

## GRA4152 Assignment 1:

Glt exercises:

Task 1:

A)

```
cd <in to the directory I wanted to clone it to>
git clone https://github.com/BI-DS/GRA-4152
git log --graph
Output:
```

```
all
* 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 async exercise from lecture 1
* 0f6036b adding 1 async 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
```

B)

```
git log -n 1 -- README.md
```

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

c)

```
git show 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
```

```
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. 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).
```

## Task 2:

First I go in to the root folder of the repository

Then I write:

```
touch .gitignore
Touch foo.py
git add .gitignore
git commit -m "Added .gitignore"
git push origin master
```

When i went to github now the gitignore file is there!

(Inside the .gitignore file I write foo.py, I also had a file DS\_Store that I want to ignore)

## Task 3:

a)

I am cloning an open git clone test repository:

<https://github.com/itsfoss/text-files.git>

I modified a .txt file then I wrote:

```
git stash
```

```
Saved working directory and index state WIP on master: 23b1322 i3 config files
```

When I write git stash the changes I made in the .txt file is gone, and back to the original when I go in to the file.

b)

```
git log --all --online
```

```
8102104 (refs/stash) WIP on master: 23b1322 i3 config files
361941e index on master: 23b1322 i3 config files
23b1322 (HEAD -> master, origin/master, origin/HEAD) i3 config files
3fd8d49 Delete I3_compressed/I3 directory
5fb6782 i3 config files
f4bea39 Add files via upload
b83599d Sort command sample text file
ed56db2 Delete split_command_sample_log_file.log
b294b23 Uniq command sample text file
3d094c1 Split command example log file
2d7a8cf Tabbed File
3647aa4 Delete tab-file
0a914ab Create tab-file
d55ee24 Add files via upload
f1e00f9 Update agatha.txt
718ae35 Add files via upload
fd4ddb0 Initial commit
```

I see the git commit log/history.

c)

```
git stash pop
```

```
error: Your local changes to the following files would be overwritten by merge:
    sherlock.txt
Please commit your changes or stash them before you merge.
Aborting
The stash entry is kept in case you need it again.
```

This command undoes what I did with git stash. Can be useful if I have stashed changes, but then I want to them to my current working directory

d)

```
git stash list
```

```
stash@{0}: WIP on master: 23b1322 i3 config files
```

```
git stash drop stash@{0}
```

```
Dropped stash@{0} (cadb50c6cbf9c683657f0bb04e5255227197bc42)
```

e) / a)

(I do changes to the file)

```
git stash
```

(do some more changes to the file)

```
git commit -m "Added a line"
```

```
e"
```

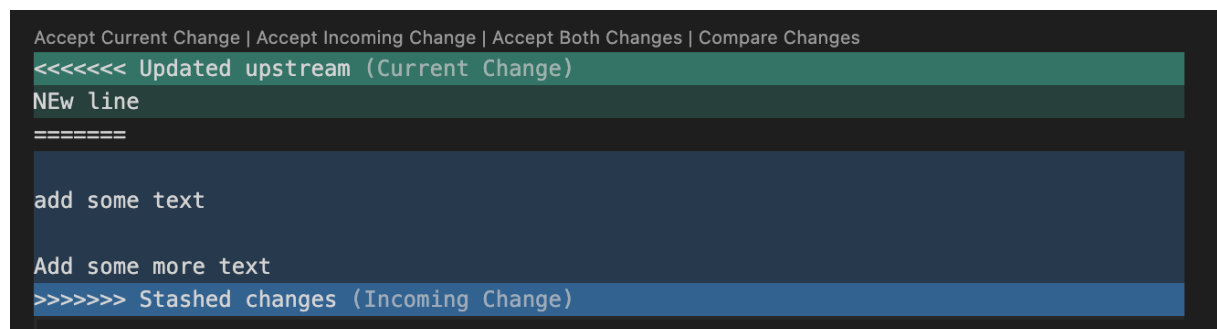
```
[master 0d95aa4] Added a line  
2 files changed, 2 insertions(+)  
create mode 100644 .DS_Store
```

```
git stash pop
```

```
Auto-merging sherlock.txt
```

```
CONFLICT (content): Merge conflict in sherlock.txt
```

```
The stash entry is kept in case you need it again.
```



This is what I have in vs code when I open the file:

The text: "NEw line" is what I added last

The two others "add som text" and "add some more text" is from the earlier stash. In case I would ever need it again.

Task 4:

a)

In the first command here I create the new branch and switch to it right away

```
git checkout -b my_test_checkout
```

```
Switched to a new branch 'my_test_branch'
```

```
git checkout master
```

```
Switched to branch 'master'
```

```
Your branch is up to date with 'origin/master'.
```

```
git checkout my_test_branch
```

```
Switched to branch 'my_test_branch'
```

(Here I jump back between the master and my\_test\_branch)

```
b)
git touch my_test_file.txt
echo "This is a new line of text" >> my_test_file.txt
git commit -m "Made a text file and added text"
[my_test_branch 3b498fb] Made a text file and added text
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 my_test_file.txt
git checkout master
```

```
c)
git merge my_test_branch
Updating da318af..3b498fb
Fast-forward
 my_test_file.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 my_test_file.txt
Git commit -m "Merged my_test_file into master"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Untracked files:
  my_text_file.txt

nothing added to commit but untracked files present
git push origin master
```

```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 296 bytes | 296.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://github.com/S1049546/GRA4152.git
da318af..3b498fb master -> master
```

Task 5:

So I go to the URL and press the fork button in github. Now I have a fork as a repository, so I can clone it:

Command:

```
git clone https://github.com/S1049546/GRA-4152.git
touch S1049546.txt
git add .
git commit -m "Added S1049546.txt"
```

```
[master 65824ad] Added S1049546.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 S1049546.txt
git push origin master
```

```
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), 270 bytes | 270.00 KiB/s, done.  
Total 3 (delta 1), reused 0 (delta 0)  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To https://github.com/S1049546/GRA-4152.git  
dc730e8..65824ad master -> master
```