Instructions for how to fork Phillip’s repo and clone it locally

Make a github account if you have not already!
Download GitHub desktop.

Forking the repo:
https://github.com/phcompeau/ProgrammingforScientists2025Undergrad
Go into phillip’s repo and click *fork* in the top right corner. You can call your new repository whatever you want or keep it the same name.

After the forking repo, you will want to clone it into your local directory so you can edit it in vscode. Run

```
git clone https://github.com/<github_username>/ProgrammingforScientists2025Undergrad.git
```

MAKE SURE THE URL HAS **YOUR USERNAME**, NOT PHILLIP’S

Once it’s cloned, we want to set the upstream branch to Phillip’s repo so we can fetch updates when there are changes. Make sure that the url has *phcompeau* in it

```
git remote add upstream https://github.com/phcompeau/ProgrammingforScientists2025Undergrad.git
```

Run `git remote -v` to check that the origin and upstream locations are correct
Output should look something like this: (but with your username)

```
(origin  https://github.com/joellemc/ProgrammingforScientists2025Undergrad.git (fetch)
origin  https://github.com/joellemc/ProgrammingforScientists2025Undergrad.git (push)
upstream        https://github.com/phcompeau/ProgrammingforScientists2025Undergrad.git (fetch)
upstream        https://github.com/phcompeau/ProgrammingforScientists2025Undergrad.git (push)
```

Now everything is set so if there are changes in Phillip’s repository we can run

```
git fetch upstream
```

Note: We can’t run git pull because we would be pulling from our own repository, and this wouldn’t make any sense since no one but us is adding changes to our repository.

How to add, commit, and push your changes

Whenever we make a change locally and want to save it to GitHub, we will have to **add**, **commit**, and **push** our changes

To stage a change for a commit, run

```
git add all .
```

Note that the period adds all files that have been changed. You can also just add a specific file that you’ve changed. But in general, it’s good to save all changes, so GitHub will match with your local code.

To commit your changes, run

```
git commit -m “<message_detailing_the_changes>”
```

Note: that the -m flag will let you include a message. This message should be a short blurb about what changes you made to the files that you are committing. Your changes will not show up in GitHub yet until you push them.

To push your changes, run

```
git push
```

This will save your code into your own repository so that you can see it in GitHub now.