Using GitHub

1. Do **not** share your `github` user name with people who are not your partners. I will be looking at repositories to make sure people do not copy. Your entire team with receive a 0 for copied work.
2. Use full URLs for any source work. Do not say `\\stackoverflow.com` or `\\kernel.org`. Use the full URL to exactly the page you got the information from.
3. Create a new account on `https://github.com/` or use an existing one.
4. One partner must make a repository for Homework 05. This repository must remain accessable until you receive a grade for Homework 05. The repository must be public.

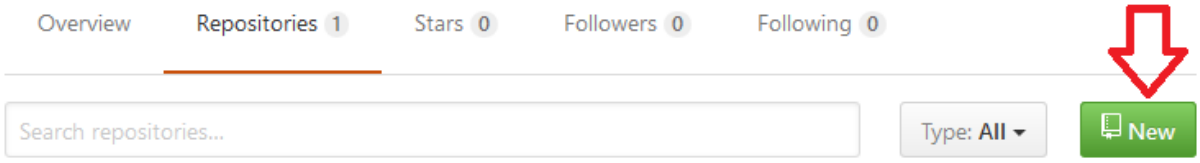

Figure 1: Create a new repository for Homework 05.

5. The partner that made the repository must give access to this repository to other partners. You allow other users to access the repository via collaborators: `https://github.com/systemsprogramming/cse384/settings/collaboration` (where systemsprogramming is your user name and cse384 is your repository name)

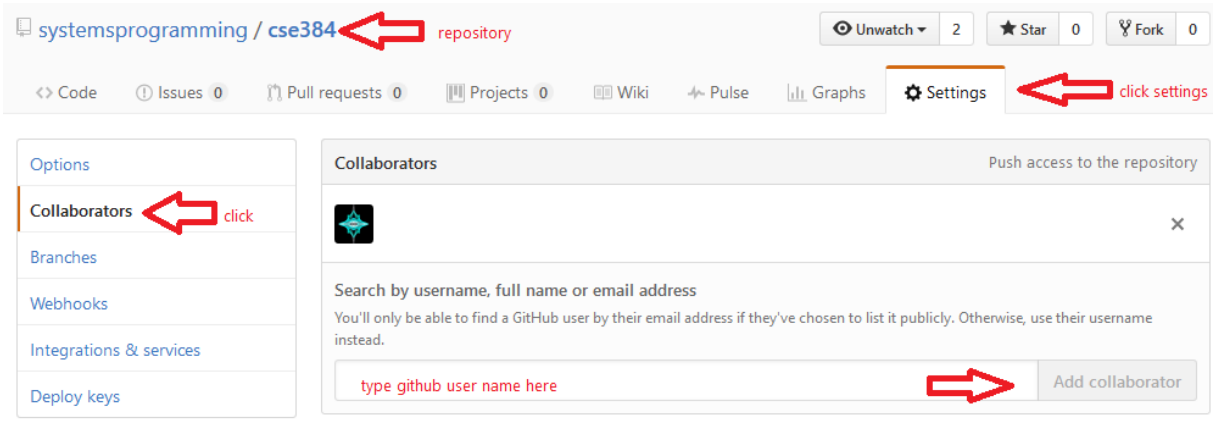

Figure 2: Adding collaborators.

6. The other partners must accept the invitation in the email.

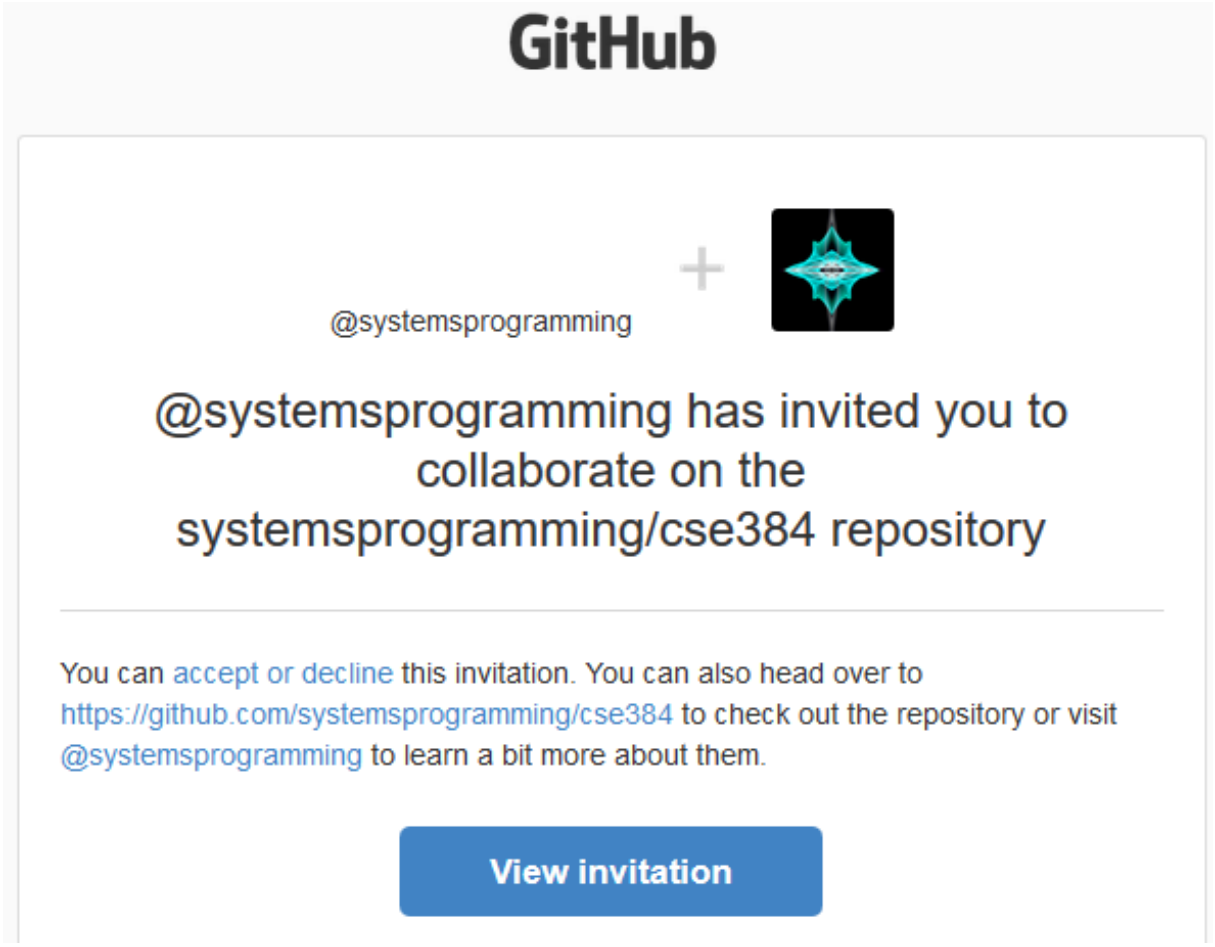

Figure 3: What the invitation looks like.

7. This is the central repository that you will use with your `git` client. You will use the URL to clone the central repository to a local repository.
8. Open your VM and install `git`: `sudo apt-get install git`.
9. Clone your `git` repository locally:
   `git clone https://github.com/systemsprogramming/cse384/`[1]
10. Verify that you have a new directory that is the name of the repository. This is the directory in which you will do all of your work for Homework 05.
11. Set up your `git` client to work with github:
    ```
    git config --global user.email "spam@example.com"
    git config --global user.name "YOUR-GITHUB-USER-NAME"
    ```

[1] Throughout the rest of this guide, I will be using *systemsprogramming* as my github user name and *cse384* as my repository. You must use the user name in the URL for whatever partner created the repository on github. You must use your own user name and password when you set up `git` and when you commit to the central repository (`push`).

12. If you know what file you want to change (or have already made changes to it), add the file to your commit list: `git add THE_NAME_OF_THE_FILE`

13. Once you have edited a file (and added it), commit your changes with `git`: `git commit -m "this is a description of what I changed"`

14. Once you have made sufficient edits to deem it necessary to share your beautiful creation with the rest of the world, `push` your changes to the mainline/central repository:
    `git push`
    Enter your github credentials

15. Remember to `pull` changes from the central repository every so often so you don't get out of sync with your partners.

16. The longer you take to `push` your changes, the larger the risk of having collisions when you merge. This means that you and another partner edited the same file at the same spot and `git` can't figure out what the file should be. This is a pain. So **always** `push` **often**.

17. There are a lot of `git` graphical clients if you are so inclined: `https://git-scm.com/download/gui/linux`. I think I've used `giggle` before, but I can't remember.