

SE-441 Continuous Delivery and DevOps
Winter 2018-2019
Homework 3
Due On: January 31, 2017
25 points

Please answer each of the following questions as completely as you can. Please submit your response in an MS Word document or PDF. Make sure that each screen capture, copy of output, or short answer references the question and part (e.g. 3a) to which it applies.

Submit your answers to D2L by January 31, 2017. Late homework will not be accepted.

1. Download and install Git (<https://git-scm.com/downloads>). At the time of writing the current version is 2.20.1. The instructions will be different for Mac and Linux users, so use your best judgment during the installation.
 - a) Start the installation process. When you are on the “Adjusting your PATH environment” screen, make sure that you select the option that says “Use Git and optional Unix tools from the Windows Command Prompt”. You should be able to keep all of the other defaults.
 - b) Verify that git has been installed correctly by running the following command from within the Git Bash application to verify that the current version is 2.20.1:
`git --version`
2. If you don’t already have one, setup a Github (<https://github.com/>) account. The accounts are free and you’ll be using them for your assignments. *Personal opinion: you should be using them for all of your school-related work anyway.*
3. Log in to Github. Follow the link below and accept the assignment:
`https://classroom.github.com/a/yoiJN90m`
Accepting the link will create a new repository in the classroom called:
`hw3-<your account username>`
You may not see this repository in your normal Github account, but you will be able to access it at the URL:
`https://github.com/depaulcdm/hw3-<youraccountusername>`
4. [25 points] Using the repository from the previous step, please perform the following tasks:
 - (a) [1 point] On your local machine, create a new directory called `hw3<your-initials>` (e.g. `hw3cj`). Change into the directory and initialize it as a Git repository. Provide a readable screenshot of the contents of the new directory.
 - (b) [2 points] Create a new file called `README.md` and add some markdown text to it. Add that file to your staging area, and commit your changes to your local repository. Make several other commits to the `README.md`. Capture the output of the `git log`.
 - (c) [2 points] Push your local repository to Github:
`git remote add origin ↵`
`https://github.com/depaulcdm/hw3-<your git account name>.git`
`git push --set-upstream origin master`

- (d) [1 point] List the contents of the `.git/refs/heads` directory and capture a readable screenshot.
- (e) [2 points] Create two branches off of the `master` branch. List the contents of the `.git/refs/heads` directory and capture a readable screenshot.
- (f) [2 points] Change to the first branch and update and commit a change to the `README.md` file. Merge that change to `master`. Review the history using `git log` and capture a readable screenshot of the output.
- (g) [2 points] Change to the second branch and update the same line in the `README.md` as you did in step g. Commit your change and attempt to merge to `master`. Resolve any merge conflicts and complete the merge. Push your `master` branch to Github.
- (h) [2 points] From the Github 'Code' page, Then click the button containing the partial SHA1 hash to the right of the commit message. Capture a readable screenshot of the difference between that version of the `README.md` file and the prior version. Use a `git diff` command to produce the same thing from the command line and capture a readable screenshot of the output. Review the history using `git log` and capture a readable screenshot of the output.
- (i) [1 point] Create a new file, `<my initials>.md` (e.g. `cj.md`) and add some content. Capture the output of `git status`.
- (j) [1 point] Add the new file to your local repository, but do not commit it. Capture the output of `git status`.
- (k) [2 points] Perform a `git reset --mixed`. What was the result?
- (l) [1 point] Make another change to the file, but do not add it to the index or commit it to your repository. Capture the output of `git status`.
- (m) [3 points] Execute a `git reset` and capture the output of `git status`. What was the affect of the reset on your index? What was the effect of the reset on the local file?
- (n) [3 points] Execute a hard reset: `git reset --hard`. What was the affect of the reset on the index? What was the effect of the reset on the local file?