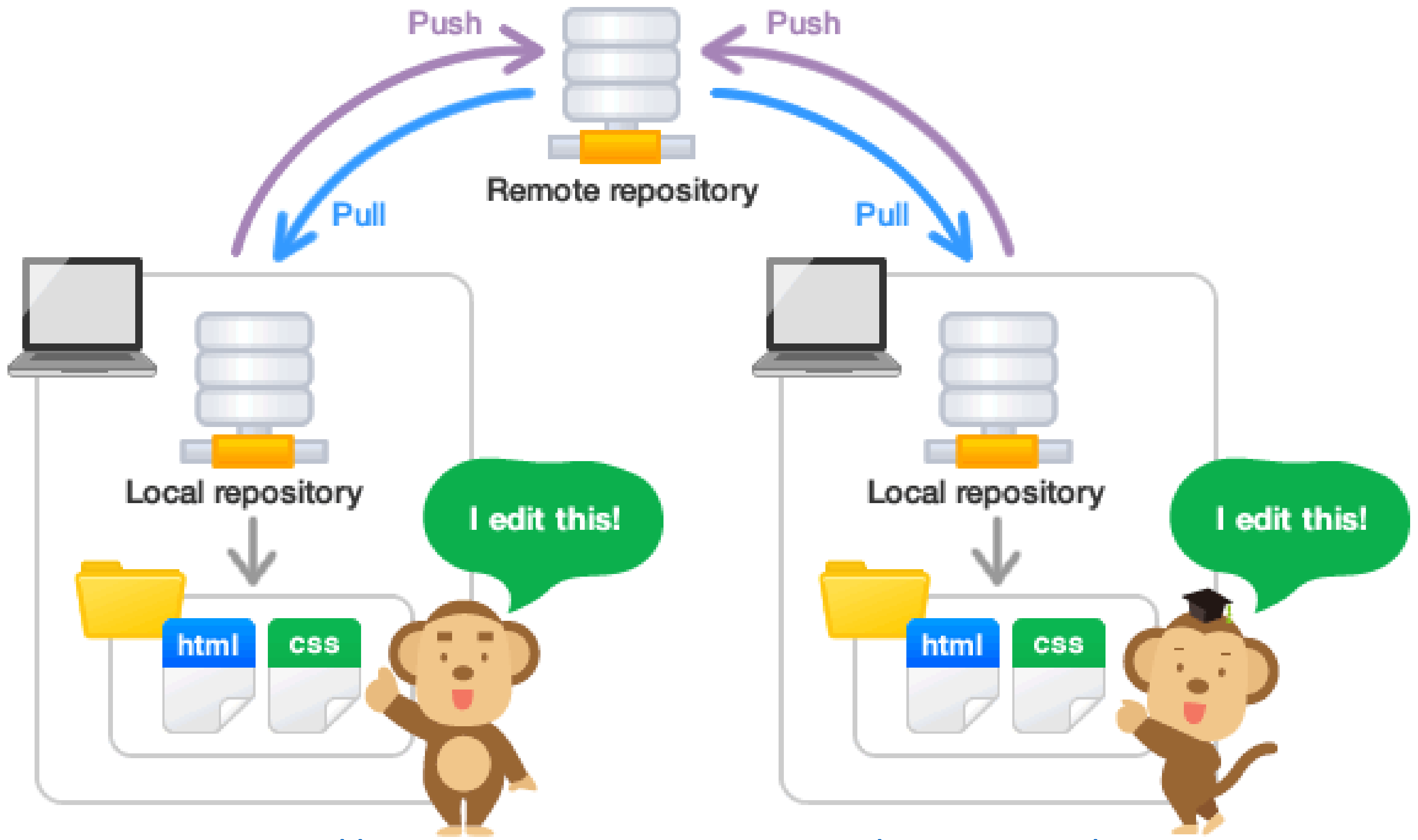
Using Git

Wonsun Ahn

Git Basics

- A means for software versioning and collaboration



Courtesy of: https://kevintshoemaker.github.io/StatsChats/GIT_tutorial.html

How to Clone Your First Repository

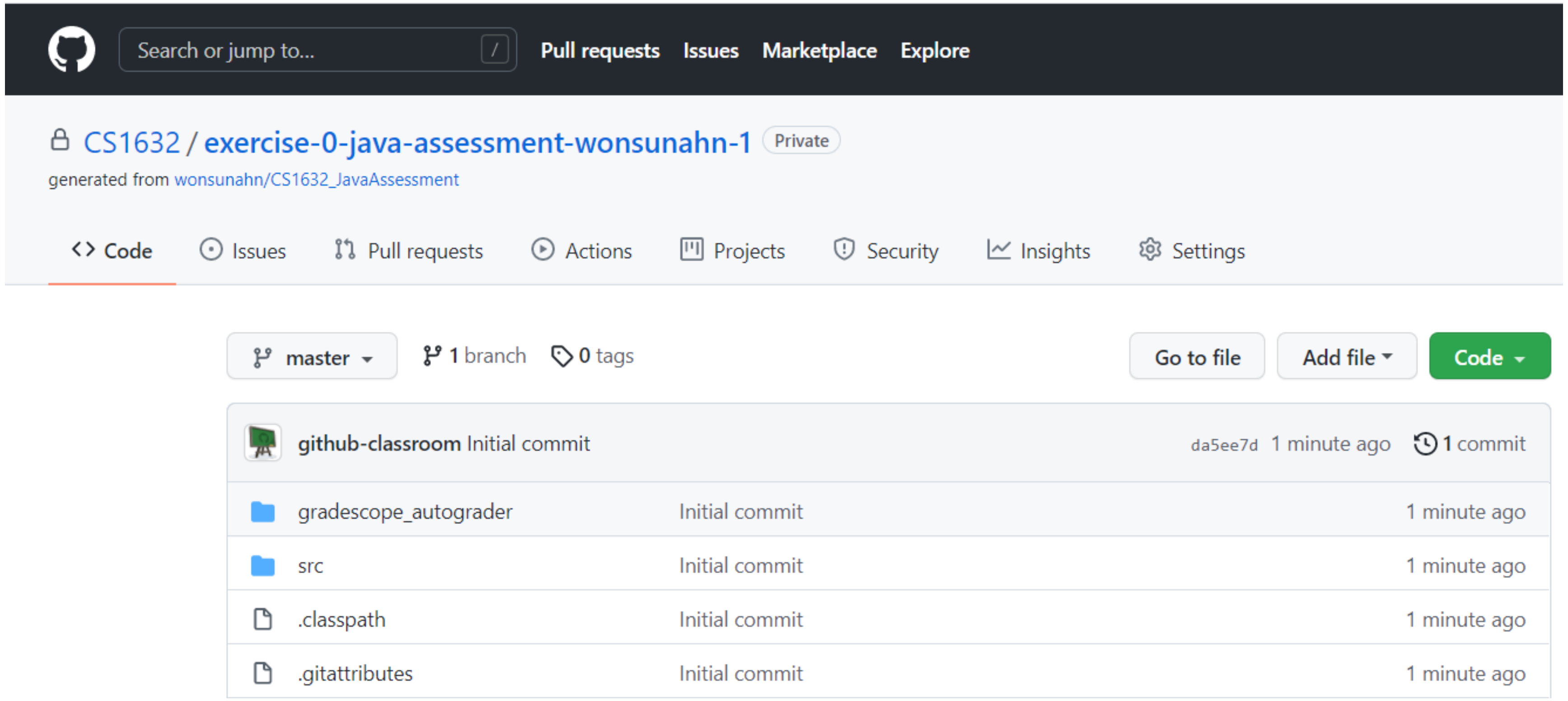
Using Exercise 0 as an example

Install GitHub Desktop

1. If you are new to GitHub, it’s easiest to use GitHub Desktop
   - Download from: https://desktop.github.com/
   - Install!

Ensure Remote GitHub Repository was Created

2. A few seconds after having accepted the assignment, if you refresh the "pending" webpage, you will see a link to your new repository



CS1632 / exercise-0-java-assessment-wonsunahn-1 Private

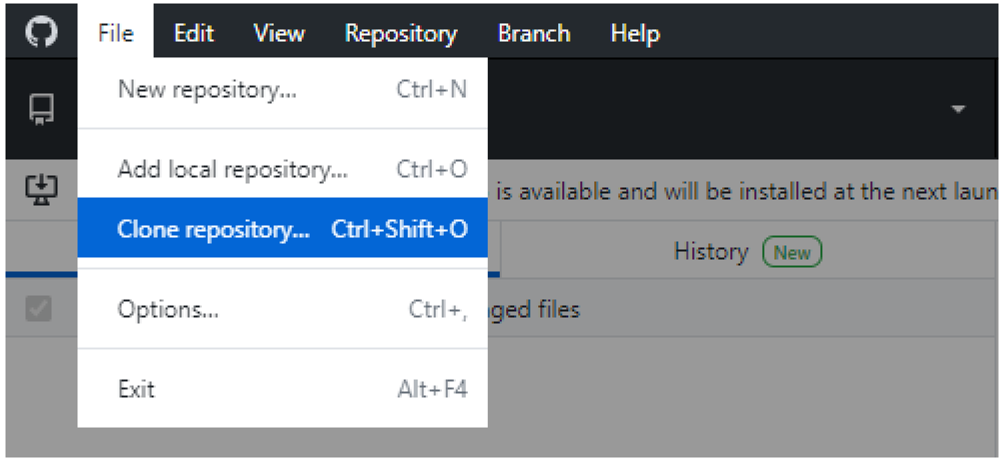
generated from wonsunahn/CS1632_JavaAssessment

github-classroom Initial commit da5ee7d 1 minute ago 1 commit

gradescope_autograder	Initial commit	1 minute ago
src	Initial commit	1 minute ago
.classpath	Initial commit	1 minute ago
.gitattributes	Initial commit	1 minute ago

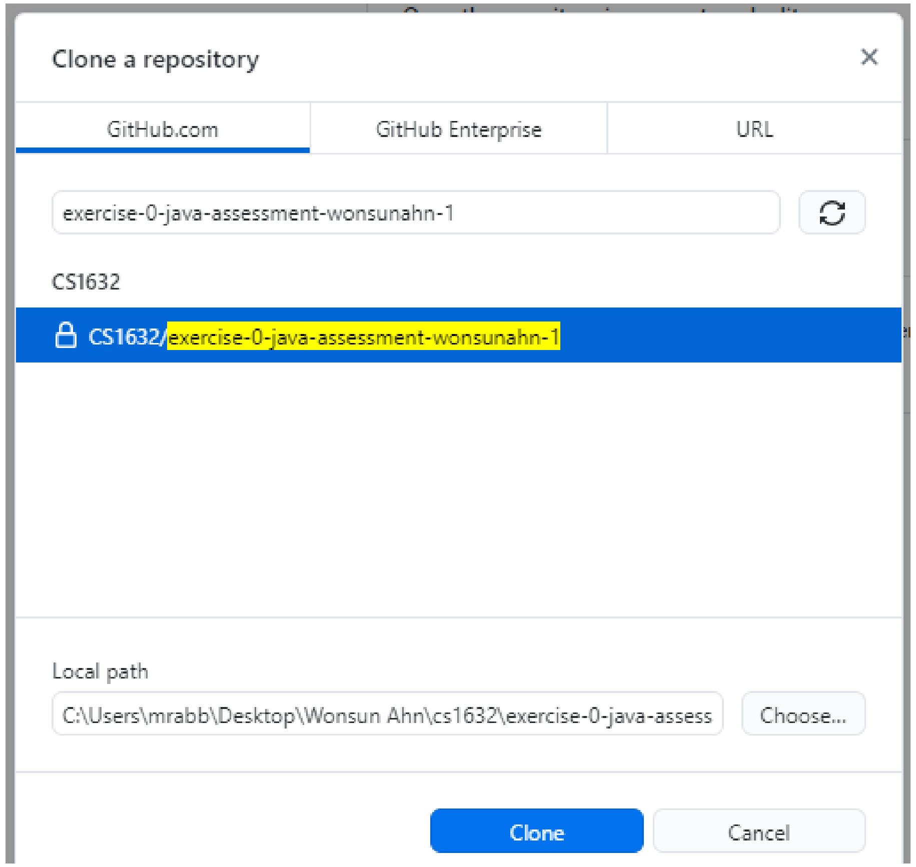
Clone GitHub Repo to Create Local Repo

3. Clone remote GitHub repository to create a local repository
   - Click on the clone menu:


File
Edit
View
Repository
Branch
Help
New repository... Ctrl+N
Add local repository... Ctrl+O
Clone repository... Ctrl+Shift+O
Options... Ctrl+,
Exit Alt+F4
is available and will be installed at the next laun
History New
ged files

Clone GitHub Repo to Create Local Repo

3. Clone remote GitHub repository to create a local repository
   - Choose the GitHub repository just created, and the local path:



exercise-0-java-assessment-wonsunahn-1

CS1632

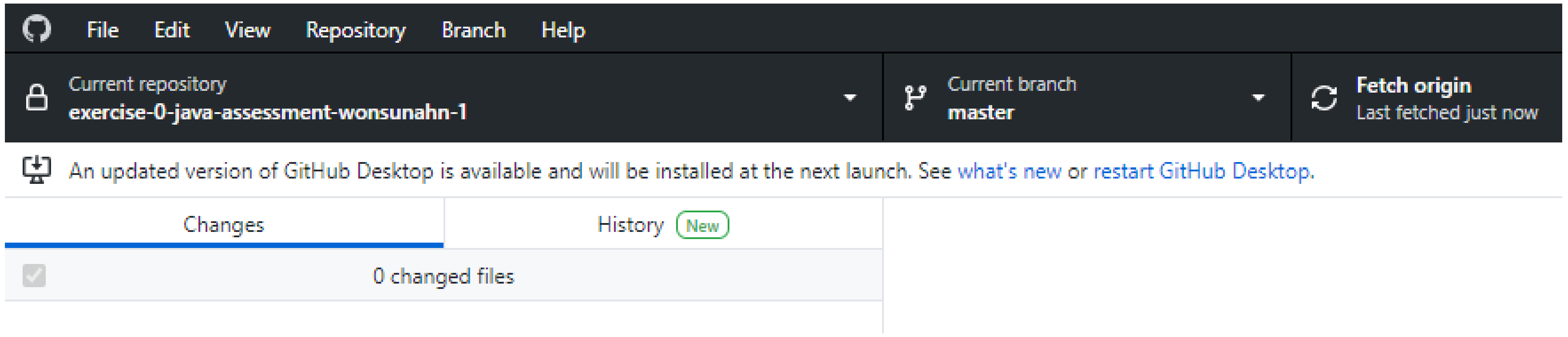
CS1632/exercise-0-java-assessment-wonsunahn-1

Local path

C:\Users\mrabb\Desktop\Wonsun Ahn\cs1632\exercise-0-java-assess

Clone GitHub Repo to Create Local Repo

3. Clone remote GitHub repository to create a local repository
   - Now you should see your newly cloned repository on GitHub Desktop:

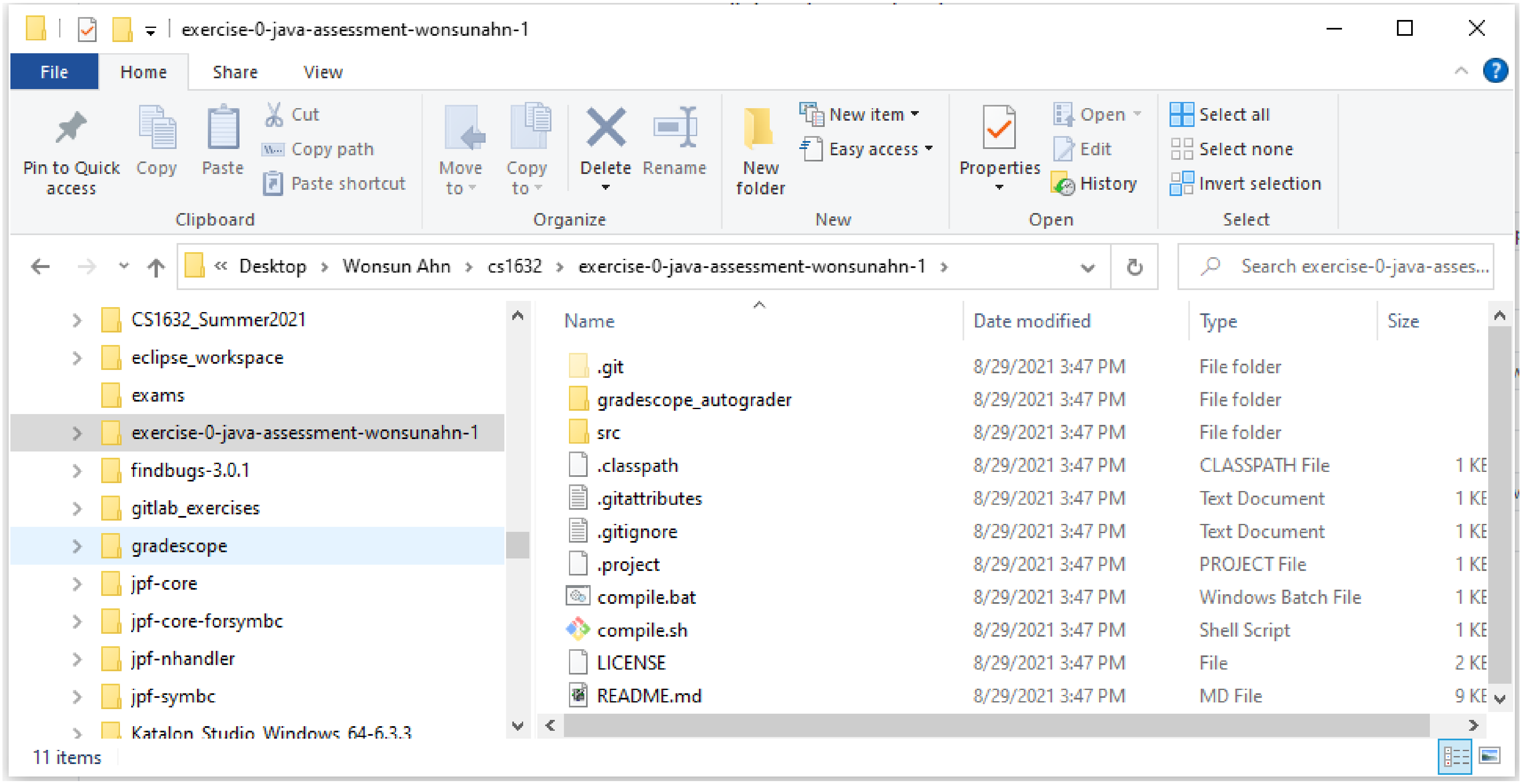


An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

0 changed files

Ensure Local Repository was Created

4. Navigate to the path where the local repository was created



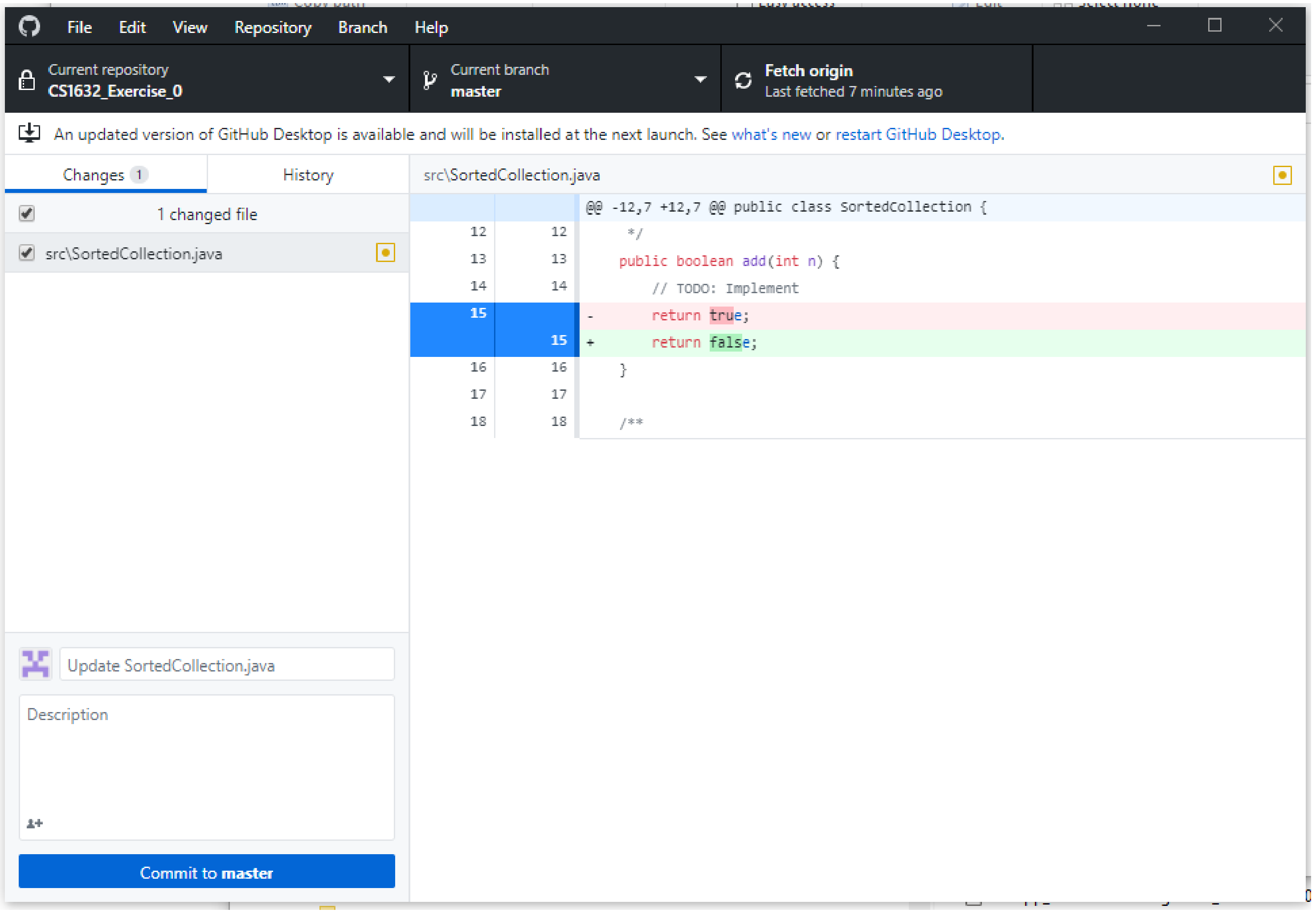
Name	Date modified	Type	Size
.git	8/29/2021 3:47 PM	File folder	
gradescope_autograder	8/29/2021 3:47 PM	File folder	
src	8/29/2021 3:47 PM	File folder	
.classpath	8/29/2021 3:47 PM	CLASSPATH File	1 KB
.gitattributes	8/29/2021 3:47 PM	Text Document	1 KB
.gitignore	8/29/2021 3:47 PM	Text Document	1 KB
.project	8/29/2021 3:47 PM	PROJECT File	1 KB
compile.bat	8/29/2021 3:47 PM	Windows Batch File	1 KB
compile.sh	8/29/2021 3:47 PM	Shell Script	1 KB
LICENSE	8/29/2021 3:47 PM	File	2 KB
README.md	8/29/2021 3:47 PM	MD File	9 KB

How to Push and Pull Updates

Using Exercise 0 as an example

How to Push Changes to github.com

1. Your changes are shown. Leave comment, click on "Commit to master".



An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

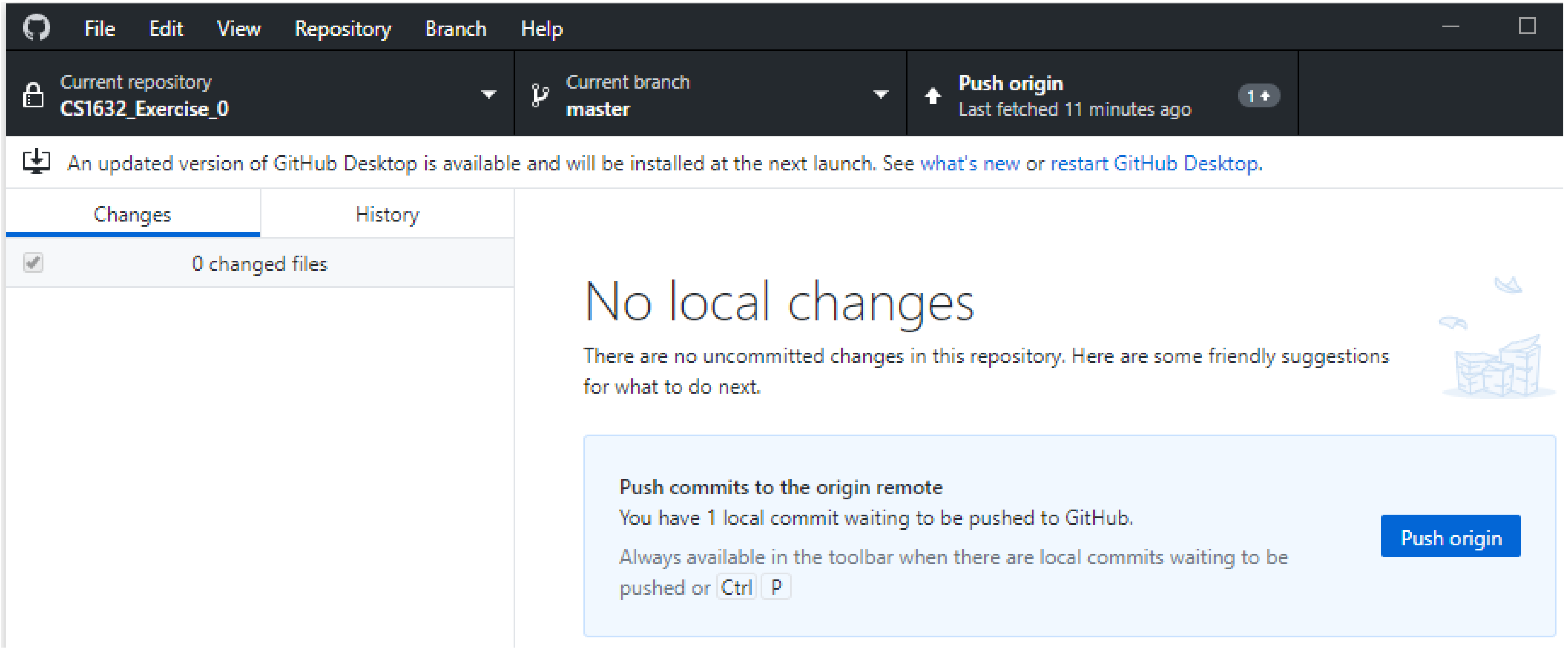
src\SortedCollection.java

```
@@ -12,7 +12,7 @@ public class SortedCollection {
     */
    public boolean add(int n) {
        // TODO: Implement
-       return true;
+       return false;
    }

    /**
```

How to Push Changes to github.com

2. Click on "Push origin" to upload changes to github.com



An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

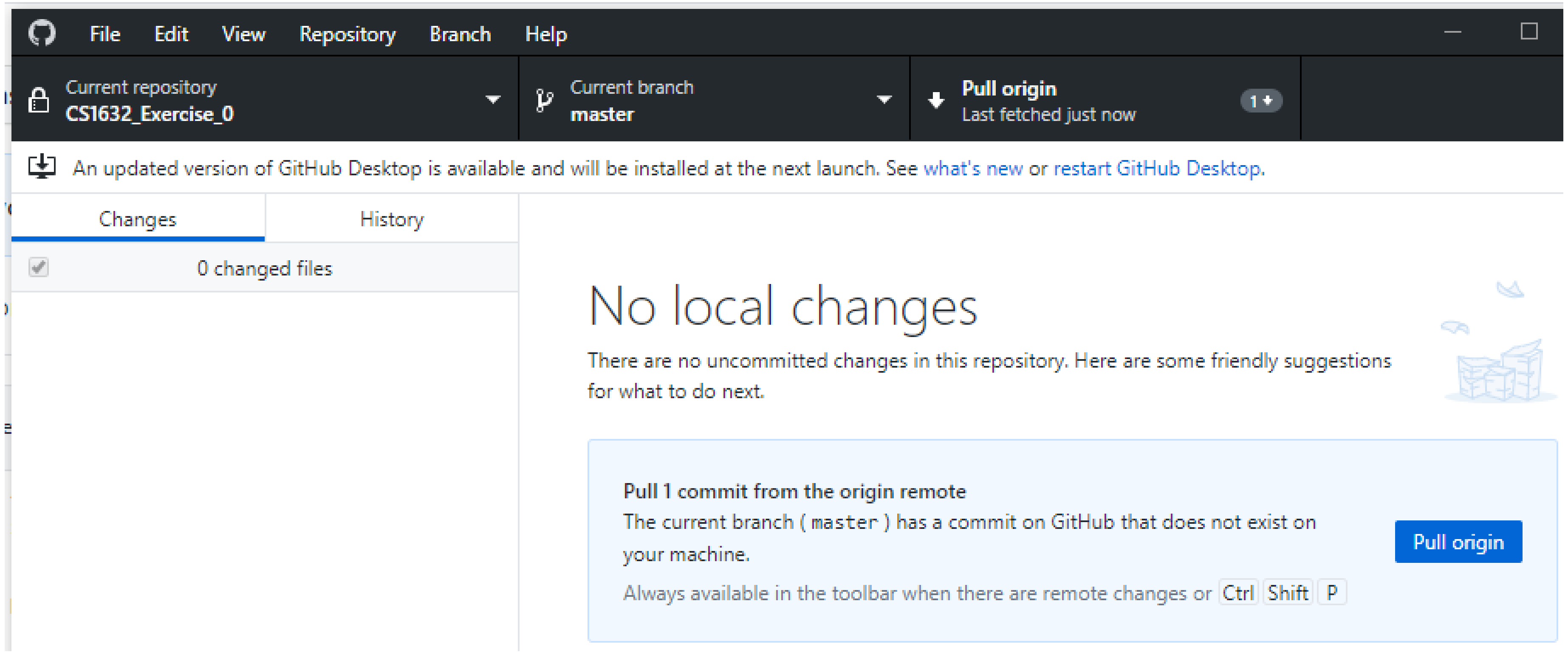
Push commits to the origin remote

You have 1 local commit waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or Ctrl P

How to Pull Changes from github.com

1. Click on “Pull origin” to download changes from github.com



An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull 1 commit from the origin remote

The current branch (`master`) has a commit on GitHub that does not exist on your machine.

Always available in the toolbar when there are remote changes or Ctrl Shift P

- GitHub Desktop will periodically check for updates and enable “Pull origin” button
- If you wish to manual check for updates, click “Check origin” button
  (Not shown here, but when no pending pulls, “Pull origin” turns to “Check origin”)

Git Etiquette

How to collaborate effectively when sharing a single repository

Git Etiquette 1: Push As Soon As Possible

- Your group member may be waiting …
  - For a feature to be implemented
  - For a bug to be debugged

- Push as soon as you have made a change that improves the project
  - If you delay pushing that change, the entire project will be delayed!
  - Pushing most of your code 2 days before the deadline is unacceptable

Git Etiquette 2: Leave a Descriptive Message

- You are required to leave a message whenever you commit
- Leave something descriptive so that your partner knows what you changed and what you are still working on

Git Etiquette 3: Do Not Push Bugs

- Worst thing you can do is to push a compile error
  - That means the project can no longer compile and no longer run
  - Entire project will be delayed until the error is fixed

- Do not push defects either
  - Do regression testing before pushing (run all unit tests written so far)
  - Make sure it doesn’t break something that used to run well

Git Etiquette 4: Pull / Push Frequently

- Before doing code changes → always pull the most recent version
  - Ensures that you work on the most up-to-date version
- After doing code changes → always push the committed changes
  - Ensures that your group members are work on the most up-to-date version
- If you do this, vast majority of changes will be sequentially ordered
  - Example of two ordered changes (blue by member 1, red by member 2):
    pull version 1 → update to version 2 → push version 2 →
    pull version 2 → update to version 3 → push version 3
    ☛ version 3 is applied on top of version 2
  - Example of two unordered changes:
    pull version 1 → pull version 1 → update to version 2 → update to version 2' →
    push version 2 → push version 2'
    ☛ Both version 2' and version 2 are applied to version 1. No ordering between them.
    What to do? Must merge the two versions to create a new combined version!

Merging Two Changes

- When two changes are unordered, Git will attempt to merge automatically
  - When the two changes are to different sets of files → success!
  - When the two changes are to the same file but different methods → success!
  - When the two changes are to the same source code line or very close → merge conflict!
- Example of a merge conflict in source code:

```
<<<<<<< HEAD
x++; (local version)
=======
y++; (remote version)
>>>>>>> 542582954eae8845ba6b0498d7d04ac09a6a63c6
```

- Must resolve by replacing above with whatever makes sense. Options:
  - `x++;`
  - `y++;`
  - `x++; y++;`
  - Or any other code that correctly merges the two changes

Merging Two Changes

- After resolving all conflicts, “commit merge” on GitHut Desktop
- Using GitHub Desktop to merge conflicts (recommended): https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/resolving-a-merge-conflict-on-github
- Using Git command line mergetool: https://gist.github.com/karenyyng/f19ff75c60f18b4b8149