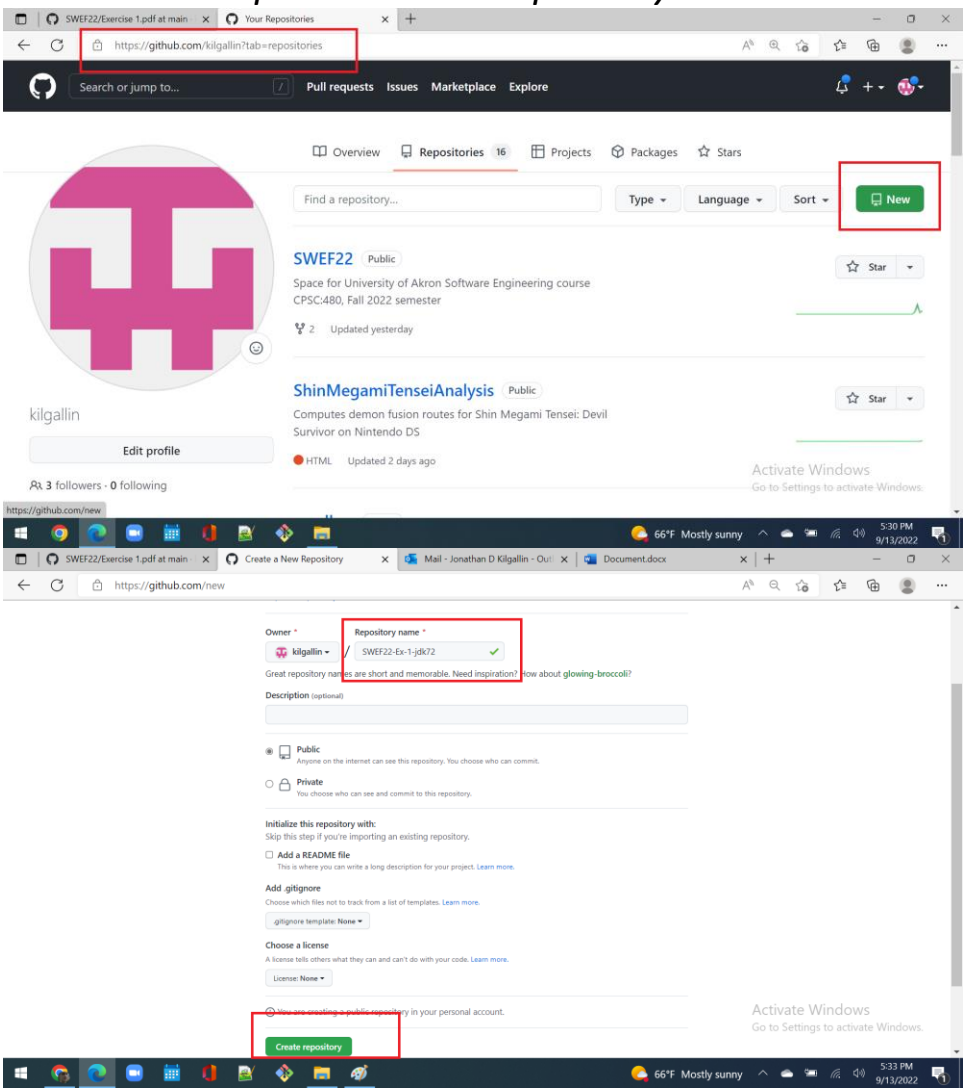
CPSC:480 Software Engineering Exercise 1

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

Note: In this walkthrough, GitHub user "kilgallin" UAid jsk72, is the primary user completing these steps, and GitHub user "JDKeyfactor" repo "SWEF22-Ex-1-otherJDK" is simulating the partner.

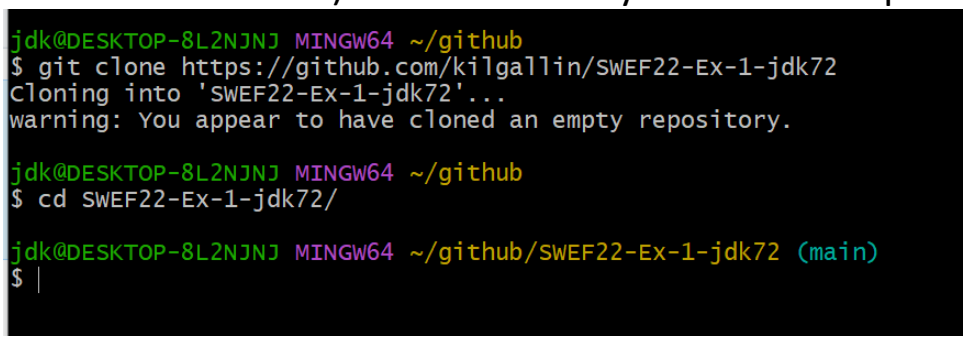
Part I – Initialization

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>".*

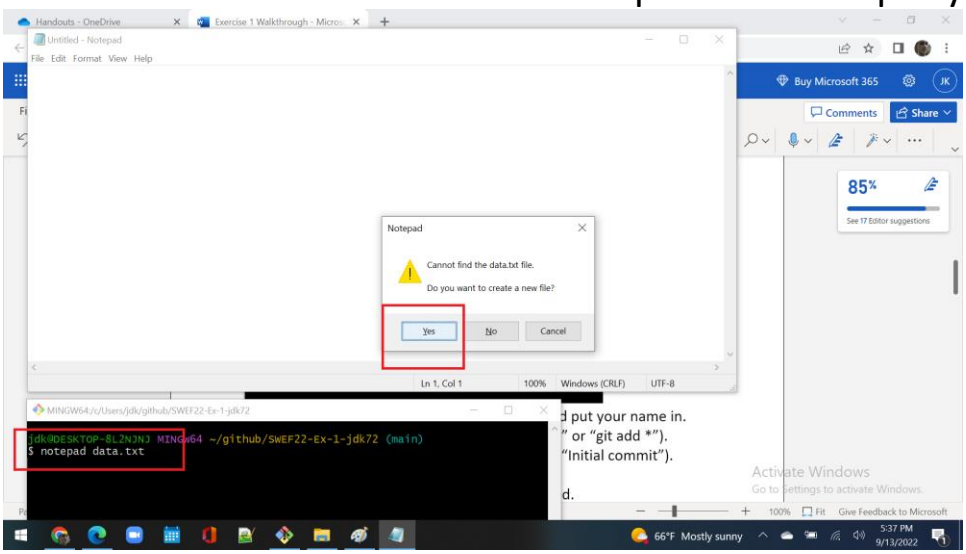


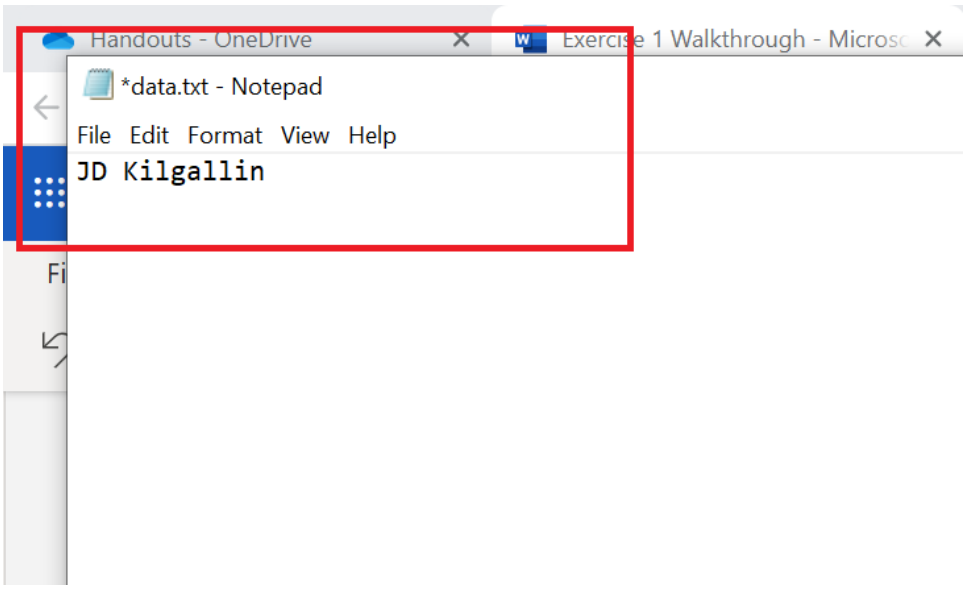
3. Open git command line and go to project parent directory.

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 *”).

MINGW64:/c/Users/jdk/github/SWEF22-Ex-1-jdk72

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ notepad data.txt

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git add data.txt

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   data.txt


jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$
```

7. Create an initial commit (“git commit -a -m “Initial commit”).

MINGW64:/c/Users/jdk/github/SWEF22-Ex-1-jdk72

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git commit -a -m "Initial commit"
[main (root-commit) bd75643] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 data.txt

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
```

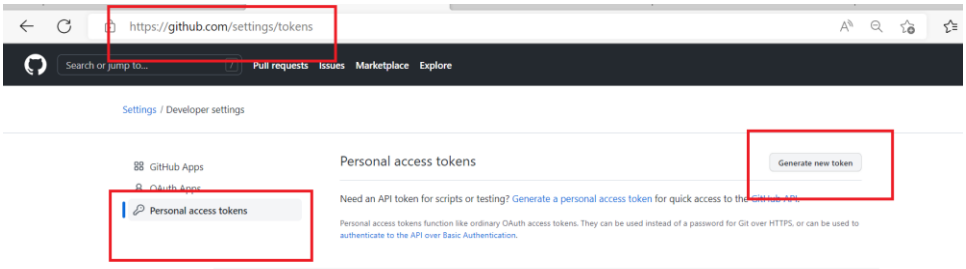
8. Push change to GitHub (“git push”).

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 225 bytes | 225.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kilgallin/SWEF22-Ex-1-jdk72
 * [new branch]      main -> main

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$
```

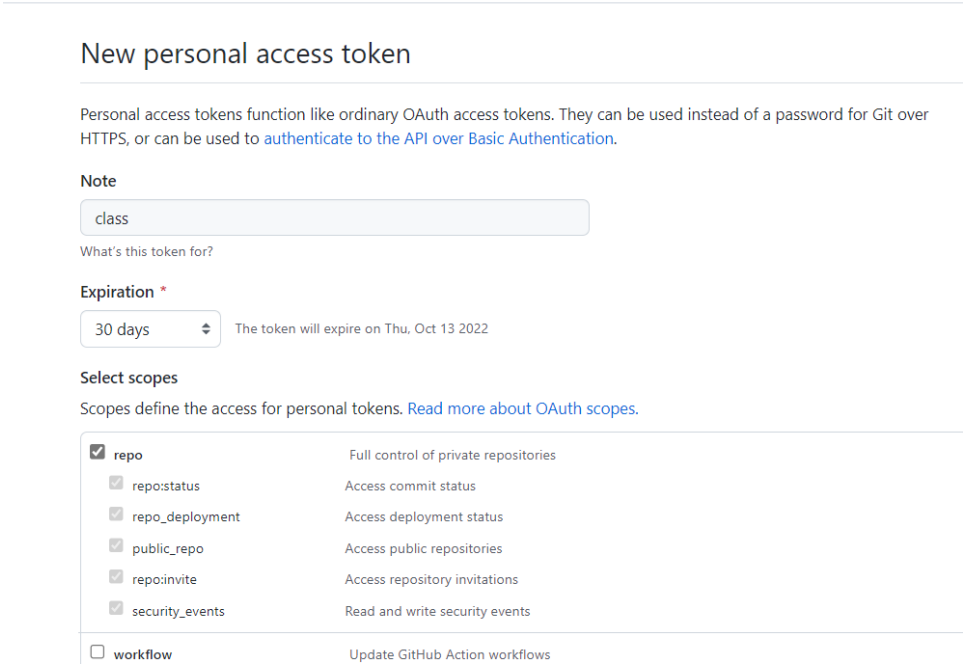
9. Set author and GitHub credentials as needed.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git config --global user.name "JDK"

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git config --global user.email
jd.kilgallin@gmail.com
```

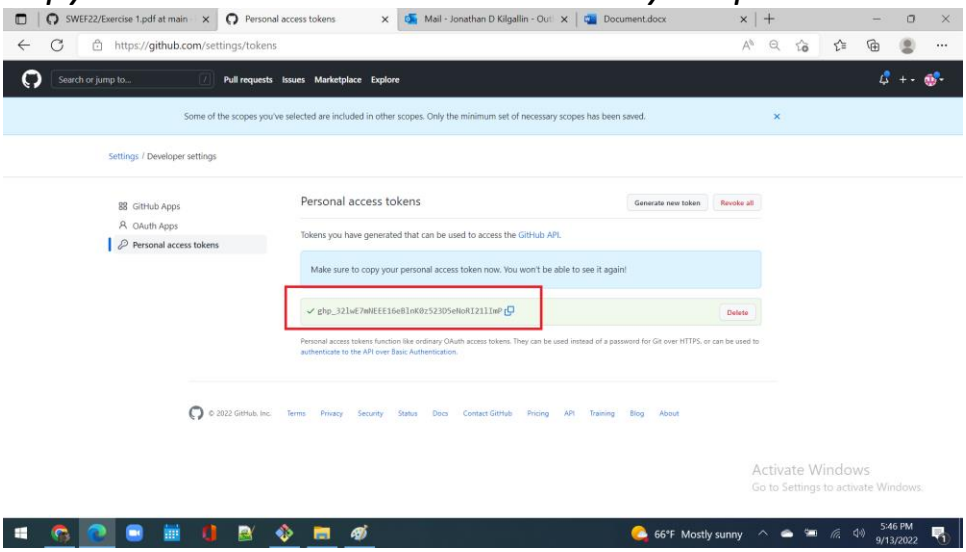
Note: If you get an error about passwords not being supported, as seems to be the case on Apple OS X, do the following:
Click your profile in the top-right of the GitHub page and go to "Settings". Find "Developer Settings" in the far bottom left. Select "Personal Access Tokens" and click "Generate new token"



Enter a name (e.g. "class") and check the box next to repo, then click "Generate Token" at the bottom.

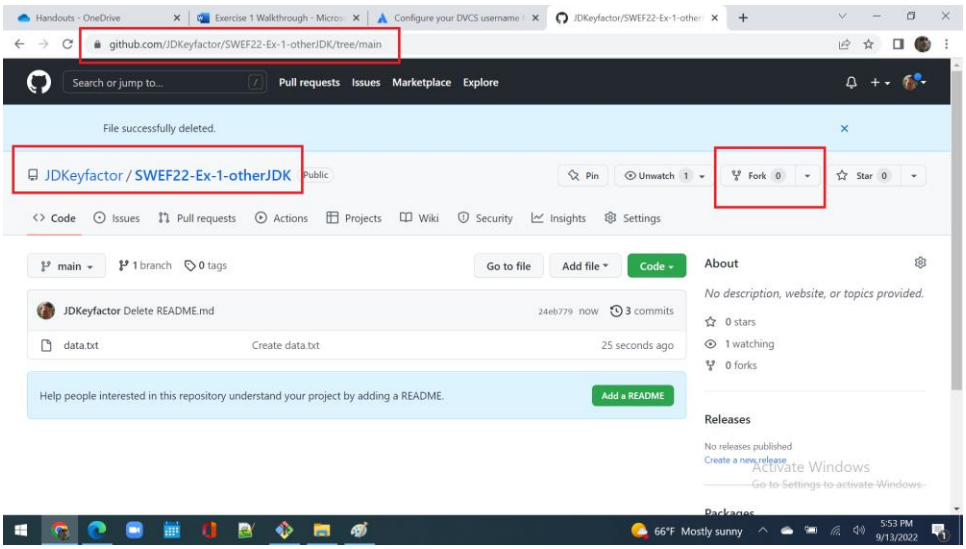


Copy the value shown and enter it as your password in Git

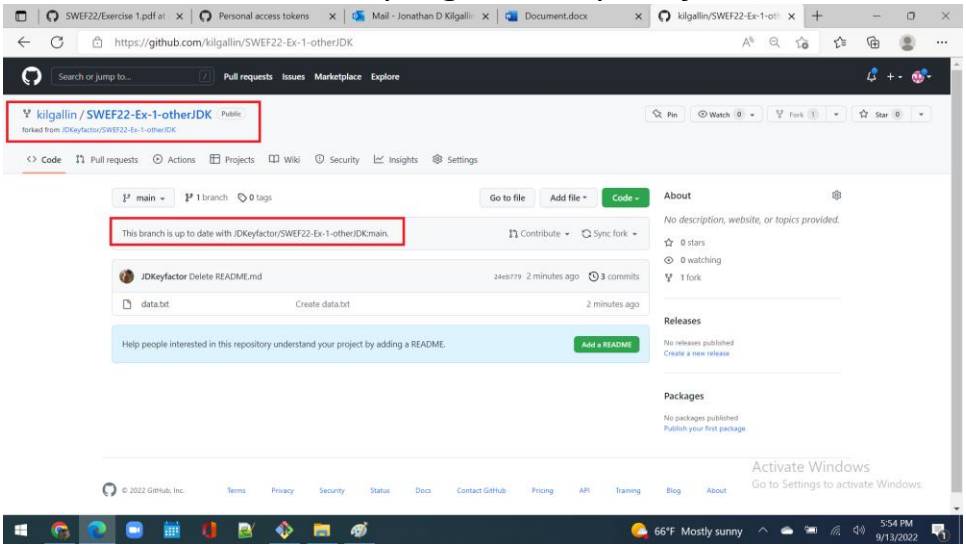


Part II – Collaboration

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

MINGW64:/c/Users/jdk/github

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ cd ..

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github
$
```

4. Create local copy of fork ("git clone https://github.com/<**your**-username>/SWEF22-Ex-1-<**partner**-UAid>") and go in new repo.

*Note: Make sure you clone **your fork**, and don't clone your partner's repo directly, as you won't be able to push.*

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github
$ git clone https://github.com/kilgallin/SWEF22-Ex-1-otherJDK
Cloning into 'SWEF22-Ex-1-otherJDK'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github
$ cd SWEF22-Ex-1-otherJDK/

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$
```

5. Edit data.txt and add today's date **above** partner's name.

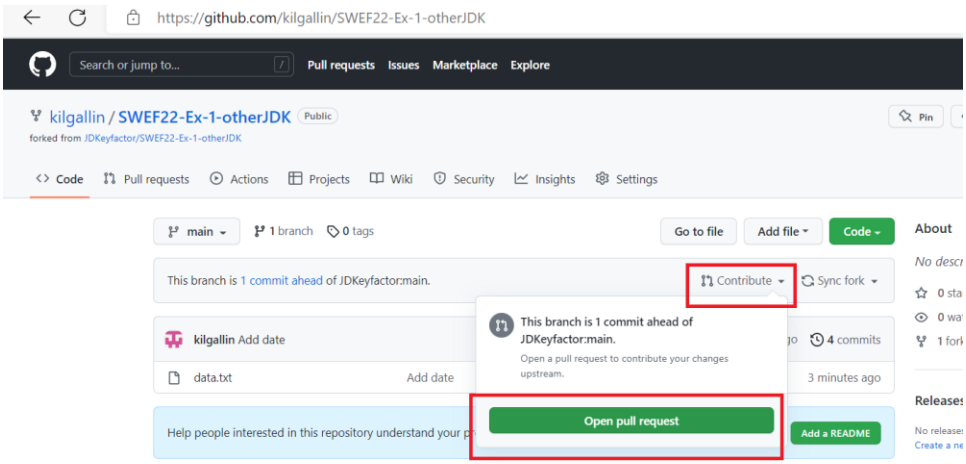
```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ notepad data.txt
```

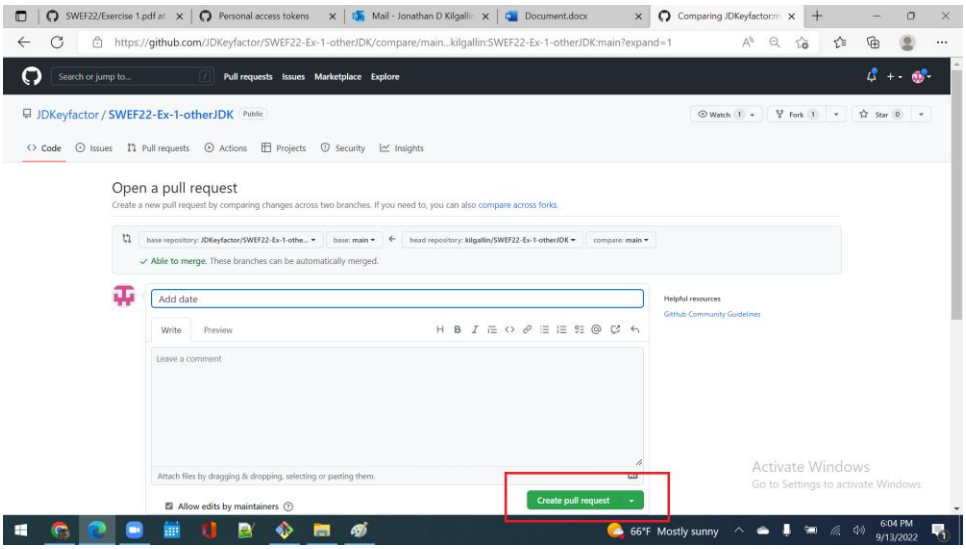
6. Commit and push to your fork on GitHub.

*Note: If you get error 403, check that you cloned **your fork***

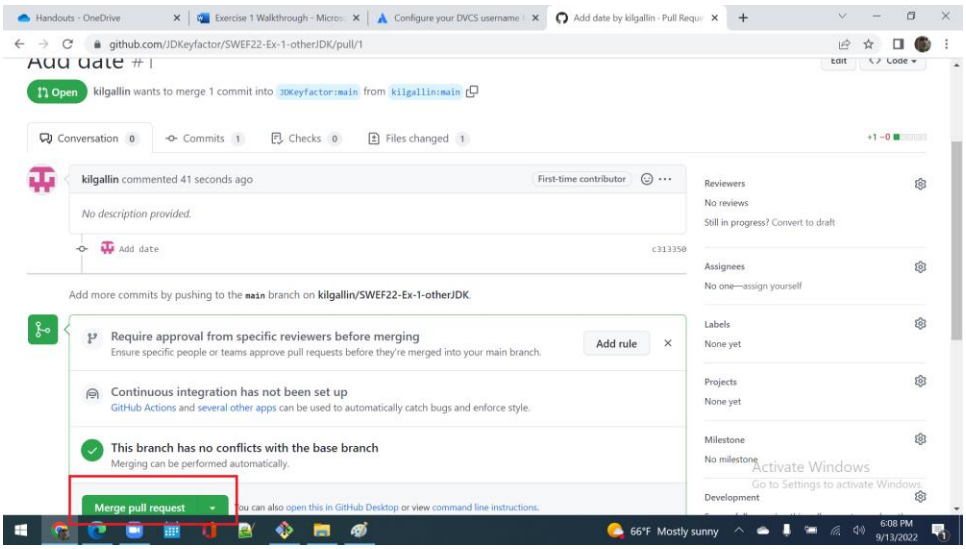
```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ git commit -a -m "Add date"
[main c313350] Add date
 1 file changed, 1 insertion(+)

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 263 bytes | 263.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kilgallin/SWEF22-Ex-1-otherJDK
   24eb779..c313350  main -> main

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$
```

7. *Open your fork in a browser and click on "Open pull request" under "Contribute", then "Create pull request".*





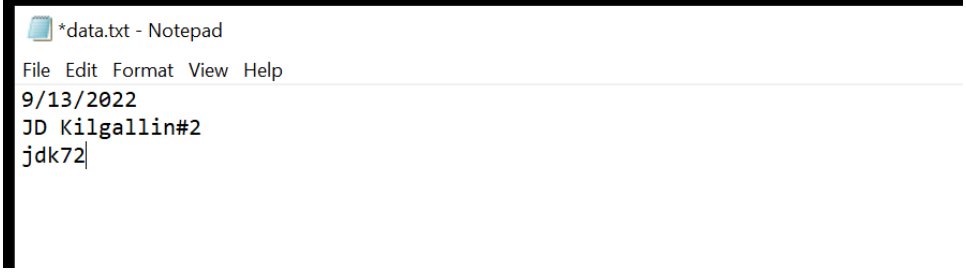
8. *Wait for your partner to complete step 7 before starting step 9.*
9. *Go to your own repo, click "Pull requests" tab, find and open your partner's PR, and click "Merge pull request", then "Confirm".*


Handouts - OneDrive
Exercise 1 Walkthrough - Micros
Configure your DVCS username
Add date by kilgallin · Pull Requ
github.com/JDKeyfactor/SWEF22-Ex-1-otherJDK/pull/1
Add date #1
Edit
Code
Open
kilgallin wants to merge 1 commit into JDKeyfactor:main from kilgallin:main
Conversation 0
Commits 1
Checks 0
Files changed 1
+1 −0
kilgallin commented 41 seconds ago
First-time contributor
No description provided.
Add date
c313350
Add more commits by pushing to the main branch on kilgallin/SWEF22-Ex-1-otherJDK.
Require approval from specific reviewers before merging
Ensure specific people or teams approve pull requests before they're merged into your main branch.
Add rule
Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.
Reviewers
No reviews
Still in progress? Convert to draft
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Activate Windows
Go to Settings to activate Windows.
66°F Mostly sunny
6:08 PM
9/13/2022

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.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ notepad data.txt
```



3. Commit & push the change **only to your fork of *partner’s* repo**.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   data.txt

no changes added to commit (use "git add" and/or "git commit -a")

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ git commit -a -m "Add uaid"
[main 9ebf3cd] Add uaid
 1 file changed, 1 insertion(+)

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 268 bytes | 268.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kilgallin/SWEF22-Ex-1-otherJDK
   c313350..9ebf3cd  main -> main

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-otherJDK (main)
$
```

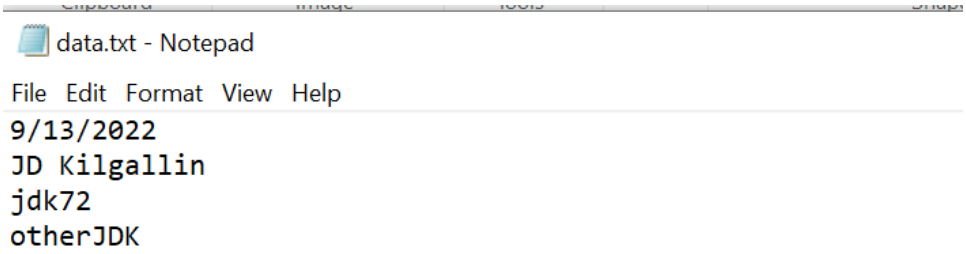
4. *Issue another PR to your partner & accept the one they issue you.*
5. *Wait for your partner to complete step 4 before starting step 6.*
6. Go to **your** repo, commit your local change and try to push.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ notepad data.txt

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git commit -a -m "Add uaid"
[main bc87d63] Add uaid
 1 file changed, 1 insertion(+)

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git push
To https://github.com/kilgallin/SWEF22-Ex-1-jdk72
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/kilgallin/SWEF22-Ex-1-jdk72'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$
```

7. Note the failure, then git pull to download the commits on GitHub that come from your partner's pull request.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git pull
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 1.24 KiB | 63.00 KiB/s, done.
From https://github.com/kilgallin/SWEF22-Ex-1-jdk72
   3e5bd80..4a178a0  main       -> origin/main
Auto-merging data.txt
CONFLICT (content): Merge conflict in data.txt
Automatic merge failed; fix conflicts and then commit the result.

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main|MERGING)
$
```

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.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main|MERGING)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 2 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   data.txt

no changes added to commit (use "git add" and/or "git commit -a")

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main|MERGING)
$ git commit -a -m "Resolve merge"
[main 056edeb] Resolve merge

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 548 bytes | 274.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kilgallin/SWEF22-Ex-1-jdk72
   4a178a0..056edeb  main -> main

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$
```

Part IV – Branching

1. Create branch in your repo named "part-4" (“git branch part-4).

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git branch part-4
```

2. Set the new branch as the active branch (“git checkout part-4").

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git checkout part-4
Switched to branch 'part-4'

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (part-4)
$ git branch
  main
* part-4
```

3. Rename data.txt to authors.txt (“git mv data.txt authors.txt”).

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (part-4)
$ git mv data.txt authors.txt
```

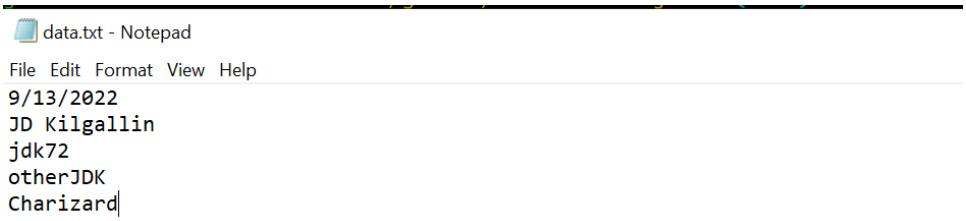
4. Commit the change to this branch (**do not push**).

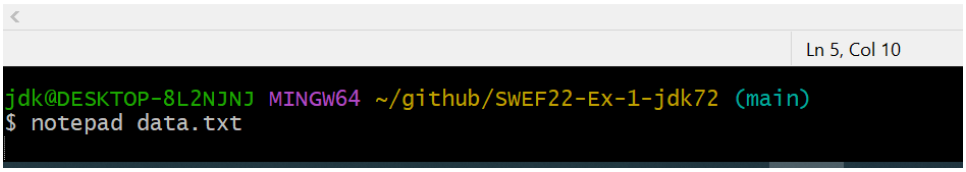
```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (part-4)
$ git commit -a -m "Rename file"
[part-4 4d3c193] Rename file
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename data.txt => authors.txt (100%)
```

5. Switch to main branch (“git checkout main”) and note that the file is again called data.txt.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (part-4)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git branch
* main
  part-4

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ ls
data.txt
```

6. Add your favorite Pokémon (eg “Pikachu”) at the end of data.txt.





7. Commit the change (**do not push**).

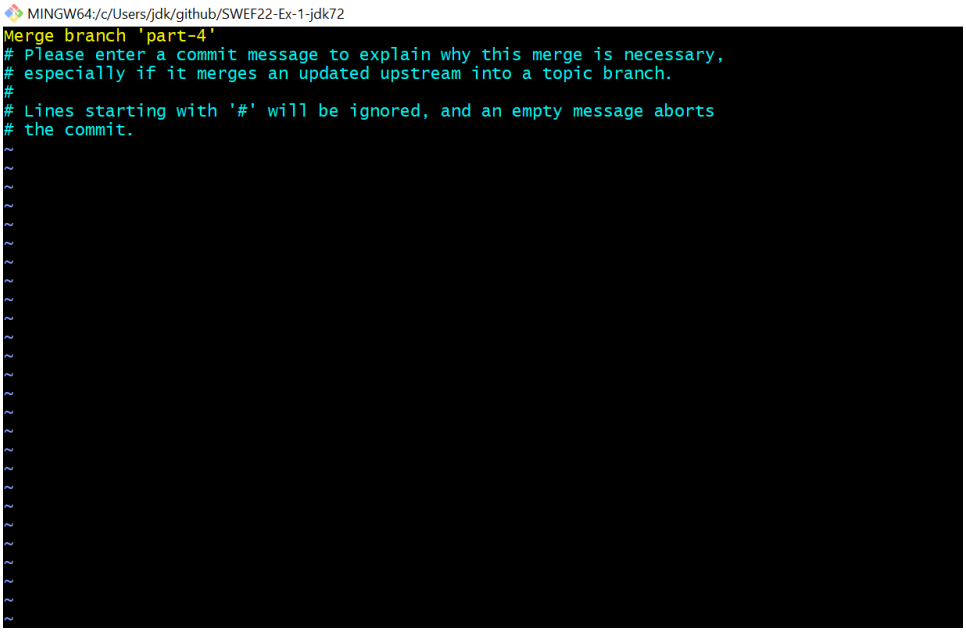
```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git commit -a -m "Add pokemon"
[main 7a7d670] Add pokemon
 1 file changed, 1 insertion(+), 1 deletion(-)
```

8. Merge part-4 branch (“git merge part-4").

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git merge part-4
Merge made by the 'ort' strategy.
 data.txt => authors.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename data.txt => authors.txt (100%)
```

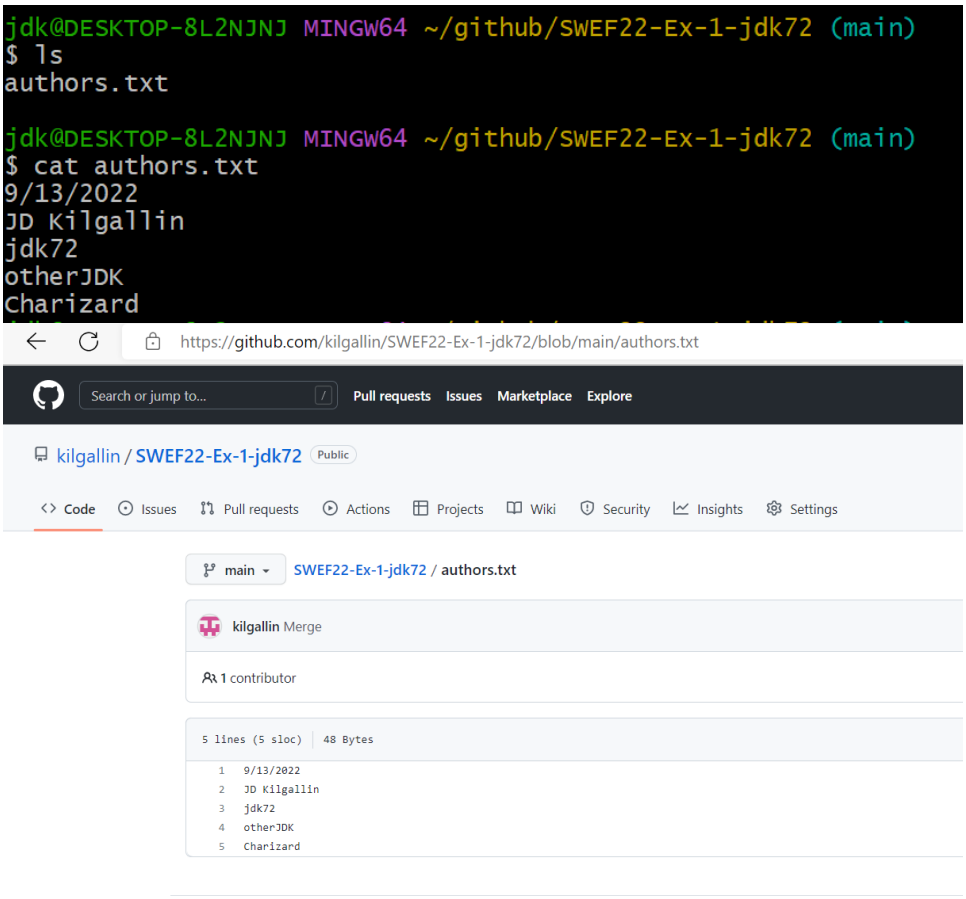
Note: Git will open its default text editor prompting you to enter a commit message for the merge. The default is likely vim, a UNIX text editor that’s popular but has a learning curve. You can accept the auto-generated commit message and close the editor by typing “:wq” and pressing enter. This should show at the bottom of the screen as you type You can read this as “:” = run command, “wq” = Write+Quit, and enter to execute. If this shows up in the editor instead, “-- INSERT --” is

probably at the bottom and you need to hit escape to switch to command mode, then type ":wq".

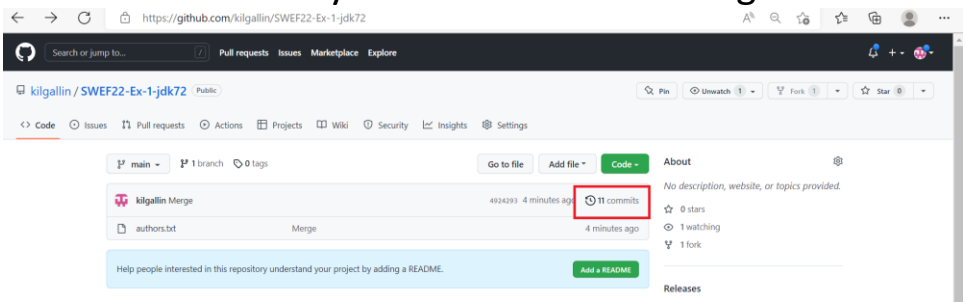


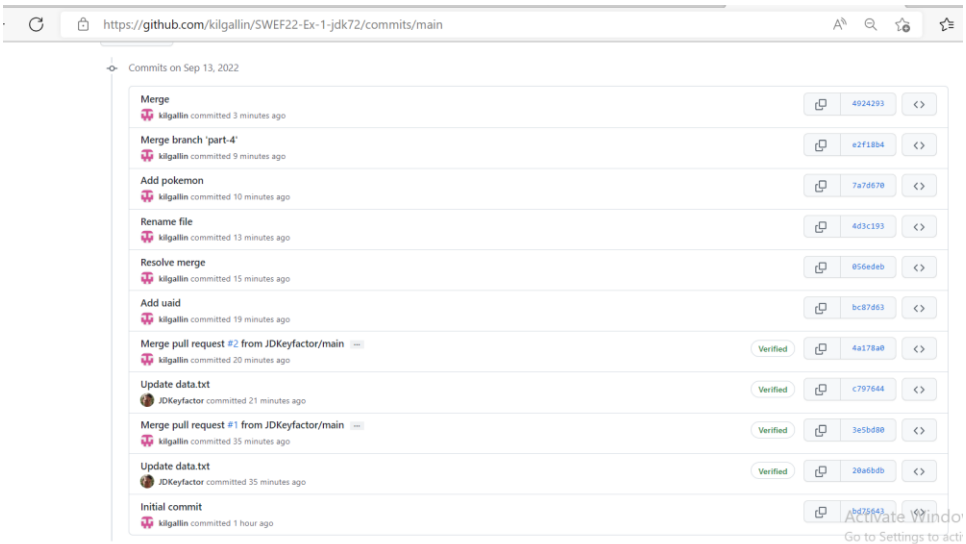
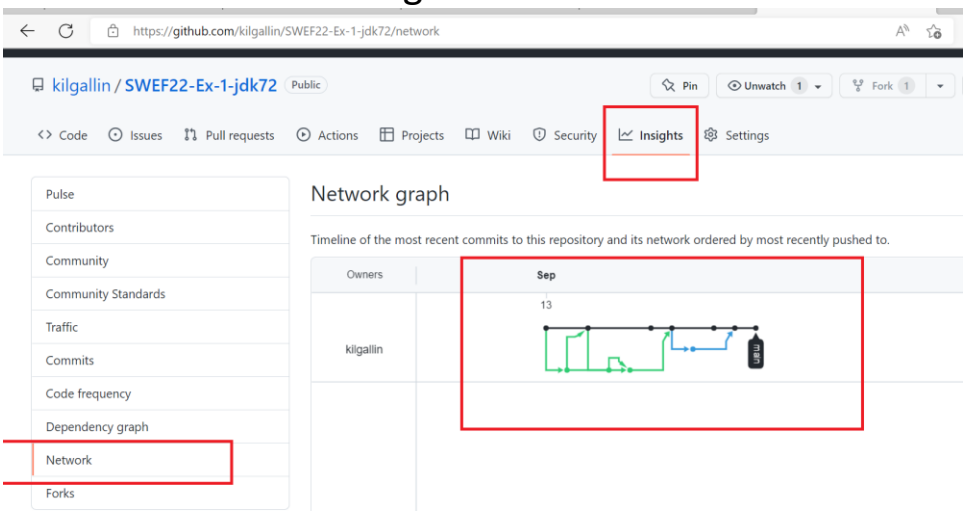
9. Commit and push the change. Note that the file is called authors.txt *and* has the Pokémon you added while the file was called data.txt.

```
jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   authors.txt

no changes added to commit (use "git add" and/or "git commit -a")

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git commit -a -m "Merge"
[main 4924293] Merge
 1 file changed, 1 insertion(+), 1 deletion(-)

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (10/10), 932 bytes | 466.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/kilgallin/SWEF22-Ex-1-jdk72
   056edeb..4924293  main -> main

jdk@DESKTOP-8L2NJNJ MINGW64 ~/github/SWEF22-Ex-1-jdk72 (main)
$
```



After this step, your repo should have the file “authors.txt” (and nothing else except possibly a README.md if you created it at the beginning) with content like the above. If you go to “Commits” you should see a history similar to the following.





If you go to “Insights->Network”, you should see something like this. It may vary slightly depending on how you and your partner committed changes. But it should show three merges: two from pull requests and one from a branch merge.