Lab 0 - Description
(Computing the n\textsuperscript{th} Fibonacci Number)

Lab Overview:
For this lab, we will be learning how to compute the nth Fibonacci number from user input. In mathematics, the Fibonacci sequence is a sequence where any number in the sequence is defined by the sum of the two previous number in the sequence. To state this mathematically:
\[ F_n = F_{n-1} + F_{n-2} \]
Typically, the Fibonacci sequence is started at 0 (however, it can be extended to all integers as well). For this lab, we will start the sequence with the following:
\[ F_0 = 0, \quad F_1 = 1 \]
Your central task is to compute the n\textsuperscript{th} Fibonacci number from the sequence when the user inputs a number n.

Core Tasks:
1. Write a main function that asks the user for the number to calculate the Fibonacci sequence to.
2. Create a class named \textit{FibSequence}.
3. Write the method \texttt{computeFib(self, n)} for the \textit{FibSequence} class that allow us to generate the n\textsuperscript{th} Fibonacci number and print it to the console.

Program Requirements:

\textbf{Task 1:} Write a main function that asks the user for the number to compute the Fibonacci sequence to.

For task 1, you will need to write a main function with a while loop that continuously prompts the user to enter the number they wish to compute the Fibonacci sequence to. For instance, if we enter ‘5’ at the prompt then the program should print the 5\textsuperscript{th} number of the Fibonacci sequence before asking us to enter another number. \textbf{Note:} \textit{The 5\textsuperscript{th} number of the Fibonacci sequence is F\textsubscript{4} not F\textsubscript{5}. This is because the sequence is 0-indexed (i.e. we start counting at zero).} This loop will exit if the user enters the character ‘q’. \textbf{Note:} See image at the bottom of this section for I/O specifications. Your program must perform \textit{exactly} as shown in the figure.

\textbf{Task 2:} Create a class named \textit{FibSequence}.

For this task, you will create a class named \textit{FibSequence}. In the constructor for the class you can add any instance variables you deem necessary.

\textbf{Task 3:} Write the method \texttt{compute(self, n)} for the \textit{FibSequence} class that allow us to generate the n\textsuperscript{th} Fibonacci number and print it to the console.
For this task, your will do the main work of this assignment: computing the n\textsuperscript{th} Fibonacci number. There are a variety of ways to calculate the n\textsuperscript{th} Fibonacci number (golden ratio, generating functions, linear brute force, recursion, etc.)

**I/O Specifications:**
Your program must function as shown in Fig. 1 below exactly. The input will be done via the command line. Valid input will take the form of either positive integers or the character ‘q’.

![Figure 1: I/O Example](image)

**Remarks:**
All programs written in this class will be done using the newest version of Python available in the lab (Python 3.7/3.8). This is because Python 3.5+ is platform independent (i.e. you can code on a PC and run it on a Mac). The computers in KLA 26 all have python on them so make sure your code runs on the computers in the lab as it will be tested on an identical system. Feel free to install python on your own computer and bring it to the lab.

If you are new to python, here are some super helpful resources to get you up to speed:
1. Cheat sheets: [https://www.pythoncheatsheet.org/](https://www.pythoncheatsheet.org/) Most recommended! Many thumbs up!
2. Python documentation: [https://docs.python.org/3/](https://docs.python.org/3/)

All programming assignments are to be done individually. Your code will be looked at with professional software for cheating. **Warning:** This includes using online sources. (e.g. Do not go online and copy code from stack overflow. People have tried this before. You will fail.) Be extra
careful with your code. Do not ever show your work to anyone other than the TA (me) or the professor. They will most likely copy your work and your will both fail.

**Submission Requirements:**
In order to receive any credit for the assignment the student **must** do the following:

1. Name your program “<Duck-ID>_lab0.py”. (i.e. my duck ID is jhall10 so my submission would be named jhall10_lab0.py. **Note:** your duck-ID is the same as your email id and the username to log on to CIS computers **not** your 951… number that is your UO PID.
2. Submit your python file onto Canvas. 

That’s it! Make sure that you test your code on the lab computers to make sure it works.

---

**Grading:**
Your work will be graded along three primary metrics: Correctness, Completeness, and Elegance.

**Correctness: (60% of total grade)**
- Your program calculates the correct Fibonacci number.
- Your program follows I/O specifications.
- There are no logic bugs in your code.
- You wrote the main function as specified.
- You wrote the class as specified.

**Completeness (29% of total grade)**
- Your main function does not have any syntax errors.
- Your constructor does not have any syntax errors.
- Your class methods do not have any syntax errors.

**Elegance: (11% of total grade)**
- Your program is well organized.
- You make good use of whitespace, comments, and the file doc string.
- You write your code in a readable manner.
- You use descriptive variable/function/class names.

**Late Policy:**
The late policy is as follows:
Your homework is always due on a Wednesday. On Thursday, I will take 10% off. On Friday, I will take 20% off. No homework will be accepted after Friday.

If you encounter an unfortunate event or are working with a disability: Please email or speak to me. I am super flexible and am always on your side. I will give extensions as needed and am willing to work with you to make sure you get the most out of this course.