

Using Git & GitHub for Collaborative Development

Recitation 1, CMU 17-313, Fall 2022

Overview: After this recitation, the students should be able to use Git and GitHub for their teamwork in this course and other projects. Students should know about the GitFlow workflow and how to collaborate with other developers in their projects.

Part 1: Git Basics & Setting up GitHub

Activity 1.1: Forking and cloning a repo

- Fork [this repository](#)
- Login into your Linux VM (using VSCode or a terminal)
 - `ssh <your-andrewid>@17313-<yourandrewid>.qatar.cmu.edu`
- Do the first-time setup of your git configuration
 - `git config --global user.name "Your Name"`
 - `git config --global user.email <your-github-email>`
- Clone the repo you just forked on your VM
 - Clone the repo in your home directory using the command line
`git clone git@github.com:<your-github-id>/github-setup.git`

Activity 1.2: Setting up GitHub with SSH to push changes

(from <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>)

- In your VM, run this command in a terminal, substituting your GitHub email address.
`ssh-keygen -t ed25519 -C "your_email@example.com"`
- Add the newly created SSH key to your GitHub account. Follow these [steps](#)
- Add your Andrew ID to the README.md file of the repo
- Add, commit, and push the changes
 - `cd ~/github-setup`
 - `git add README.md`
 - `git commit -m 'added andrewid'`
 - `git push`

Part 2: GitHub for Collaborative Development

Activity 2.1: Forking and cloning a repo

- You and your partner forks [this repository](#)
- Add the other person as a collaborator on GitHub
- Under the settings tab, check the “Issues” checkbox to enable the issues pane
- Clone the repository on your VM (inside your home directory)

```
git clone git@github.com:<your-github-id>/github-recitation.git
```

Activity 2.2: Setting up Kanban board and creating issues

- On GitHub, under the project tab, one person creates a new project. Select “Automated Kanban” or “Board” as the template
- Create two issues, one for each bug found in the fib.py file.
- Add the two issues to the Kanban to-do board and assign both issues, one for each person
- Move the issues into To-do!

Activity 2.3: Branching, committing, and pushing

- In your VM, create a new branch with a relevant name to the issue you are addressing (e.g. “fix-header-sizing-issue”, “fix-multiple-dialog-bug”, “add-calendarfeature”)
- Fix the issue/bug that you were assigned, then add and commit the changes
 - Use `git commit -m 'meaningful message'`
 - Commit messages should start with a verb and what it does to the codebase (e.g., “Fix failing CompositeTestCase”, “Fix issue #21”)
- Push the branch to remote
 - `git push -u origin <branch-name>`

Activity 2.4: Writing pull requests

- On GitHub, create a pull request to merge the changes from the branch you have just pushed to the main/master branch.
 - Make sure it is to the main/master of your own forked repository! By default, Github selects the original repository instead of your own forked repository
 - Name your pull request appropriately
 - In the description, describe what changes has been made to address the issue, and how has the changes been tested
- Assign the issue you created to the pull request. On the right-hand side, under Development
- Request your partner for a code review!

Activity 2.5: Doing code reviews and resolving merge conflicts

- Review your partner’s code (use the review change button). Approve changes if it looks good

- Once your PR has been reviewed, merge your changes! Resolve any merge
- Return to the board, and if you have done things right the issue should automatically move into Done