**GITHUB INTRODUCTION – Day Two**

Software developers use Git to collaborate all the time and so you will get practice with collaborating on Git today.

You and your partner should decide who is Partner A and who is Partner B in the steps below.

**Part I: Collaborating on the same repo**

1. Partner A should go to the GitHub website and click on the gitpractice repo.

2. Partner A should go to Settings – Collaborators and enter Partner B’s GitHub username or email address.

3. Partner B should check their email or click on the GitHub notification and Select “Clone”. Partner B should copy the URL that is listed.

4. In a terminal, Partner B should navigate down into the Day4CollaboratingOnGit folder. They should type “pwd” to confirm that they are there.

5. Partner B should then type “git clone” followed by the URL address copied from the above step.

It will be something like “git clone <https://....gitpracticepartnername.git/>”

6. Partner B should navigate down into this gitpractice folder.

7. Now, here comes the fun part. You and your partner will be collaborating on the same code. One of you will fill in the getNRandom function and the other person will fill in the multiplyRandom function. The final product will be a program that generates 10 random numbers and returns the product of those 10 numbers. Fill in your part of the code on your local computer now.

8. Now comes the pivotal part where we sync our codes up. When you are collaborating with someone, you always want to push before you pull or else you will get merge conflicts (meaning, the code on your local computer is not up to date with the code located on GitHub). So first, while you are inside this gitpractice directory, Partner A should type “**git pull**”.

9. \*\*\*Note\*\*\* be sure you did last the step before continuing on.

Now Partner A should type:

git add -A

git commit -m ‘did my part’

git push

10. Partner B should now complete Steps 8-9. Make sure they type “**git pull**” first!!! Then, Partner B will have Partner A’s work on their computer.

11. Now, Partner A should type “git pull” once more so that they have Partner B’s work on their local computer as well.

12. Both partners should run “python practice.py” on their local computers to verify that their combined code works.

**Part II: Switch**

Complete the same steps as above but now Partners A and B should switch roles. In the end, you will have two repos up on GitHub, one called “gitpracticeNAME1” located within your Day3 folder and one called “gitpracticeNAME2” located within your Day4 folder.

**Part III - Branching**

Often, collaborators will not be pushing to the main branch of the repo. For example, consider all of the code that goes into making the Nike website. Would Nike want one disgruntled employee to be able to push a bunch of nonsense to the main code base? Instead, employees make branches for whatever features they are currently working on, which eventually get merged into the main branch after a pull request. This is also the process by which people contribute to open source software, a super cool thing to do.

Your homework will be to follow the steps on this website <https://codeburst.io/a-step-by-step-guide-to-making-your-first-github-contribution-5302260a2940> in order to contribute to your first open resource. More specifically, if you follow the steps correctly, your name will be featured here <https://github.com/firstcontributions/first-contributions/blob/master/Contributors.md>

as soon as the administrator approves your pull request.