Oriented Programming/bbit-learning-labs"
[(base)] bbit-learning-labs % git stash
Saved working directory and index state WIP on main: 7e97419 Merge branch 'main'
  of https://github.com/bloomberg/bbit-learning-labs
(base) bbit-learning-labs %

```



```

# Copyright 2022 Bloomberg Finance L.P.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

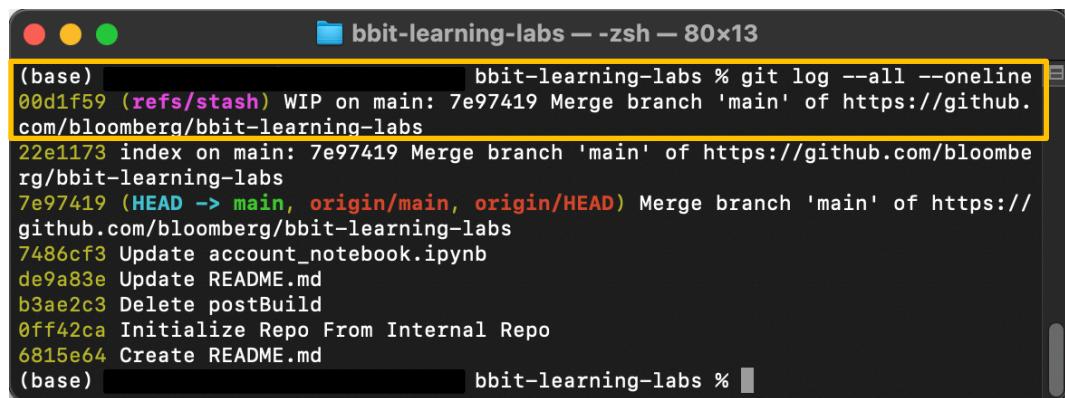
#Class to generate random position data
import random

class positionUpdates():
    def __init__(self) -> None:
        self.__securityTransactionsSize = 10
        self.__currentTransactionPosition = 0
        self.__securityTransactions = self.__generateTransactionList(self.__securityTransactionsSize)

```

The change was stashed in a working directory away and it is only possible to see the original file. I check the log to verify the stash:

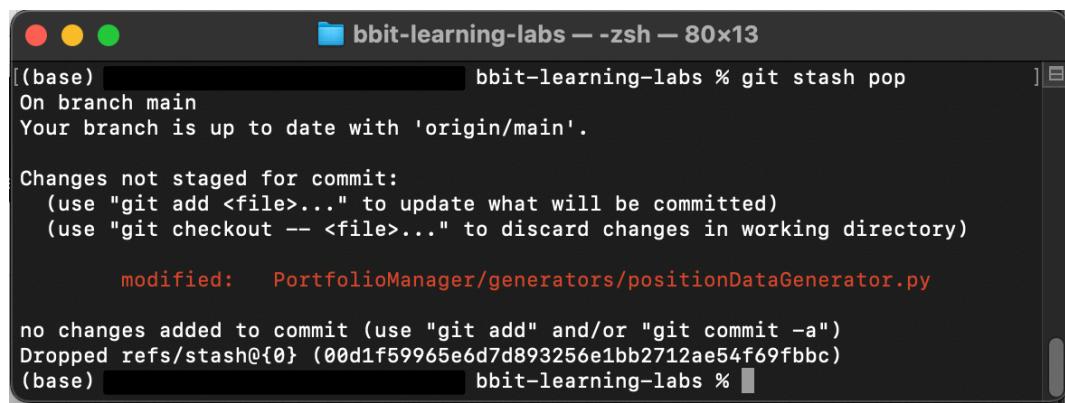
```
git log --all -- oneline
```



```
(base) bbit-learning-labs % git log --all --oneline
00d1f59 (refs/stash) WIP on main: 7e97419 Merge branch 'main' of https://github.com/bloomberg/bbit-learning-labs
22e1173 index on main: 7e97419 Merge branch 'main' of https://github.com/bloomberg/bbit-learning-labs
7e97419 (HEAD -> main, origin/main, origin/HEAD) Merge branch 'main' of https://github.com/bloomberg/bbit-learning-labs
7486cf3 Update account_notebook.ipynb
de9a83e Update README.md
b3ae2c3 Delete postBuild
0ff42ca Initialize Repo From Internal Repo
6815e64 Create README.md
(base) bbit-learning-labs %
```

Now I undo the git stash:

```
git stash pop
```

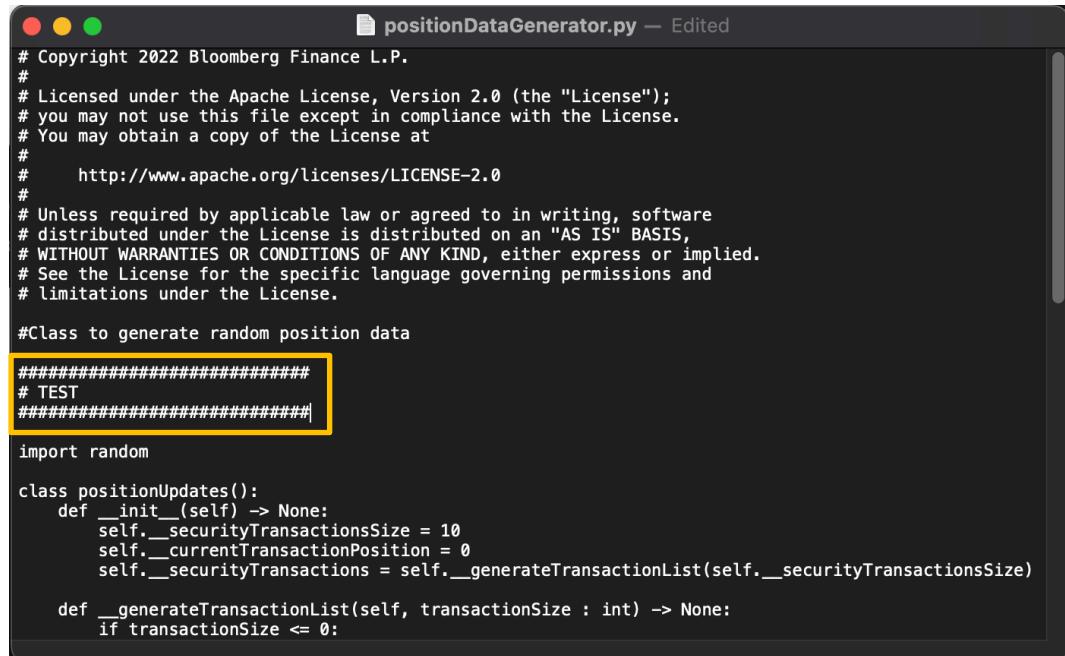


```
(base) bbit-learning-labs % git stash pop
On branch main
Your branch is up to date with 'origin/main'.

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

    modified:   PortfolioManager/generators/positionDataGenerator.py

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (00d1f59965e6d7d893256e1bb2712ae54f69fbcb)
(base) bbit-learning-labs %
```



```
# Copyright 2022 Bloomberg Finance L.P.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#Class to generate random position data
#####
# TEST
#####

import random

class positionUpdates():
    def __init__(self) -> None:
        self.__securityTransactionsSize = 10
        self.__currentTransactionPosition = 0
        self.__securityTransactions = self.__generateTransactionList(self.__securityTransactionsSize)

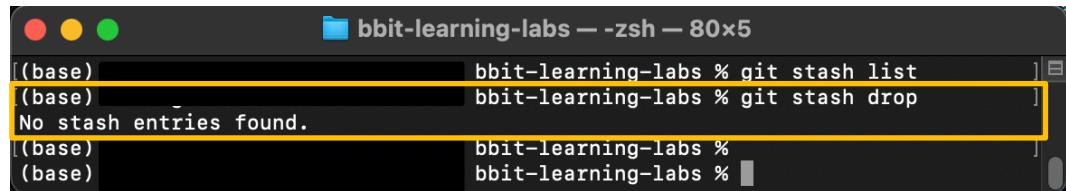
    def __generateTransactionList(self, transactionSize : int) -> None:
        if transactionSize <= 0:
```

This scenario (git stash) would be useful when one doesn't want to overwrite local changes that haven't been committed. Perhaps because the changes are not good enough to be committed or there are other priorities to address first.

Finally, I list the current stashes and delete them:

```
git stash list
```

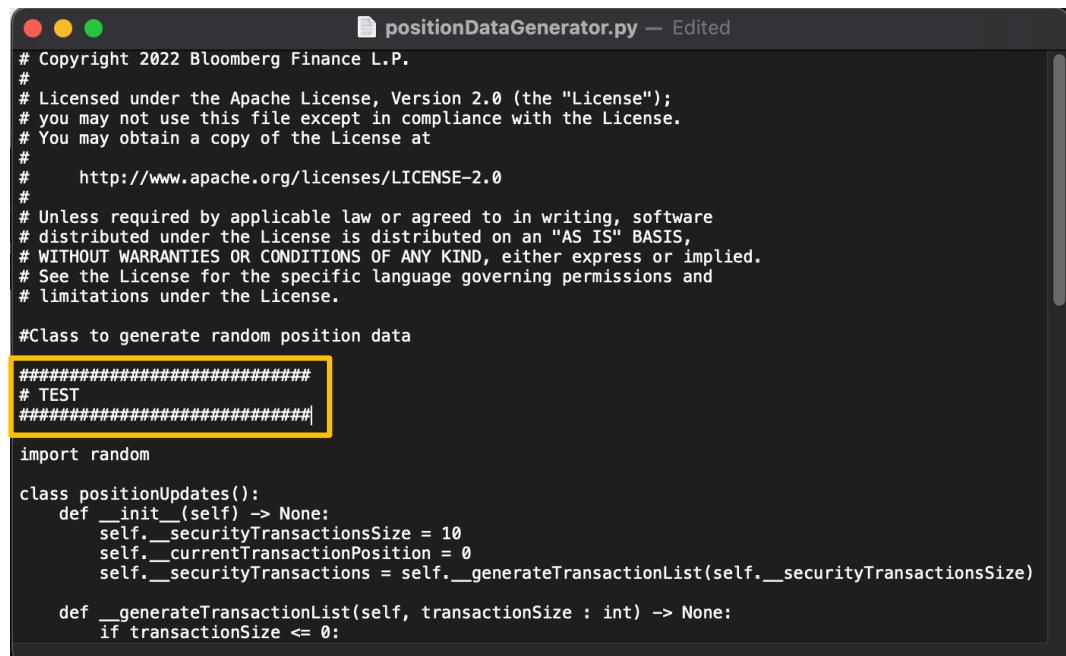
```
git stash drop
```



```
bbit-learning-labs — zsh — 80x5
[(base)] bbit-learning-labs % git stash list
[(base)] - bbit-learning-labs % git stash drop
No stash entries found.
[(base)] bbit-learning-labs %
[(base)] bbit-learning-labs %
```

**b) Now, modify a file and stash changes. Make a new modification to the same file, but this time commit those changes. What happens if you type git stash pop and open the file that you have modified? What do you see in the file?**

I modify the file “positionDataGenerator.py” within the repository:



```
positionDataGenerator.py — Edited
# Copyright 2022 Bloomberg Finance L.P.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#Class to generate random position data
#####
# TEST
#####

import random

class positionUpdates():
    def __init__(self) -> None:
        self.__securityTransactionsSize = 10
        self.__currentTransactionPosition = 0
        self.__securityTransactions = self.__generateTransactionList(self.__securityTransactionsSize)

    def __generateTransactionList(self, transactionSize : int) -> None:
        if transactionSize <= 0:
```

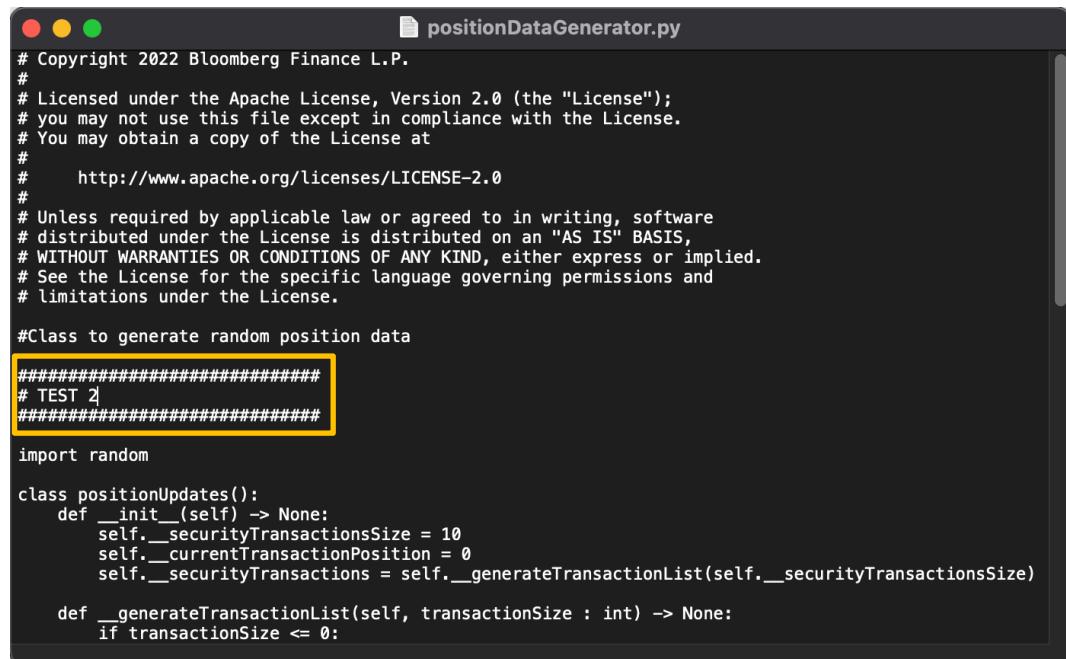
I type git stash, modify the file again, and commit the changes:

```
git stash
```

```
git add positionDataGenerator.py
```

```
git commit -m "test 2"
```

```
git push
```



```

# Copyright 2022 Bloomberg Finance L.P.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#Class to generate random position data
#####
# TEST 2
#####

import random

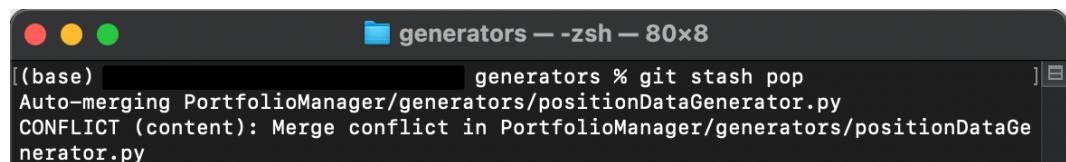
class positionUpdates():
    def __init__(self) -> None:
        self.__securityTransactionsSize = 10
        self.__currentTransactionPosition = 0
        self.__securityTransactions = self.__generateTransactionList(self.__securityTransactionsSize)

    def __generateTransactionList(self, transactionSize : int) -> None:
        if transactionSize <= 0:

```

Now, I reapply the changes to the working copy:

```
git stash pop
```

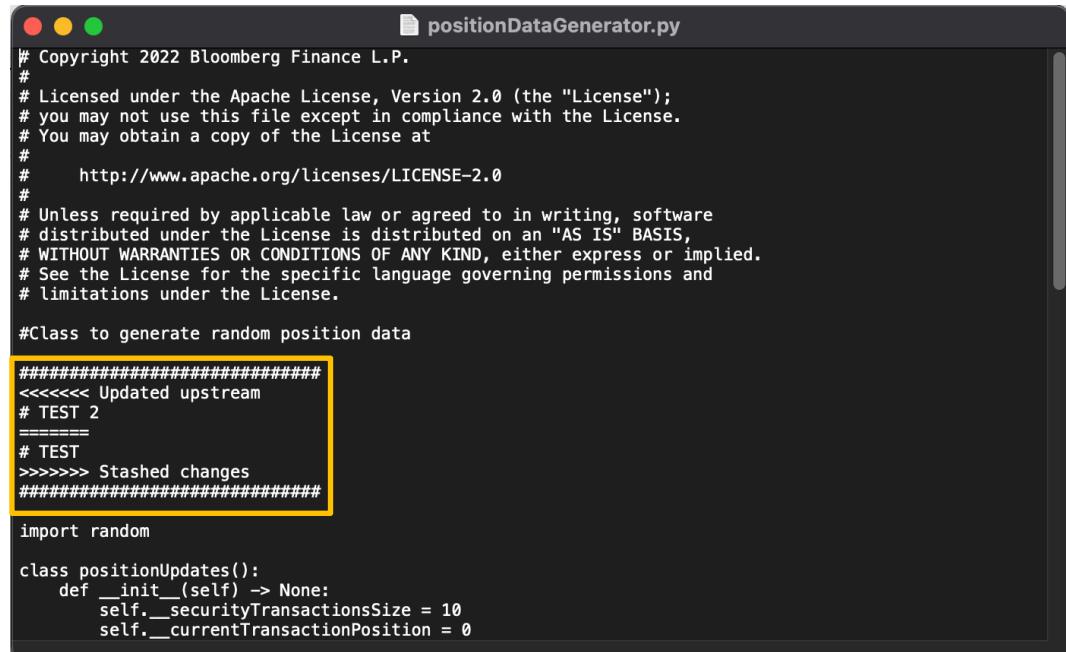


```

generators -- zsh -- 80x8
[(base)] generators % git stash pop
Auto-merging PortfolioManager/generators/positionDataGenerator.py
CONFLICT (content): Merge conflict in PortfolioManager/generators/positionDataGenerator.py

```

Once I open the file, I can observe that the two changes (stash and committed) show up in the same version:



```

# Copyright 2022 Bloomberg Finance L.P.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#Class to generate random position data
#####
<<<< Updated upstream
# TEST 2
=====
# TEST
>>>> Stashed changes
#####

import random

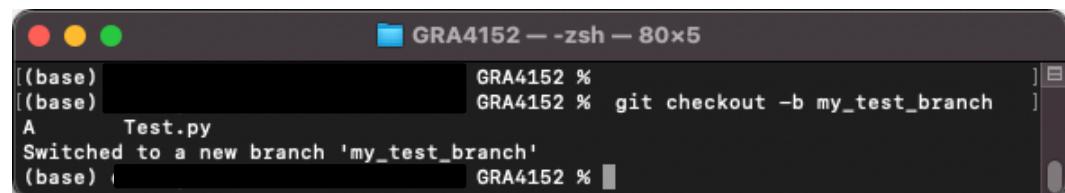
class positionUpdates():
    def __init__(self) -> None:
        self.__securityTransactionsSize = 10
        self.__currentTransactionPosition = 0

```

- 4. Create a new branch in your class repository (<https://github.com/SXXXXXXX/GRA4152>) and call it `my_test_branch`. Explore both branches, by switching back and forth. Add a comment line in any file in the branch `my_test_branch`, add and commit your changes. Finally, merge `my_test_branch` into master.**

In my class repository I create a new branch called “`my_test_branch`”:

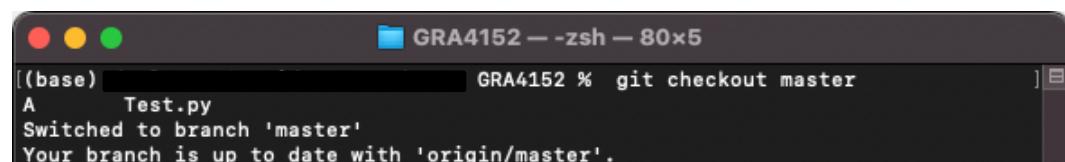
```
git checkout -b my_test_branch
```



A screenshot of a terminal window titled "GRA4152 — zsh — 80x5". The command "git checkout -b my\_test\_branch" is entered, followed by the output: "Switched to a new branch 'my\_test\_branch'".

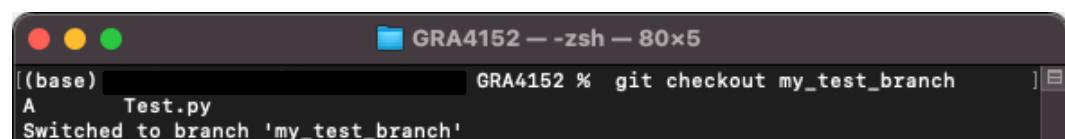
It is possible to switch back and forth between branches using the following code:

```
git checkout master
```



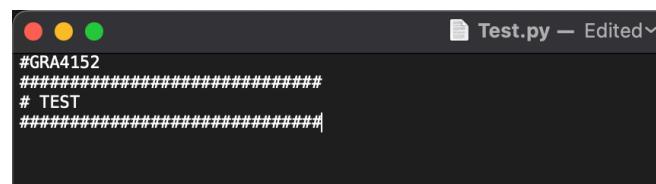
A screenshot of a terminal window titled "GRA4152 — zsh — 80x5". The command "git checkout master" is entered, followed by the output: "Switched to branch 'master'. Your branch is up to date with 'origin/master'."

```
git checkout my_test_branch
```



A screenshot of a terminal window titled "GRA4152 — zsh — 80x5". The command "git checkout my\_test\_branch" is entered, followed by the output: "Switched to branch 'my\_test\_branch'".

I add a comment line in the file “`Test.py`”:

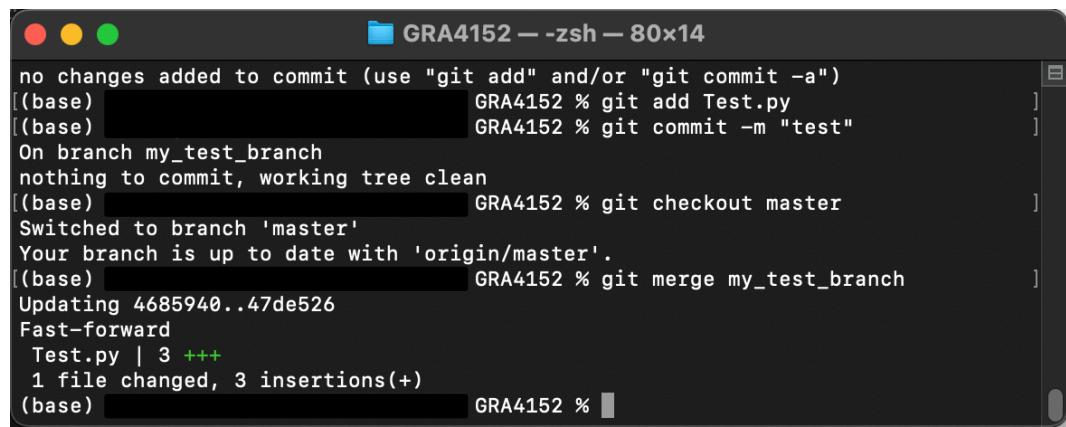


A screenshot of a code editor titled "Test.py — Edited~". The file contains the following code:  

```
#GRA4152
#####
# TEST
#####
```

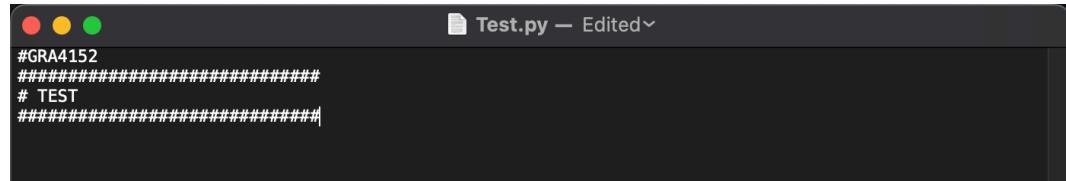
Finally, I add, commit and merge “`my_test_branch`” into “`master`”:

```
git add Test.py
git commit -m "test"
git checkout master
git merge my_test_branch
```



```
no changes added to commit (use "git add" and/or "git commit -a")
[(base)]                                     GRA4152 % git add Test.py
[(base)]                                     GRA4152 % git commit -m "test"
On branch my_test_branch
nothing to commit, working tree clean
[(base)]                                     GRA4152 % git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[(base)]                                     GRA4152 % git merge my_test_branch
Updating 4685940..47de526
Fast-forward
  Test.py | 3 +++
  1 file changed, 3 insertions(+)
(base)  GRA4152 %
```

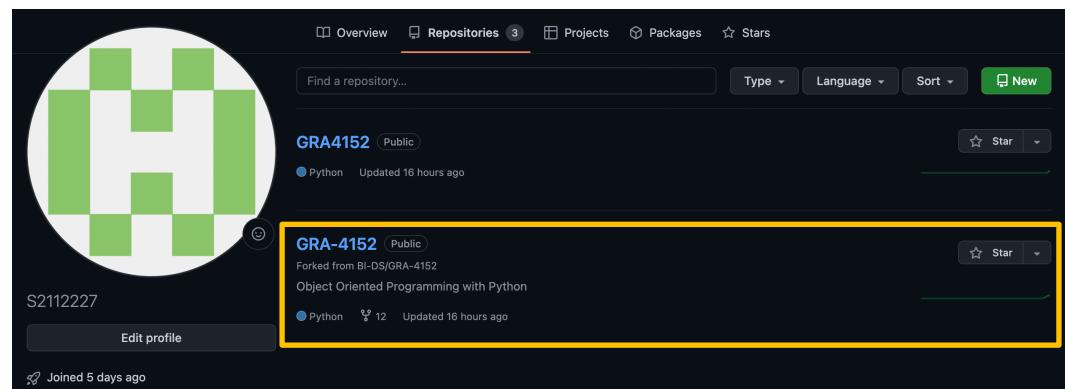
If I check the file again in the “master” branch, it is possible to see that the changes are now in the main branch.



```
#GRA4152
#####
# TEST
#####
```

- 5. Fork the class repository (<https://github.com/BI-DS/GRA-4152>) and clone it, so you can contribute to its development. Send me a pull request to add a text file with your student id, i.e. SXXXXXXX.txt.**

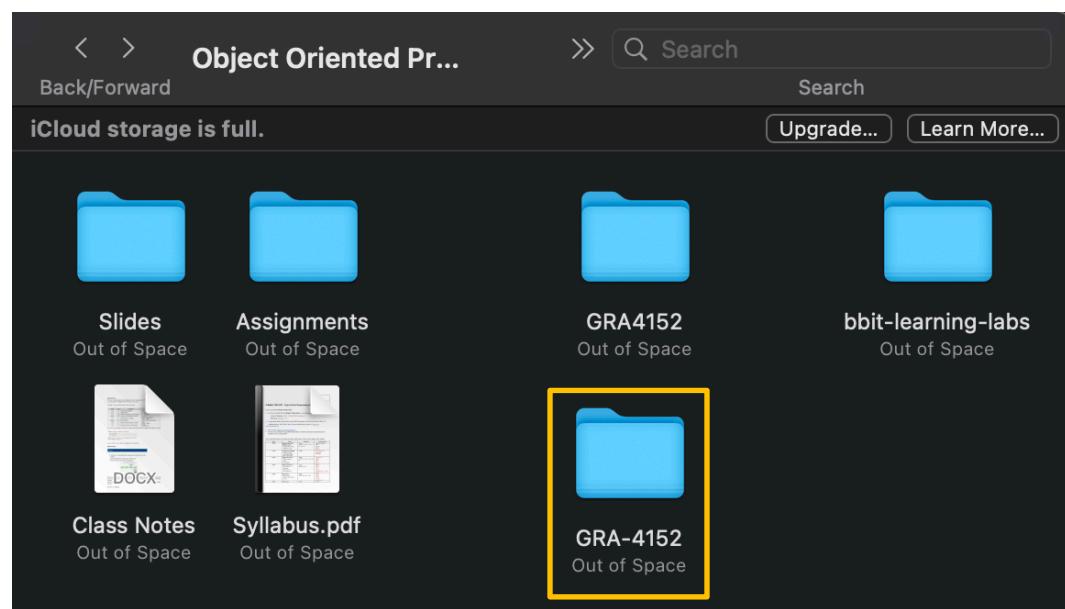
First, I fork the repository and clone it:



```
git clone https://github.com/S2112227/GRA-4152.git
```

```
git remote add upstream https://github.com/BI-DS/GRA-4152
```

```
(base) [112227/GRA-4152.git] Object Oriented Programming % git clone https://github.com/S2112227/GRA-4152.git
Cloning into 'GRA-4152'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 32 (delta 8), reused 27 (delta 3), pack-reused 0
Unpacking objects: 100% (32/32), done.
(base) [112227/GRA-4152.git] Object Oriented Programming %
(base) [112227/GRA-4152.git] Object Oriented Programming % git remote add upstream https://github.com/BI-DS/GRA-4152
(base) [112227/GRA-4152.git] Object Oriented Programming %
```



Now, I create a text file with my student id in the class repository and send a pull request:

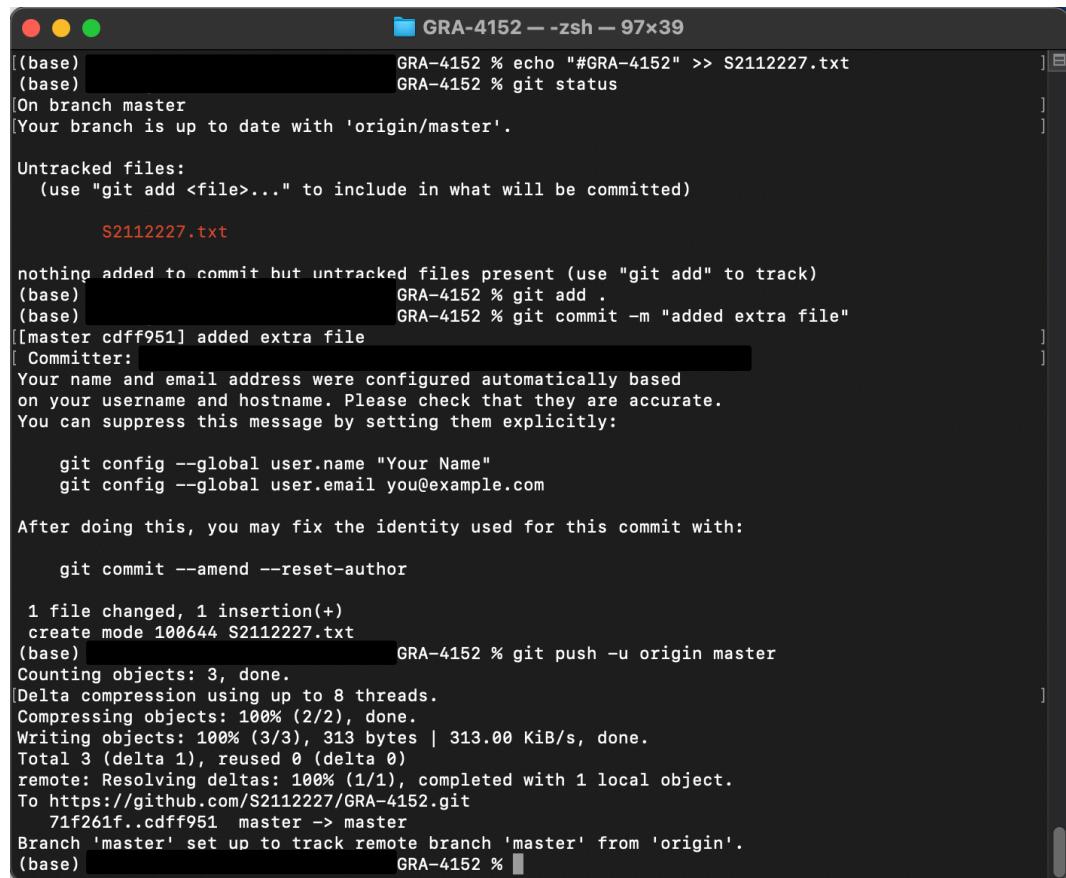
```
echo "#GRA-4152" >> S2112227.txt

git status

git add .

git commit -m "added extra file"

git push -u origin master
```



The screenshot shows a terminal window titled 'GRA-4152 -- -zsh -- 97x39'. The user runs several commands to update a local repository:

- Creates a file: `echo "#GRA-4152" >> S2112227.txt`
- Checks the status: `git status`
- Adds the file: `git add .`
- Commits the change: `git commit -m "added extra file"`
- Pushes the changes to the remote branch: `git push -u origin master`

The terminal also displays configuration prompts for the committer's name and email, which are set to 'Your Name' and 'you@example.com' respectively. It also shows the progress of the push operation, including object counting, compression, and writing to the remote repository.

After pushing the new text file to GitHub it is possible to observe a “Create pull request”. I click it and create the pull request for the class repository. As seen in the images below, the pull request shows up in the original repository:

**Comparing changes**

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base repository: BI-DS/GRA-4152 → base: master ← head repository: S2112227/GRA-4152 → compare: master →

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

→ 1 commit      1 file changed      ↗ 1 contributor

→ Commits on Sep 4, 2022

added extra file  
David Malagón Triana authored and David Malagón Triana committed 6 minutes ago

Showing 1 changed file with 1 addition and 0 deletions.

Split Unified

1 S2112227.txt

... ... @@ -0,0 +1 @@ 1 + #GRA-4152

**Open a pull request**

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base repository: BI-DS/GRA-4152 → base: master ← head repository: S2112227/GRA-4152 → compare: master →

✓ Able to merge. These branches can be automatically merged.

added extra file

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Allow edits by maintainers ?

Create pull request

**added extra file #11**

[Open](#) S2112227 wants to merge 1 commit into BI-DS:master from S2112227:master

Conversation 0 Commits 1 Checks 0 Files changed 1

S2112227 commented now

No description provided.

added extra file

Add more commits by pushing to the master branch on S2112227/GRA-4152.

This branch has no conflicts with the base branch  
Only those with write access to this repository can merge pull requests.

