Tools for collaborative software development

Recitation 1, CMU 17-313, Fall 2020

Git & GitHub

Goal: 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.

Introduction:
- [GitHub's flow](#)
- [GitFlow](#)

Activity 1: Creating a repo and committing files
- Create a [GitHub](#) account.
- Create an empty repository.
- Clone the repository to your machine.
- Create a README.md file with your name.
- Commit and push the file.

Activity 2: Branching and merging
- Create a new branch called "develop"
- Push the new branch. If you are able to push the branch without any error, let me know.
- You should have gotten an error message with the command to correctly push the new branch. Use that command.
- On GitHub, set the "develop" branch as the default branch.
- Create a [HelloWorld.java](#) program. Commit and push the new file.
- Switch to the "master" branch.
- Merge the changes from the "develop" branch into the "master" branch. Push the changes.

Activity 3: Issues, pull requests and merge conflicts
- On GitHub, create a new issue on your repo. The issue should be about adding how many years you have been at CMU.
- On the master branch, modify the [HelloWorld.java](#) program by adding your name to the print statement. Commit and push the changes.
- On the develop branch (do not pull or merge any changes), modify the HelloWorld.java program by adding how many years you have been at CMU to the print statement. Commit and push the changes.
- On GitHub, create a pull request to merge the changes from the develop to the master branch. You should get an error about a merge conflict. If you do not get the error, let me know.

- Despite the error, you should have been able to create the pull request. Assign the issue you created to the pull request by commenting on the pull request the number of the issue (i.e., #n, where n is the number of the issue you created).
- Follow the instructions on GitHub to fix the conflict.
- Commit and push the new change. On GitHub, resolve the pull request.

Activity 4: Adding collaborators and pulling changes
- Pair up
- Add your partner as a collaborator on your repo.
- Clone your partner's repo.
- On your partner's repo, make a change to the HelloWorld.java program. Commit and push the changes.
- On your own repo, pull the changes and check the changes that your partner made.