CS 441 Software Engineering

Assignment 5

Due on Monday, May 11, 11:59pm

Individual Portion:

In this assignment, we are going to work on the GIT version control system. We will create a remote repository and two local repositories. Based on them, we will try a number of basic operations of GIT, including staging, committing, branching, merging, pushing, pulling, and tracking branches.

Please follow the instructions and finish all the steps below. During this process, you need to make **three** screenshots, answer **two** questions, and finish **one** task.

1. Installing GIT and first-time setup.

Download GIT from https://git-scm.com/downloads and install it in your computer.

After installation, open GIT command line window, and type in

git config --global user.name “YOUR NAME”

2. Setting up a remote repository and doing an initial “push”.

Create a folder named assignment5 in your computer (mkdir assignment5), and enter the assignment5 folder (cd assignment5).

Create a folder named myGitRemote.git in the assignment5 folder (mkdir myGitRemote.git).

Enter the myGitRemote.git folder (cd myGitRemote.git).

Initialize the folder as a GIT repository (git init --bare).

Check the content of the myGitRemote folder to see what a GIT repository looks like (ls).

Screenshot 1: make a screenshot that shows the content above.

Next, we are going to create a local repository and push its content to the remote repository, which is myGitRemote.git.

Go back to the assignment5 folder (cd ..).

Create another folder named myGitLocalA (mkdir myGitLocalA).

Go to the new folder (cd myGitLocalA), and initialize a GIT repository there (git init).

Open the Windows file browser, copy and paste the three Java files (attached to this assignment) into the myGitLocalA folder.

Now tracking these files with GIT (git add *), and do an initial commit (git commit -m “Initial Import”).

All the files have been saved into your local repository.

Next, add the remote repository into the list of your remote servers (git remote add origin file:///…/assignment5/myGitRemote.git).

Push your local repository to the remote repository (git push origin master).

Done!

3. Cloning the remote repository and creating a new branch.

Now let’s create another local repository, assuming that it will be used by a different developer.

Go back to the assignment5 folder (cd ..).

Clone the remote repository we created earlier (git clone file:///…/assignment5/myGitRemote.git myGitLocalB).

Go to the myGitLocalB directory (cd myGitLocalB), and check its content (ls). All three Java files should be there.

Create another branch (e.g. named development) and switch to it (git checkout -b development).

Open the file SpaceCraft.java, make some changes, save and close the file.

Check you current status (git status).

Screenshot 2: make a screenshot that shows the status.

Stage the file SpaceCraft.java to be committed (git add SpaceCraft.java).

Commit your changes (git commit -m “Developing in progress.”).

Push this branch to the server (git push origin development).

Now switch back to the master branch (git checkout master).

Question 1:

Open the file SpaceCraft.java again. Are your changes still there? Why?

4. Remote Branching and Merging.

Go back to your first local working directory (i.e., myGitLocalA).

Check the information of the remote repository first (git remote show origin).

Let’s fetch the latest data from the repository first (git fetch origin).

Now you should have two remote branches: origin/master and origin/development.

Next, we are going to create a tracking branch from the remote branch origin/development (git checkout -b development origin/development).

Question 2:

Open the file SpaceCraft.java. Do you see the changes you just made? Again, why?

Now we are going to merge the branch development to the master branch.

(git checkout master)

(git merge development)

5. Resolving Conflicts.

Task 1:

Now, you need to design a scenario where merging conflicts actually happen. Resolve your conflicts. You scenario must involve remote branches.

Screenshot 3:

Make a screenshot that shows the step that you resolved the conflicts.

Create a document that includes your three screenshots, answers to the two questions, and description of what you have done in the last task (e.g., how you created and resolved merging conflicts). Submit the document to Cougar Course before the deadline.

Group Portion:

None.