cphbusiness

git exercise 1

This exercise is meant to teach you basic skills regarding git and github. This exercise is meant to be done in pairs (but it is possible to do most of it individually)

prerequisites

You need to have git installed so you can use git bash. Git bash is where we will do all the work and exercises about git.

You also need to have a github account where you can create remote repositories

part 1: getting started

(If you do this exercise individually just work with a single github repository and clone it to 2 different locations locally)

1. Create each a new repo on github.
2. Add each other as collaborators.
3. Clone your own project in a folder locally.
4. In your own repo - add a .gitignore file to exclude all netbeans and java specific files (use gitignore.io to create the content). Also exclude .docx files
5. In your own repo add a README.md file describing this exercise
6. Add both files to index
7. Make a commit with a message about what the committed files are
8. Push commit to github (reference to remote should already be set when you cloned the project)
9. Now clone each others project.
10. Add a word file and a normal text file to the newly cloned project – add –commit-push.
11. Check what you can see on github.
12. Create a new text file save and add (but do not commit)
13. Use git diff to see difference between working area and last commit.
14. Now undo all changes since last commit (check that the file has gone)
15. Create a netbeans project in the repo and push it.
16. Try both creating a Person.java file with a different toString() method, commit and push to upstream (parent commit on github).
17. The last person to push will not be allowed
18. Use git fetch and git diff to see difference between local commit and remote tracking branch (in this case origin/master) – hint: origin is the default hostname and master is the default first branch name.
19. Person 2 must (after fetch) do a merge (git merge).
20. Solve the conflict by opening the Person.java file together and find out how it should look - then after changing the file – do add and commit (git add solves the conflict by accepting that the added/indexed file is the solution to the merge conflict – no matter how the added file looks!)

cphbusiness

part 2:

For this part you only need one repo, so choose one of the two you made in part 1.

With repo from part 1 containing a netbeans project do the following:

1. Both fetch and merge latest changes from origin/master. Then both (locally) add a few java classes in netbeans with the same names but different implementation in netbeans and add – commit. Push to upstream and solve the conflicts. (Try solving the conflicts without talking to each other but rather use the repos issues to have the dialog of what to keep and what to chuck.
2. Change the files and commit again
3. Change 2 files and add to index.
4. Using git - Revert back one file so it looks like it did before changing it. (like it looks in the last commit).
5. Change a file (without adding it to index) and then revert to how the file looks in the index.
6. Revert back the file to how it looked 2 commits ago.
7. Play around with reset, checkout and revert to see what you can do to recover files and versions.