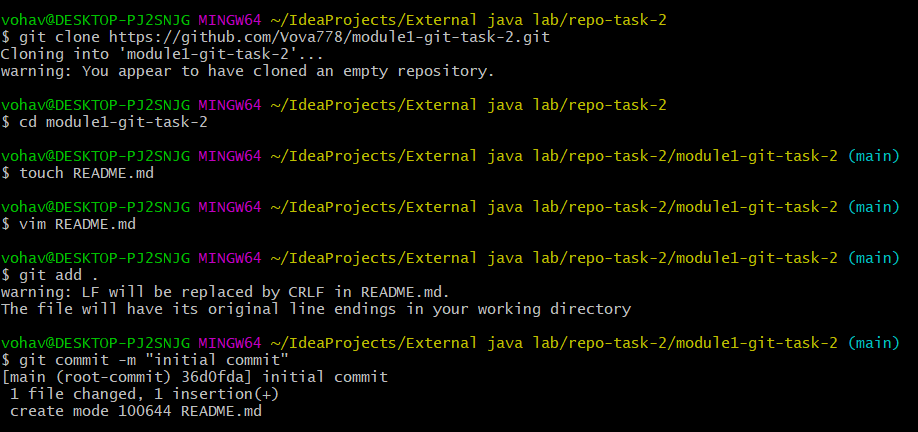
### **Task 2**

#### **Part 1**

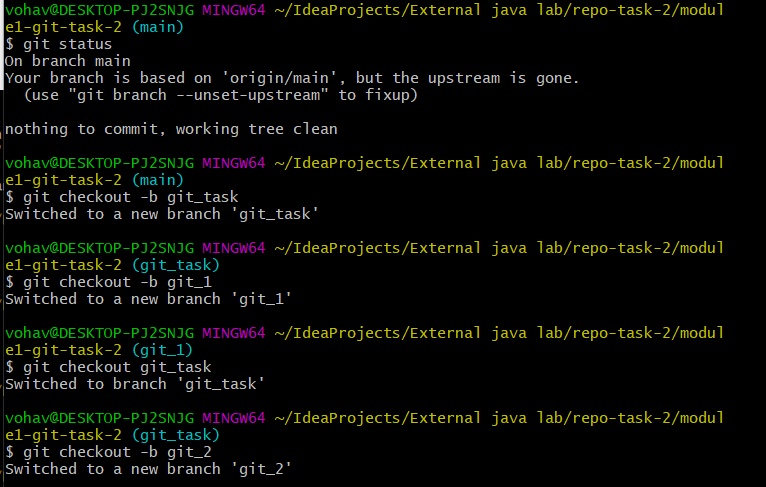
We are going to practice some skills obtained in the previous task. If you come across something you still don’t know, please use links provided in the descriptions, internet search, other mentees and your mentor as sources of knowledge and help.

First create a new remote repository. Clone it. Create new file named *README.md* with any text and commit it with "initial commit" message.



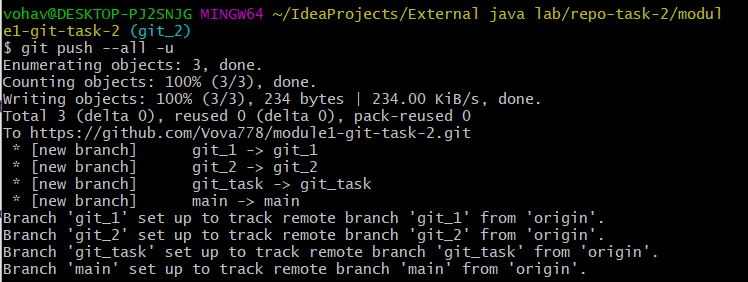
When you have first commit, you're able to create branches. To start this task, you need to create 3 branches:

1. Start from *master*. From *master* create branch named *git\_task* and checkout it.
2. From *git\_task* you should createw two branches: *git\_1* and *git\_2*.



Push every branch to the remote (see git push --all -u and read about upstream and tracked branches). At the end you should have 4 branches and *README.md* file both in your local and remote repositories:

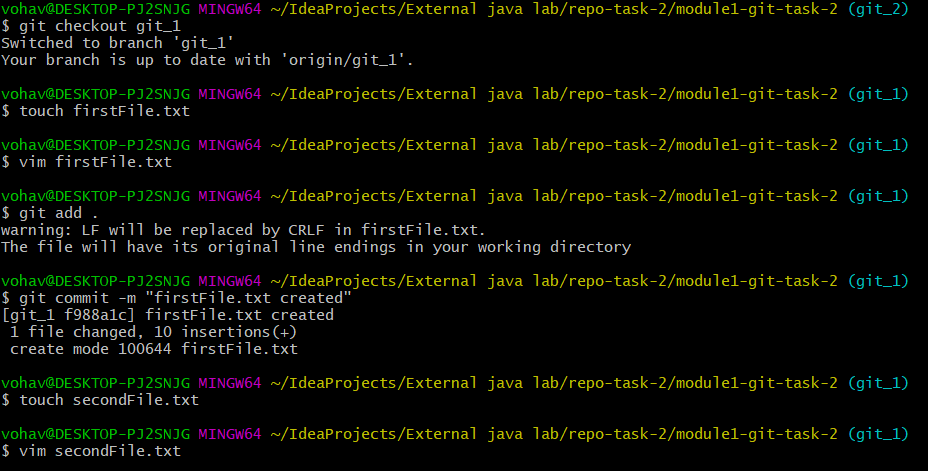
* *master*, *git\_task*, *git\_1* and *git\_2*

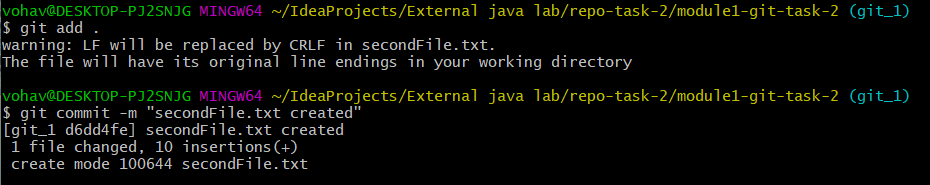


*The -u(upstream) flag means your local branch will become a tracking branch. That branch tracks a remote branch (the "upstream" branch) so that future git pull will know which branch to merge from and git push will be directed to the correct remote branch.*

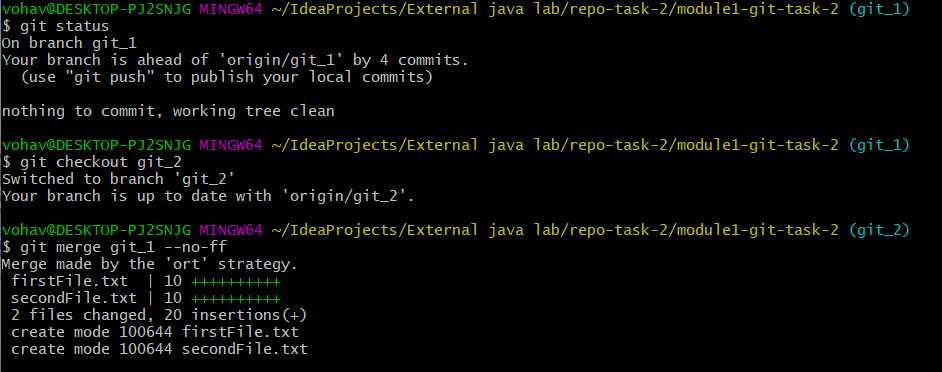
Now proceed these steps:

1. *git\_1*: Add and commit *firstFile.txt* file with 10 lines.
2. *git\_1*: Add and commit *secondFile.txt* file with 10 lines.





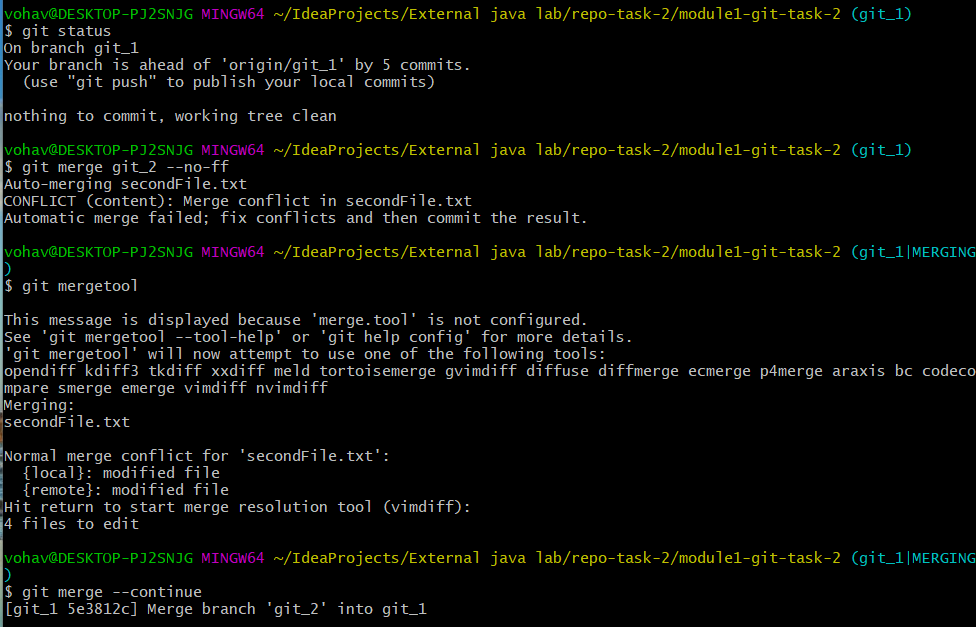
3. merge branch *git\_1* to *git\_2*

**

4. *git\_2*: Update and commit any two lines in *secondFile.txt*.

5. *git\_1*: Update and commit the same 2 lines with the different info in *secondFile.txt*

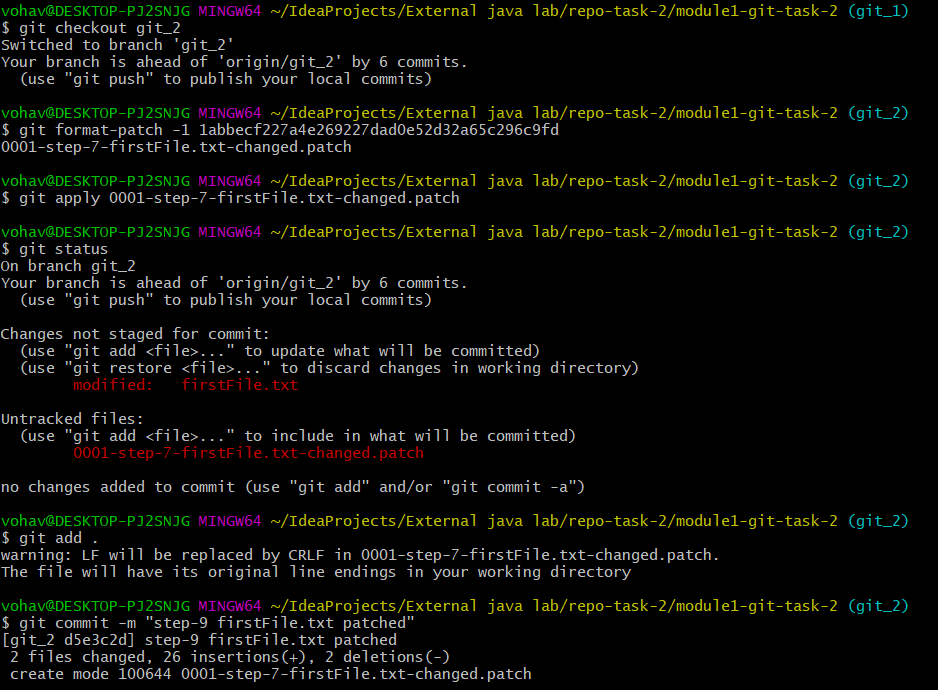
6 merge branch *git\_2* to *git\_1*, resolve conflict. Left all (4) modified lines. Commit.



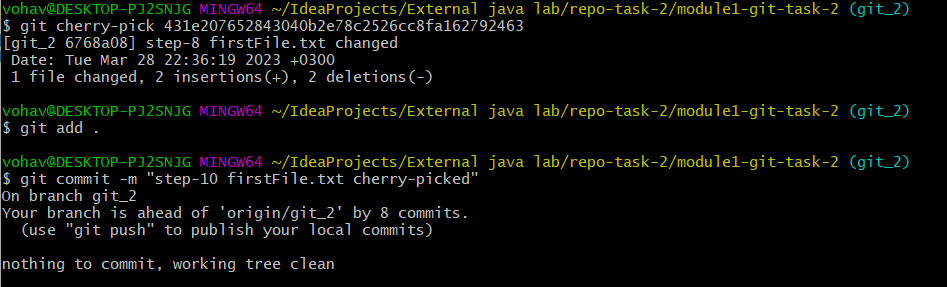
7. *git\_1*: Update and commit *firstFile.txt* file, modify two lines.

8. *git\_1*: Update and commit *firstFile.txt* file, modify another two lines.

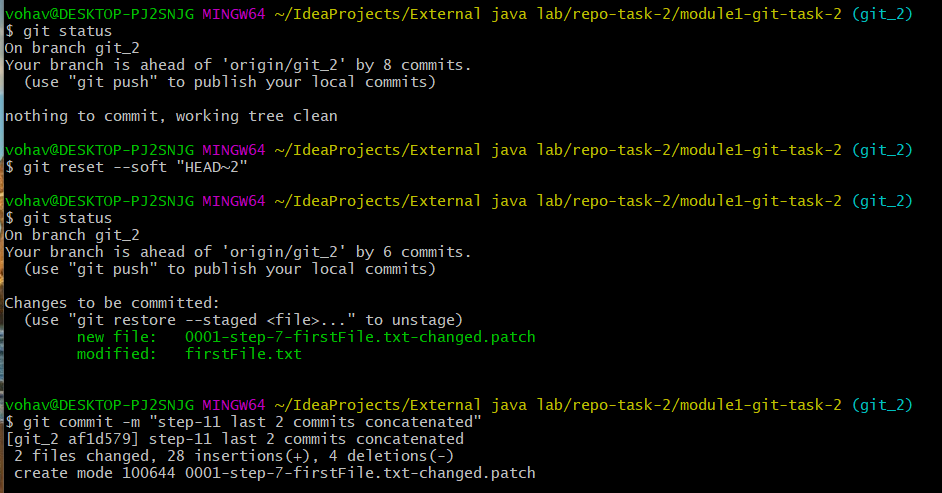
9. Transfer changes of commit from Step 7 only to *git\_2*, using format patch.



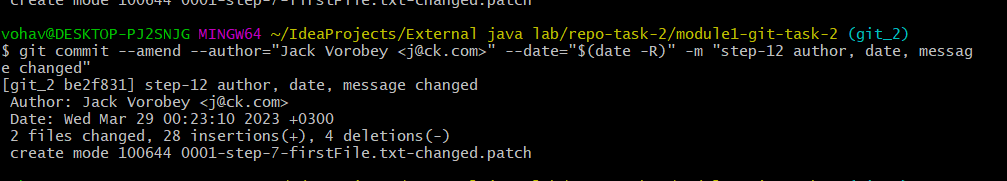
10. Transfer changes of commit from Step 8 only to *git\_2*, using cherrypick command.

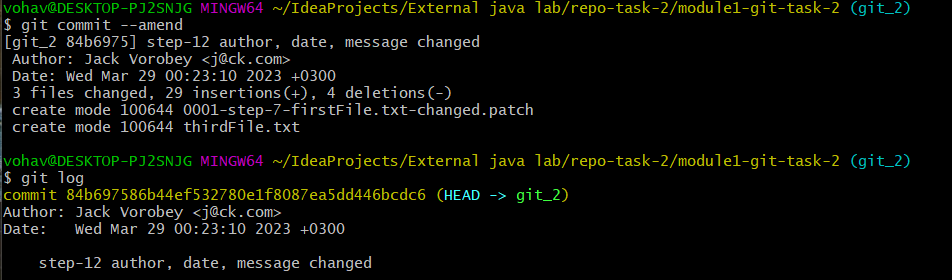


11. *git\_2*: Concatenate the last two commits using reset + commit commands.

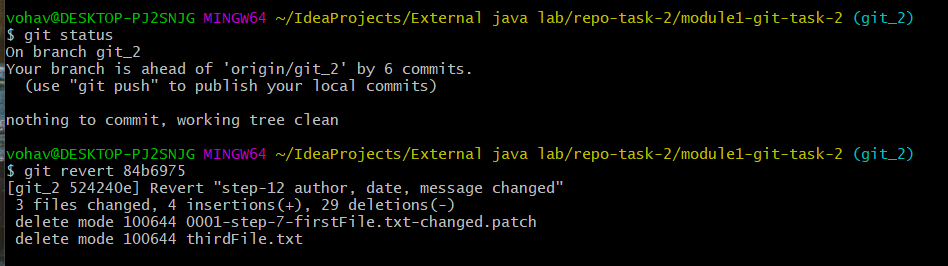


12 .*git\_2*: Change date, author and message of the last commit and add non-empty *thirdFile.txt* file to it.

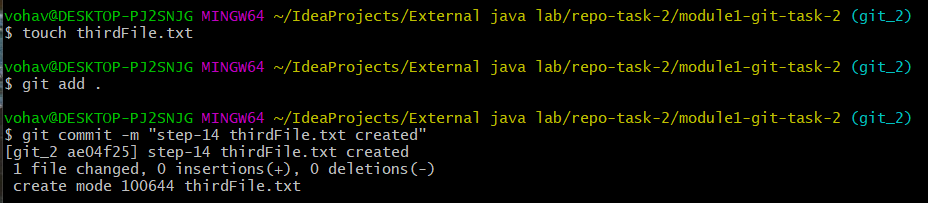




13. *git\_2*: Create a new commit that reverts changes of the last one.



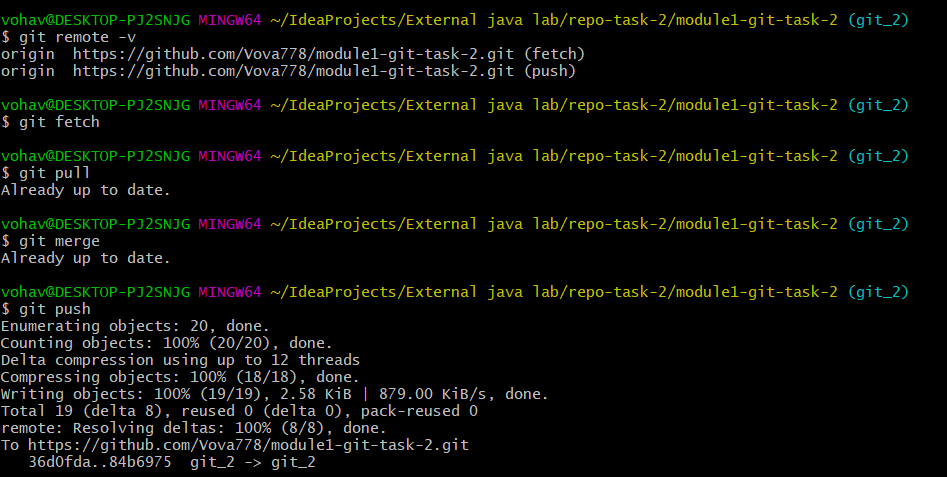
14. *git\_2*: Create and commit *thirdFile.txt* file.



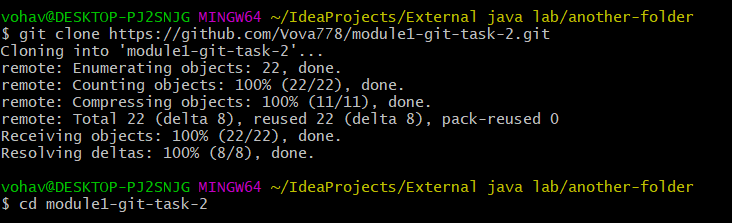
15. *git\_2*: Run command that removes all changes of the last two commits.



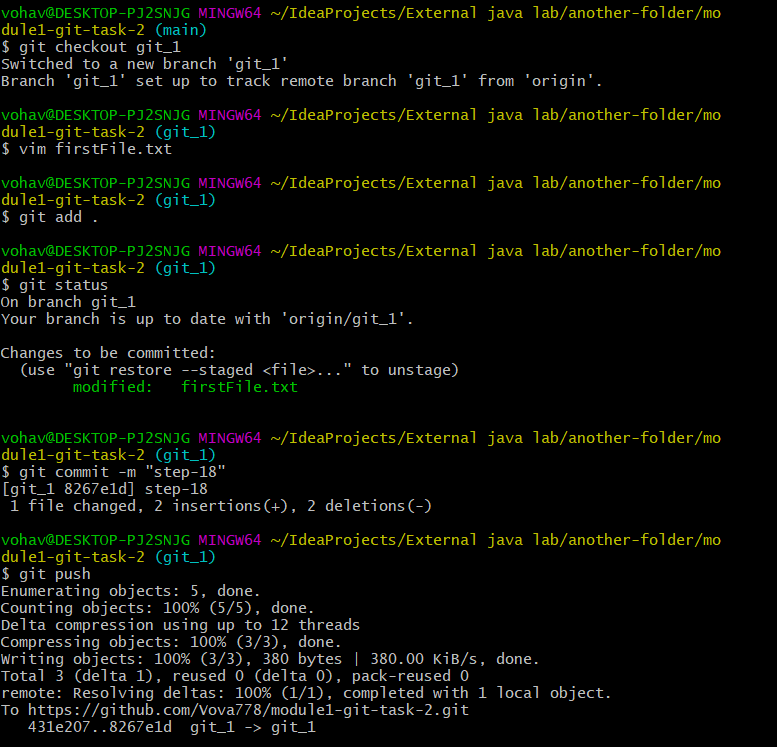
16. Synchronize *git\_1* and *git\_2* with a remote repository.



17. clone your project to another folder.



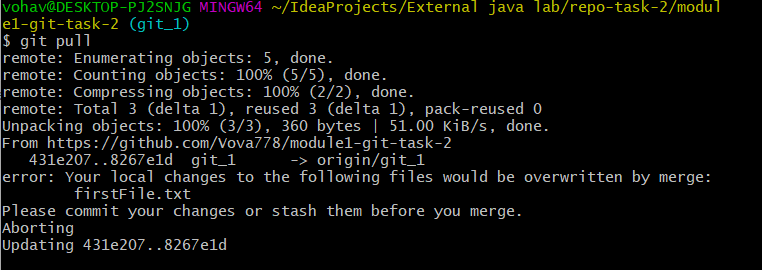
18. folder2: *git\_1*: Change two lines in *firstFile.txt*. commit + push.

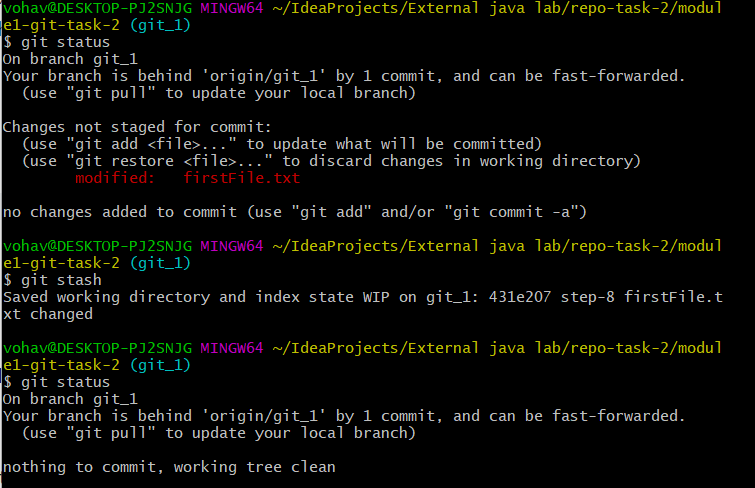


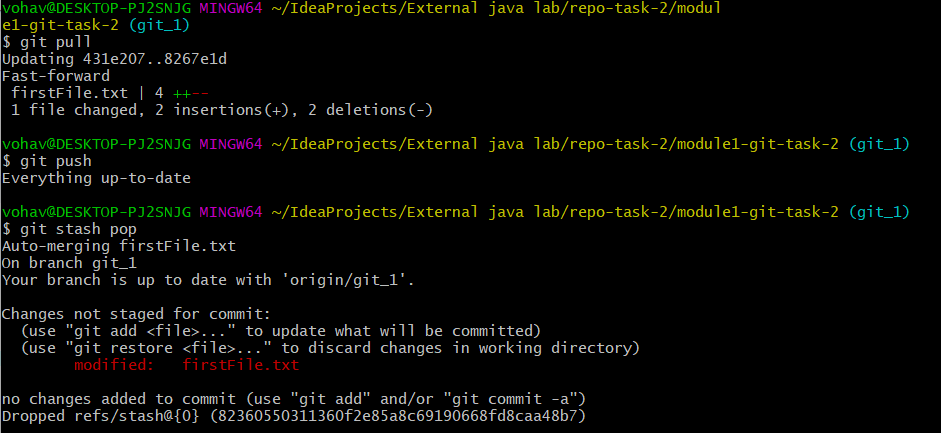
19. folder1: *git\_1*: Change another two lines in *firstFile.txt*.

20. folder1: *git\_1*:

* + Change another line in *firstFile.txt* (not the same as in 18, 19).
  + merge changes from Step 18 (pull) without committing changes from Step 19 and any additional commits.
  + push without commit changes.
  + Return to local state of Step 19. (stash)





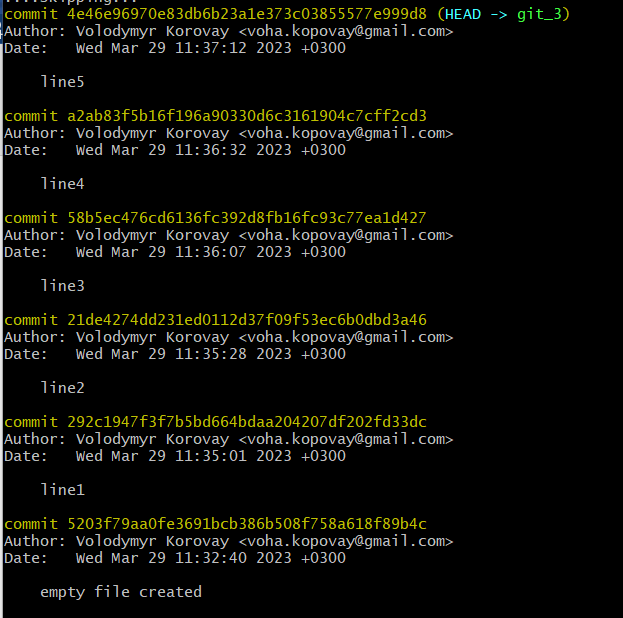


#### **Part 2**

For this task you should learn how to use interactive rebase (rebase -i), thus other ways of achieving the same are prohibited.

Show the following steps to your mentor during the demo:

1. Create *git\_3* branch from *git\_task*. Checkout to *git\_3*.
2. Add new empty file *doubtingFile.txt* and commit it.
3. Add a line to a file and commit changes. Do it 5 times. You should end up with 5 lines in a file and 6 commits: 1 for creating an empty file and 5 for adding a line.
4. Check you log and copy it somewhere.



5. Launch interactive rebase for 5 last commits, squash all the latest commits into the first one. Reword first commit. You should end up with 2 commits: one for creating an empty file and the second for adding 5 lines. Second commit should have a new commit message.

