Week 1 Assignment

For this week's assignment, we will be writing some basic Python code as a review of IS210. We will start the semester off a little slow, so expect this assignment to be easier than assignments in the future.

Prerequisites

1. Our first step will be to create a github repo for this assignment. Please call it IS211_Assignment1.
2. After that is completed, use git to clone this repository to a local folder in your development environment. All development will be in this directory.

Part 1 - Functions and Exceptions

1. Create a new python file called assignment1_part1.py. All code for this part should be in this file and eventually pushed to Github.
2. Create a function named *listDivide* that takes in two parameters. One parameter is a list called *numbers*. The second parameter is an integer called *divide*. The *divide* parameter should have a default value of 2. The function returns the number of elements in the *numbers* list that are divisible by *divide*.
3. Create a custom exception class called 'ListDivideException'. This should be two lines of Python code.
4. Write another function called *testListDivide* that performs the following tests on *listDivide*:
    a. listDivide([1,2,3,4,5]) should return 2
    b. listDivide([2,4,6,8,10]) should return 5
    c. listDivide([30, 54, 63,98, 100], divide=10) should return 2
    d. listDivide([]) should return 0
    e. listDivide([1,2,3,4,5], 1) should return 5

    The function *testListDivide* does not return anything. However, if any of the tests fail, the function should raise the ListDivideException.

1. When your script runs, it should call *testListDivide*. Ideally, if your *listDivide* and *testListDivide* function is written properly, no exception should be raised. If the exception is raised however, try to figure out what is wrong. Keep updating the code until you know *listDivide* is working properly.
2. Once this is completed, run a 'git status' command. This should indicate to you that there is a file called assignment1_part1.py that is not in the repository yet. Run the correct git command to add this file to the repository.
3. Once that is done, commit this change (to the default master) using 'git commit'.
4. You can confirm this works by using 'git log' to see your commit
5. Push this commit to the 'origin', which is Github in this case.

Part II - Simple Class

1. Create a new python file called assignment1_part2.py. All code for this part should be in this file and eventually pushed to Github.
2. Create a class called Book. The class should have the following properties:
    a. Two attributes, author and title, both of which should be initialized to the blank string
    b. An __init__ function that takes in an author and a title, and sets them to the object variables

    c. A function called display, which when called, simply prints out a string representing the book. The output should be in the form of “title, written by author”. Example: “Of Mice and Men, written by John Steinbeck”.
3. Instantiate two objects from this class. The first object represents the book ‘Of Mice and Men’, written by John Steinbeck; the other is ‘To Kill a Mockingbird’ by Harper Lee.
4. Print both of these objects to the screen by calling their display() functions
5. Once this is completed, run a ‘git status’ command. This should indicate to you that there is a file called assignment1_part2.py that is not in the repository yet. Run the correct git command to add this file to the repository.
6. Once that is done, commit this change (to the default master) using ‘git commit’.
7. You can confirm this works by using ‘git log’ to see your commit
8. Push this commit to the ‘origin’, which is Github in this case.