Assignment #1

Git Exercises

1. **Clone the repository for the class**

   #Cloning the repository from Github

   PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git clone https://github.com/BI-DS/GRA-4152

   #change directory to the cloned repository

   PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> cd .\GRA-4152

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

   The graph was taken from Github's Insight feature, showing a graph of all commits made in the repository GRA4152.

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

   #check git commits version history through terminal

   PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> git log README.md

```
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code

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

    adding instructions for UML
```

The last time README.md was updated was Wednesday August 31 2023 with commit key "71f261f8dbb09c828dfd2be1ad664a14b1fbc498".

c. What was the last commit message associated with the last modification to the README.md?

#check the message put in the latest commit/last modification on README.md

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> git log README.md

```
PS C:\Users\justi\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> git log README.md
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code
```

You can also use git log to show all the details of the latest commit including the comment made which is below saying "added honor code."

2. *#cd out of the cloned repository back to own repository*

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> cd ..

#creating a new .gitignore file in the repository.

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> notepad .gitignore

#add, commit, and push it in the repository

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git add .gitignore.txt

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git commit -m "Add gitignore txt file"

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git push

#creating a new foo.py file and then adding it to the repository.

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git add foo.py

#commit the changes and push it in the repository
PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git commit -m "Added foo.py"

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git push

3. **Cloning a new repository (freecodecamp)**
    a. **use git stash**

    #Clone the new repository if your choice

    PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git clone https://github.com/EbookFoundation/free-programming-books

    #change cd

    PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152> cd .\GRA4152\free-programming-books

    #use git stash

    PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git stash

    When you do git stash the changes you made will disappear and will be stashed somewhere and can be seen using git stash list.

    b. Git log –all –oneline

    # using git log –all –oneline

    PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git log --all –oneline

    When you do this code, it will show you a simplified and summarized view of all the git commit history of all the branches in the repository including all the comments and commit keys.

c. Git stash pop

 # using git stash pop

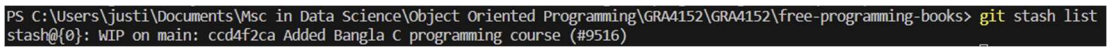
PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git stash pop

This is useful in cases when you want to revert back to the original state of the file without the temporary changes. If you have made changes to the working directory that you are working on but later on decided not to keep them, you can use git stash pop to get rid of all the changes you have made and revert back to the original.

d. Current stashes and dropping it.

 # show all current stashes made.

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git stash list



 # drop the current stash id you have stash@{0}

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git stash drop "stash@{0}"

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

a. *# staging the changes made and committing*

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git add "README.md"

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git commit -m "Added wazzzupppp in the text"

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\free-programming-books> git stash pop

Once you used git stash pop again and opened the file you made changes in, the changes you made which were initially stashed (disappeared from the file) will reappear again.

4. **Create a new branch called my_test_branch**

 # creating a new branch called my_test_branch

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git checkout -b "my_test_branch"

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

```
PS C:\Users\justi\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
PS C:\Users\justi\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git checkout my_test_branch
Switched to branch 'my_test_branch'
```

b. **Make changes to readme.md file in the branch**

 # add a new comment in the readme file stage it and then commit it.

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git add "README.md"

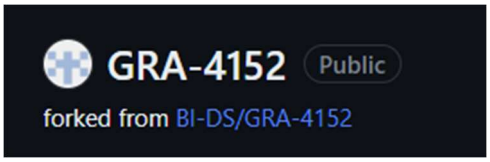
PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git commit -m "added a comment about new branch"

c. **Merge the branches**

 # merge the my_test_branch into the master branch.

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git merge my_test_branch

5. **Fork the class repository.**



 # cloning the forked

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152> git clone https://github.com/S1044345/GRA-4152.git

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> git add "S1044345.txt"

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> git commit -m "Added student txt file"

PS C:\Users\xxxx\Documents\Msc in Data Science\Object Oriented Programming\GRA4152\GRA4152\GRA-4152> git push