Git and PyTest Tutorial

Note: any time you see a word in brackets [], replace it with your own string

Create a New Branch

```
$ git fetch                                 # Grab all changes from origin
$ git checkout development_0/1              # Switch to the “master” branch for your team
$ git pull                                  # Pull all changes from origin into local branch
$ git checkout -b [new_branch]              # Create a new feature branch for your task
```

After Development is Finished

```
$ git add .                                 # Add all modified files on your current branch
$ git add [file1] [file2] […]               # Add specific modified files by name
$ git commit -m “[your description]”        # Commit your staged files with a description
```

You can repeat the above steps for as many commits as you want while you are developing. Typically a commit has some completed code or functionality.

```
$ git push                                  # Push all your local commits to origin
$ git push --set-upstream origin [branch name]  # First time you push will need to execute this
```

[Create a pull request on Github]

How to Run PyTest – may have to use “python -m pip” if “pip” by itself is not recognized

<u>Installation:</u>

```
$ pip install pytest
$ pip install pytest-html
```

<u>Execution:</u>

```
$ cd /path/to/your/folder/CIS322_SP22       # Change to the CIS322_SP22 folder on your computer
$ cd test                                   # Need to be in test folder for html component to run
$ pytest                                    # Try “python -m pytest” if pytest is not recognized
                                            # This will run every test file in the “test” folder
```

```
$ pytest -q [test_file]     .py                 # This will run a single test file
```

Make sure you have a function in your test file with a name beginning with “test_” or nothing will run

Make sure your test file is named using either “test_*.py” or “*_test.py”

```
$ pytest -q [test_file].py --html=[test_file].html  # Use this “--html” option to generate a test report
                                                    # Run this at the end when your test is already passing
$ git add [test_file].py [test_file].html           # Make sure to add and commit your test report
```