Portfolio with Multiple IntelliJ Projects

You may have noticed that Professor Diekhoff's code for in-class activities is all in one IntelliJ project. This project is separated into packages representing the week.
For example, if we did JOptionPane stuff in class during week 5, you would find the main/test class inside of the package `week5`. This has been done for every week.
The in-class instructions for Lab 15 (the one where you integrated your GitHub account into IntelliJ and committed/pushed from inside IntelliJ) refer to this single project with all of the packages separated by week.

If you're reading this, chances are **you separated each of those activities into its own IntelliJ project.** Which is fine! It just means that your portfolio has to go together in a different manner. This document will show you how.

These are the steps to the process:

1. Install GitHub Desktop (and Git), and login to your account.
2. Create a new repository in GitHub Desktop.
3. Add your files to the (local) repository.
4. Commit and push those files to the remote (online) repository.
5. Publish the branch to the GitHub Repository (online) using GitHub Desktop.

Install/Setup GitHub Desktop

1. Go to https://desktop.github.com/ and download the correct installer.
   (The website should already recognize your device and provide you the correct installer.)

   - If you're on Windows, download the 64-bit installer.
   - If you're on Mac, download the archive containing the app.

2. Open the installer and it should automatically install itself.

   - If it doesn't, follow setup instructions here.

3. Login with your GitHub account.

   - If you're having trouble, see more about authenticating your GitHub account here.

4. Install Git, if you don't have it already. You should, but if not, see this page.

   - You do not need to open any of the git applications when they're installed. You can leave `Git GUI`, `Git Bash`, and `Git CMD` alone.

5. Go to **File > Options > Git** and ensure that the email is your anonymized GitHub email and NOT the personal email you actually used to create your GitHub account.
   -> My anonymized email is `153114658+katrazo@users.noreply.github.com`. Yours will look similar.

6. Your default text editor should be `JetBrains IntelliJ Idea Community Edition`. If it is not, go to **File > Options > Integrations** and change **External Editor** to IntelliJ.

Create a New Repository in GitHub Desktop

Before we begin, an important note:
Like I mentioned before, all of your IntelliJ projects that you want in the repository should be in one folder. The name of this folder is important.

1. Go to GitHub.com and make a new repository. ***You should give the same name to the repository that you gave your folder with all of the projects.***
   -> If all of your projects are in a folder called `cs121`, then the name of your GitHub repository should also be `cs121`.

2. Open File Explorer your `cs121` folder away from the desired location of the local repository.
   -> (If I want my repository to be in my `Documents` folder, I should make sure that my current `cs121` folder is NOT in that directory.)

3. In GitHub Desktop, go to **File > Clone Repository...**

4. In the popup menu, click on the repository you just made on GutHub.com. In the bottom section of the popup, put the file path of where you want the repository. Then hit **Clone**.
   -> (For me, this would be my `Documents` folder.)

5. Now, take your actual folder of projects and drag-n-drop it to the same directory as the cloned repository.
   ⭑ Since both folders have the same name, it should merge the subfolders (projects) into the repository.

6. Back in GitHub Desktop, along the left side of the window you will see a bunch of added files. This is good. This is your preview of what your next *commit* will be. Commits are essentially just file changes.

7. In the bottom left, give your commit a **summary** (required) and, if you want, a **description** (optional).

8. Click the blue button to push the commit. Then, in the window that takes up most of the screen (to the right), press the big blue button to Push to `<branch>`, or possibly to Publish the branch itself.

Congratulations! You did it! In GitHub Desktop, press `Ctrl` + `Shift` + `G` to open the repository online. All of your projects should now appear as their own folders in the repository. Now you can continue with a Readme file as expressed in Lab 15/Project 4.

Additional Resources

For GitHub:

- GitHub Desktop documentation
  - Installing GitHub Desktop
  - Connecting your GitHub acoount
  - Configuring Git
  - Basic Settings for GitHub Desktop

For the lab/project:

- Project 4 on Canvas

- Lab 15 on Canvas
- Ask questions in Slack!!!