First, you want to create a a github account and create a fork from the github repository. If you already have an account, go ahead and fork the github repository.

Next we need to clone the your fork of the github repo into a local git repo on your machine

```
git clone https://github.com/YOUR_REPO.git
```

Now that we have the github repo as a local repo we want to set a few git variables to make pushing/pulling easier.

First you'll want to set your name and email using,

```
git config --global user.name "your_name"
```

and

```
git config --global user.email “your_email@example.com"
```

Now you’ll want to set an upstream, this will be the primary repository that you will pull from to update any local branches or repositories.

```
git remote add upstream https://github.com/ORIGINAL_OWNER/
ORIGINAL_REPOSITORY.git
```

In our case the github link is https://github.com/acpaquette/CS386_project.git.

Now that we have set the username, email, and upstream, we are ready to start using this local repo to modify the master repo. If you use the command

```
git branch
```

You will get a list of branches associated with the repository, but right now it will look something like this

```
* master
```

To start making any changes we want to create a new branch. This will be were all the changes are made in regards to the current job. To create a branch you use the command

```
git branch name
```

Were name is the name of the branch. We then want to enter this branch using

```
git checkout name
```

Were name is the name of the newly created branch. If you enter the previous git branch command, the list of branch

```
  master
* test
```

The green shows that we are now in the test branch of the local git repo. While we are in the branch we can make any changes we want and retain the original code that we had pulled from the master repo intact in master. Once we have made some changes that we want to merge into the master repo there are a few commands we need to use. The general flow will be

```
git add/rm path/to/file
```

You will probably need to use this command a few times to add all of the files into the staging area for committing. You will also be able to use

```
git checkout path/to/file
```

which will remove any changes you have made to a specific file. Once you have added or removed all files you want to merge with master, use

```
git commit -m “My commit message.”
```

This will make all of the changes you have made in the branch permanent to the branch and allow you to push the branch into your github repo of the project. To do this, you use

```
git push origin name
```

Were the name is the name of the branch you have been working in. Now that you have the changes on your github repo for the project you can create a pull request from github that will allow others to view your changes to double check that nothing you are changing will break other things in the code base. Once others have reviewed your PR, and you have made any changes to files in the PR someone else in the group will merge it into the master code base. Now you have made a successful PR and have made various changed within the master code base.