1) Create branch “HW1” locally – this will be a branch for your first homework.

```
> git branch HW1
```

2) Checkout this branch, create “HW1” directory with three files “hw1.txt”, “test_revert.txt” and “test_revert_merge.txt” inside (you have examples in your Google Drive):

```
> git checkout HW1
# files created manually
```

3) Add and commit your changes.

```
> git add .
> git commit -m 'Creating .txt files.'
```

4) Push this branch to the repo. Be sure to indicate the correct name of your branch. When pushing a new branch that exists only locally, remember that you have to also indicate ‘-u’ flag standing for ‘set upstream’ to set up the remote tracking for your local branch.

```
> git push -u origin HW1
```

5) Now create branch “testing” but do not checkout it.

```
> git branch testing
```

6) Change “hw1.txt” on “HW1”

```
# done manually
```

7) Add, commit and push your changes to “HW1” once again.

```
> git add .
> git commit -m 'Change in hw1.txt'
> git push origin HW1
```

8) Now checkout “testing” branch and change “test_revert.txt”:

```
> git checkout testing
# changed manually
```

9) Add and commit your changes.

```
> git add .
> git commit -m 'Change in test_revert.txt'
```

10) Push this branch to the repo (similar to p.3).

```
> git push -u origin testing
```

11) Now you have two branches with two different changes to your files. To make both appear in “HW1” you have to merge branches. Checkout to “HW1” and merge “testing” into

“HW1”. Commit and push this change (into the “HW1” branch). Describe your repo state at this point in your report.

```
> git checkout HW1
> git merge testing
    Merge made by the 'ort' strategy.
     HW1/test_revert.txt | 4 +++-
     1 file changed, 3 insertions(+), 1 deletion(-)
>git push origin HW1

# At this point, branch "testing" has change in test_revert.txt, and branch "HW1" has
change in both hw1.txt and test_revert.txt.
```

12) Next, assume you want to retract your merge commit as your forgot to also change “test_revert_merge.txt” prior to merging - apply git revert indicating the parent commit to keep. Now you’ve reached the state we had discussed at the lecture (Fig.1).

```
> git revert -m 1 HEAD
    [HW1 fff46dc] Revert "Merge branch 'testing' into HW1"
     1 file changed, 1 insertion(+), 3 deletions(-)
```

13) Checkout back to “testing”. Change the file “test_revert_merge.txt”:

```
> git checkout testing
# change made manually
```

14) Now all the changes are done and we are ready to merge. Add, commit and push changes (into “testing” branch).

```
> git add .
> git commit -m 'Change in test_revert_merge.txt'
> git push -u origin testing
```

15) Checkout “HW1” and try to merge “testing” into “HW1” again. Have you succeeded? What was expected and what has happened instead? Describe in the report and explain Git’s behavior.

```
> git checkout HW1
> git merge testing -m 'Merging testing into HW1.'
    Merge made by the 'ort' strategy.
     HW1/test_revert_merge.txt | 4 +++-
     1 file changed, 3 insertions(+), 1 deletion(-)

# I expected that all 3 files in branch "HW1" will contain a change, but only hw1.txt
and test_revert_merge.txt have changes, while test_revert.txt has not. It seems like
after reverting merge the next merge picked up only the change that was made after
first merge, while I expected that all differences will be merged.
```

16) Now your task is to find a workaround for this issue (preferably, using revert command) – your goal is to finally combine “testing” with “HW1” accounting for the changes that have been made to all the files. At the end you are expected to delete “testing” branch. 17) When finished, you should provide a report (./HW1/git_report.pdf) documenting your steps and the commands you used and containing

the answer to the question from p.13. Your final “HW1” branch should contain report and three .txt files with a proper content.

```
#On the first try I created some real mess, but this time the following solution
worked properly:
> git revert -m 1 fff46dc428d71ff17e56d927a5d5322a90679a49
    [HW1 87b11cc] Reapply "Merge branch 'testing' into HW1"
     1 file changed, 3 insertions(+), 1 deletion(-)
# where commit ID is a initial revert commit found in git log.
# now all 3 files seem to have a change.
```

18) Create pull request to merge “HW1” into the master branch

Done :)