Great—here’s the clean, reliable way to create **BUA477Lectures** as a new repo that starts as an exact copy of **BUA400Lectures**, with full commit history preserved. Then you’ll clone the new repo locally to work on it.

**Part A — Create BUA477Lectures on GitHub**

1. On GitHub, click **+ → New repository**.
2. Name it **BUA477Lectures**.
3. **Important:** leave it **empty** (do **not** add a README, .gitignore, or license).
4. Click **Create repository** and keep the HTTPS URL handy, e.g.  
   https://github.com/YOUR-USERNAME/BUA477Lectures.git

**Part B — Mirror-copy BUA400Lectures → BUA477Lectures (preserves history)**

You can do this from any folder; it does **not** depend on where your old local copy is.

# 1) Make a temporary bare mirror of the source repo

git clone --bare https://github.com/YOUR-USERNAME/BUA400Lectures.git

cd BUA400Lectures.git

# 2) Push \*everything\* (all branches + tags) to the new repo

git push --mirror https://github.com/YOUR-USERNAME/BUA477Lectures.git

# 3) Clean up the temporary mirror folder

cd ..

rm -rf BUA400Lectures.git

Result: On GitHub, BUA477Lectures now contains the exact contents and history of BUA400Lectures.

**Part C — Get a fresh local working copy of BUA477Lectures**

cd ~/Documents # or wherever you keep projects

git clone https://github.com/YOUR-USERNAME/BUA477Lectures.git

cd BUA477Lectures

Make your edits (add files, modify lectures, etc.), then:

git add .

git commit -m "Initial updates for BUA477 lectures"

git push origin main # or 'master', depending on your default branch name

Great question 👍 — those three lines are the **core Git workflow** that take your local changes and publish them to GitHub. Let me break them down:

**1. git add .**

* This stages **all changes** in the current folder (. means "everything here and below").
* Includes:
  + New files you created
  + Modifications to existing files
  + File deletions

Think of it like “marking these changes for inclusion in the next snapshot.”

**2. git commit -m "Initial updates for BUA477 lectures"**

* This creates a **snapshot** (a commit) of everything you staged with git add.
* The -m flag adds a short descriptive message so you (and others) know what this commit represents.  
  Example:
  + "Initial updates for BUA477 lectures" → tells you this commit was the first set of edits you made to the repo.

Now you have a new commit locally, but it’s still only on your laptop.

**3. git push origin main**

* This sends your local commits to **GitHub**.
* origin = the default name for the remote repo you cloned from.
* main = the branch you’re pushing to (older repos sometimes use master).

So this line says:  
👉 “Upload my local main branch (with my new commit) to the main branch on GitHub.”

**🔄 End Result**

After running all three:

1. Your edits are staged (git add).
2. They’re saved in a commit (git commit).
3. That commit is pushed to GitHub (git push).

Now your GitHub repo will show the updated files.

# Changes to Slides or Worksheets:

If you make a change to these rmd files, then SAVE these rmd files. They will be under the slides folder. Then, “knit” them. This will create the html file in the slides folder.

Now, to get these changed html files to the docs/slides folder, you need to “Build Website”.

Then, to push it to github, open up Git GUI, and open the local repository. Then, scan for changes, commit, and push. You can safely run the 5 buttons, one by one.