CPSC:480 Software Engineering Exercise 1

Complete this exercise in pairs with the student next to you. *Steps in Italics are completed on GitHub, others on local machine using git.*

Part I – Initialization

1. Open laptop with Git installed and create parent directory for projects (e.g. C:\Users\me\github).
2. *Create a public GitHub repository named "SWEF22-Ex-1-<UAid>".*
3. Open git command line and go to project parent directory.
4. Clone a local copy (“git clone https://github.com/<user>/SWEF22-Ex-1-<UAid>”) and make sure you’re in the repository directory.
5. Create file "data.txt" in local repo folder and put your name in.
6. Add to files tracked by Git (“git add data.txt” or “git add *”).
7. Create an initial commit (“git commit -a -m “Initial commit”).
8. Push change to GitHub (“git push”).
9. Set author and GitHub credentials as needed.

Part II – Collaboration

1. *Open a browser to https://github.com/<partner-username> /SWEF22-Ex-1-<partner-UAid> while logged in to your account.*
2. *Click “Fork” in top right, accept defaults, and click “Create fork”.*
3. On your machine, go back to github directory, outside of repo,
4. Create local copy of fork (“git clone https://github.com/<**your**-username>/SWEF22-Ex-1-<**partner**-UAid>”) and go in new repo.
5. Edit data.txt and add today’s date **above** partner’s name.
6. Commit and push to your fork on GitHub.
7. *Open your fork in a browser and click on “Open pull request” under “Contribute”, then “Create pull request”.*
8. *Wait for your partner to complete step 7 before starting step 9.*
9. *Go to your own repo, click “Pull requests” tab, find and open your partner’s PR, and click “Merge pull request”, then “Confirm”.*

Part III – Conflicts

1. On your local machine, "git pull" **both** repos to current state.
2. Edit data.txt in both repos' copies and add your UAid at bottom.
3. Commit & push the change **only to your fork of *partner's* repo**.
4. *Issue another PR to your partner & accept the one they issue you.*
5. *Wait for your partner to complete step 4 before starting step 6.*
6. Go to **your** repo, commit your local change and try to push.
7. Note the failure, then git pull to download the commits on GitHub that come from your partner's pull request.
8. Open data.txt and view the conflict to be resolved. Edit the file to contain your UAid, then your partner's.
9. Commit the merge and push to GitHub.

Part IV – Branching

1. Create branch in your repo named "part-4" ("git branch part-4).
2. Set the new branch as the active branch ("git checkout part-4").
3. Rename data.txt to authors.txt ("git mv data.txt authors.txt").
4. Commit the change to this branch (**do not push**).
5. Switch to main branch ("git checkout main") and note that the file is again called data.txt.
6. Add your favorite Pokémon (eg "Pikachu") at the end of data.txt.
7. Commit the change (**do not push**).
8. Merge part-4 branch ("git merge part-4").
9. Commit and push the change. Note that the file is called authors.txt *and* has the Pokémon you added while the file was called data.txt.

Part V – Navigation

1. View the list of branches and the active branch ("git branch").
2. Delete data.txt (using "git rm data.txt" will stage the deletion for the next commit, any other method will not; doesn't matter here)
3. View the changes to the current state ("git status").
4. Restore data.txt ("git restore data.txt").

Part VI – History

1. Find the commit where you added the date in Part II (“git log”) and note the first 4 characters of the hash for this commit.
2. Set your local repo state to this commit (“git checkout <hash>”).
3. Create a new branch “part-6” (“git branch part-6").
4. Add your favorite movie to data.txt. Commit but don’t push.
5. Check out the main branch.
6. Merge part-6. Note the message about conflicts. Edit the file to include the content from Part IV, followed by the movie. Commit and push.
7. *Wait for your partner to complete step 6 before starting step 8.*
8. *Go to your fork of your partner’s repo and download their changes by clicking “Sync Fork” -> “Update Branch”*
9. *Click “Insights” -> “Network” in each repo to see visualization of timeline.*

Part VII – Documentation and release

1. Create README.md file and add to Git.
2. Write a summary of what you did in this exercise in Markdown.
3. Use at least 5 different formatting or special features (see docs).
   a. 1% Extra Credit for every 5 additional features used. List the ones you claim in the README (using md list counts for 1)
4. Commit and push the change.
5. *View README on GitHub and verify formatting is correct.*
6. *Check partner’s repo to ensure you’ve both followed directions.*
7. *Add instructor mention (@Kilgallin) in README.md, commit, push*
8. *On the right side of the repo, click “create release”, click “Create a tag”, enter a tag name “submission”, and click “Publish release”.*

@mention + release will be considered exercise submission. You may make corrections until 4 PM Wednesday. Grading will only be based on following directions, approximately 3% per gradable step. Do not delete repo until grading is complete and any re-grade requests are resolved.