**Exercise 05**

This assignment focuses on using Git for version control. You should do this lab from virtual machine that you have access to.

Git is a fairly complex tool that can be used in many different ways. We will show you common ways you might use git in a course or for your own projects. But if you run into problems with Git, be aware that doing a web search for answers could lead you to a solution that refers to a different problem than the one you have.

We assume you will have finished any setup required from exercise 04.

**Task 0: Getting ready to use Git and the GitLab service**

1. **Log on to GitLab** – You have been given access to <https://gitlab.cranecloud.io/> for the next 1 year, and accounts provisioned accordingly. Please use the same credentials as the virtual machine you are working with, and ensure that the LDAP login option is selected.

**Task 1: Contribute to a shared Git repository for FAANG on GitLab**

It’s your second week at FAANG and you’re starting to feel more comfortable. The frontend team needs help with the website, and wants to get you integrated into their version-control workflow using git. The purpose of this task is to get practice creating and merging feature branches. You should have been given access to the shared repository described below. **If you are having trouble cloning or pushing to the repository due to a permissions problem, you may need to double check your email for an invitation to the repository.**

All of the students in BSE 4105 have recently been hired by FAANG and the frontend team wants to build a staff page that features all of the incredible staff at the company, for this task, you will be adding yourself to that page!

**First, Clone a copy of the remote repo** – On your VM machine ( cd into the directory where you would like to create your local copy of the repo and execute this command☺

git clone [git@gitlab.cranecloud.io:2024-bse-4105/faang.git](mailto:git@gitlab.cranecloud.io:2024-bse-4105/faang.git)

The purpose of this task is to get practice creating and merging feature branches.

**Problem 1** Update your local repository to contain the latest main

Since this repo is shared by so many people, we want to keep the main branch clean, never pushing directly to it, but instead working on a feature branch and eventually opening a merge request on gitlab. First, make sure that your current branch is on main by running

**git branch**

You should see the branch name main with an asterisk (\*) next to it, indicating that main is your current branch. If you are **not** on the main branch, you should switch to it with: **git checkout -b main**

\* main

Then, to update your local copy of main to match main on the remote repository, type

**git pull origin main**

If any changes have been made since you first cloned the repo, you should see output that looks something like this, followed by a list of updates made to the repository that you’re pulling down.

From gitlab.cranecloud.io/2024-bse-4105/faang

\* branch main -> FETCH\_HEAD

6745c4b..6c8d66e main -> origin/main

Updating 6745c4b..6c8d66e

Fast-forward

**Problem 2: Create a new branch for your additions to the staff page**

Your work on the staff page should be on a feature branch, separate from the main branch. Run

git checkout -b <your-username>\_staff\_page

**Problem 3: Add your bio to the staff page**

From the faang directory open up Staff.java in your favorite text editor. Staff.java is a simple program that contains one method for each employee at FAANG. Create a method, which **must be named after your git username**,

Where is my username?

that returns an Employee object that represents you. An Employee has a bio, picture and job title. In addition, you should add a photo that represents you to the images/staff directory. It doesn’t need to be an actual photo of you, but something that you feel like represents you! Simply include the name of the file (not including the images/staff prefix) as part of your Employee object. Be sure that Staff.java still compiles after editing!

The front-end team also wants you to add a new product for faang to sell, open Products.java, and add a product to the file. A product object returns the name of the product, a description, a link to an image of the product, and the price of the product. Once you have added your product make sure to put an image of your product in images/products!

**Problem 4: Push your change to the remote repo**

Use the appropriate commands to commit your changes to your branch and push to the remote repo. You should be able to see your commits in Gitlab using the following link

https://gitlab.cranecloud.io/2024-bse-4105/faang/tree/YOURUSERNAME\_staff\_page

Alternatively, you can navigate there by going to the repository main page <https://gitlab.cranecloud.io/2024-bse-4105/faang> and clicking on “Branch” at the top of the page. Here you will be able to see all of the branches for this repository, including yours.

**Problem 5: Make a merge request on Gitlab**

When you’ve gotten your work in a completed state, you will typically have another teammate review your code before merging your changes in the main branch. In Gitlab, you can create a “merge request”, which allows a teammate to review, comment on and approve your work. Create a merge request for your feature branch in Gitlab. You can do so by selecting “Merge Requests” on the left side-pane (the icon with the arrow), clicking on “New Merge Request” and then selecting your branch in the dropdown under “Select source branch”. Your target branch should be main. Click “Compare branches and continue”. Give your merge request a title without the WIP: prefix and write a description of your changes. Leave the other dropdowns and checkboxes blank and submit your merge request.

**Problem 6: Leave a comment on your merge request**

After creating a merge request and requesting a review from your teammate(s), they can leave comments on your changes. You can leave comments on your own merge requests, as well. Leave at least one comment on a change in your merge request. It can say whatever you’d like.

**Problem 7: Merge your branch from gitlab**

Let’s incorporate your changes into main so it can make it on the main website! The merge request that you created in Gitlab should have a Merge button at the top of the page. Click it. Your changes should now be in main! If the merge button is disabled because your branch has merge conflicts with main, use the resolve conflicts button to resolve the merge conflicts from Gitlab then proceed to merging to main.

**Warning**

Make sure you don’t accidentally delete other people’s changes when resolving any conflicts

**Problem 8: Turn In**

You will submit homework5.sh to MUELE, this file contains one function called print\_username in which you will **leave the echo command** but replace the text with your gitlab username. You can find this by going to gitlab, and clicking on the profile icon in the top right, your username is the text following the @ sign. (It will usually be your username i.e. mine is “amwotil”).

**Caution**

Do not remove the echo from this command!! But only for this command!

You can get the homework5.sh file from MUELE

You should be able to edit this function and submit:

function print\_username {

echo "your gitlab username here"

}

(Optional) Task 3: For fun!

Compile all java files in the repository’s root directory and run the GenerateSite program. This will produce an index.html file containing a list of all of the products our company sells and the staff who work there, as well as an html file for each product and employee. To view the website you have a few options.

1. If you’re working locally you can navigate to the faang directory using Finder (macOS) or File Explorer (Windows) and double click on the index.html file. This should open the website in your default browser.
2. If you’re working on the VM you will need to copy your files over to your local machine to view them. You can do this with the following command:

Now that this is copied over, you can view them in a web browser using part a).

1. If you’re working on the VM you can use the File Manager application to navigate to the faang directory and double click on the index.html file, which should open the site in Firefox.

*Note*: Compiling .java files will generate .class files and GenerateProducts.java will generate several .html files. These should NOT be included in the repository. A git repository should only contain the source code needed to generate all necessary artifacts, but not the artifacts themselves. Every git repository has a .gitignore file in the toplevel directory that contains the names of files should be ignored by git (for example, these files should not appear in git status or git add). If you inspect the .gitignore file for our repository you will see that .class and .html files are listed. **DO NOT MODIFY THE .gitignore FILE.**

Tips[¶](https://courses.cs.washington.edu/courses/cse391/22au/homework/hw5/#tips)

* If you ever run the git status command and see the output "Your branch is ahead of 'origin/main' by N commits", then this means that you have local commits that have not been pushed to the remote repository yet.
* Git will not allow you to pull updates into your repository while you have unstaged changes. Therefore, you must stage (git add) and commit all of your changes before pulling.
* **DO NOT** try to go back and modify previous commits. This can get you into a huge mess if you are working on a shared project.