Intro to GitHub and Python

Description

The purpose of this assignment is to become familiar with GitHub and Python which will be the main forms of submissions for every programming assignment.

While some of you may not have heard of GitHub or used it before, the objective here is to incorporate industry likeness to each assignment. GitHub is meant to illustrate the record-keeping, secret-stashing, and collaboration opportunities that industry is about. For example, GitHub tracks every change made to repository (think of a repo as a drawer and everything that goes in and out of it) and allows repos to be either private or public (like a locking mechanism). Collaboration wise, GitHub allows multiple users to work together (if invited to the same repo) and tracks each individual change per user.

On the other hand, Python will be the main language used, as it is one of the most common languages used for analysis and calculations by mathematicians and statisticians alike. This assignment will help you familiarize with basic syntax and operations you will see frequently in this class.

For full credit on this assignment, the following is required:

- create standard structure
- create a python script with a specified name
- ensure there's a readme with instructions on how to compile your program
- Name the repository **“cot-4500-intro”**

Constraints

Standard Structure

When discussing “standard structure”, many can argue what is the best for each project. The consensus is that projects should not contain 20+ files at the top level with no subfolders (come on, this isn’t your desktop). However, it is important to note that every language has its own syntax guide and expected “structure”[1]. With a standard structure implemented, it becomes clear to the user where the main code lies and any other documentation.

For this assignment, you will need the following structure:

[1] There are deviations to structures within a single language. These are usually based on the project’s purpose and complexity.

```
Top Level
|-- src/
|   |-- main/
|   |   |-- __init__.py
|   |   |-- intro_to_python.py
|   |-- test/
|   |   |-- __init__.py
|   |   |-- test_intro_to_python.py|
|-- requirements.txt
|-- README.md
```

Specific File Name

While many can disagree that enforcing a specific file name is not ideal, this is the most efficient way to grade multiple submissions in a faster time frame.

This *OPTIONAL* section is about the pitfalls about enforcing filenames.
If interested continue reading, otherwise skip to the next section

Usually enforcing a filename on multiple submissions takes away a coder's *signature* or identity. Everyone has a unique way they code, and this can be in the number of spaces (or tabs if you're rebellious enough to do that) used to the length of variable naming. All of this contributes to **uniqueness** of one's code and how it can be compared against similarity checkers like TurnItIn and MOSS. For this assignment, we are not checking similarity but for future programs, we will.

For the assignment, you will need to have a file called **intro_to_python.py** in your *src/main* subfolder of your repo. If you choose to have additional files to maintain your *"identity",* that is okay, but all code must be called from the main file (i.e. if you make a second file called helper_.py, intro_to_python.py must call functions from helper.py) .

Compilation Instructions

Akin to industry standards, it is required to include a README file per repository. A README is a file used to describe a repo's purpose including compilation instructions and purpose of project. A proper README.md is typically filled out with sections like an outline of an essay. However, this will not be required.

For this assignment, a README is required. The amount of information you put into the README is up to you, but you will need to include the following:

- Mention of requirements.txt

    - Since this is part of the structure, you will need to figure out how to incorporate and use requirements.txt (hint: it's used to install third party libraries in python)
- Mention of running python
    - Must include instructions on how to run intro_to_python.py from the command line

Expected Output

With the above requirements addressed, we now arrive at the output section of the assignment. Since this class involves matrices and several mathematical operations, the intro assignment will address some basic libraries you might need to use.

The following must be output to the console (NumPy is required for this assignment)

Note: i refers to row index and j refers to column index

1. Print a specific 3x3 matrix where a cell is 1 if i == j, else 0
2. Print the 3x3 matrix from #1 and then add 3 to every cell where i ≠j
3. Print the 3x3 matrix from #2 as a 3x2 by deleting the last column from the matrix created

Based on the following rules, you will have the following output pattern:

NOTE: These values correspond to the question number and its output pattern. These are NOT the real expected values.

```
#1 #1 #1
#1 #1 #1
#1 #1 #1

#2 #2 #2
#2 #2 #2
#2 #2 #2

#3 #3
#3 #3
#3 #3
```