

# Lab 9 - Description

(Computing the  $n^{\text{th}}$  Fibonacci Number)

## Lab Overview:

For this lab, we will be learning how to how to compute the  $n^{\text{th}}$  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 numbers 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:

$$F_0 = 0, F_1 = 1$$

Your central task is to compute the  $n^{\text{th}}$  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 step to calculate the Fibonacci sequence to.
  2. Create a class named *FibSequence* according to the given specifications.
- 

## Program Requirements:

**Task 1:** Write a main function that asks the user for the step 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<sup>th</sup> number of the Fibonacci sequence before asking us to enter another number. **Note:** The 5<sup>th</sup> number of the Fibonacci sequence is  $F_4$  not  $F_5$ . 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'. **Note:** See the image at the bottom of this section for I/O specifications. Your program must perform exactly as shown in the figure.

**Task 2:** Create a class named *FibSequence* according to the given specifications.

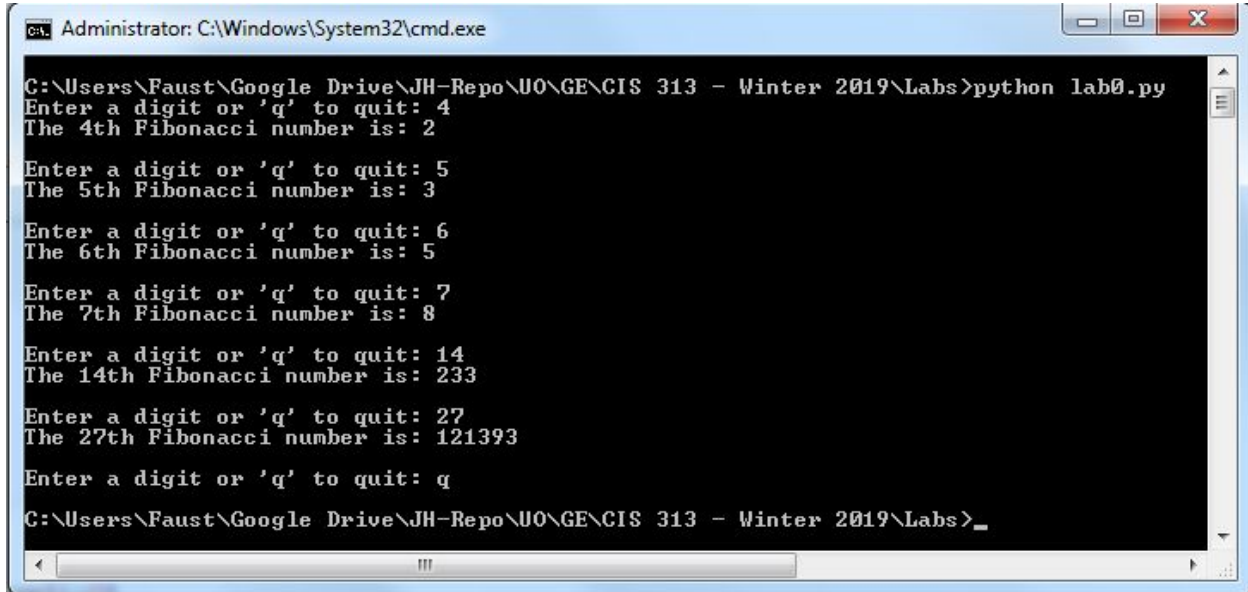
For this task, you will create a class named *FibSequence*. The methods of the class are as follows:

- `__init__(self) → None`
  - **Complexity:**  $O(1)$
  - Add any instance variables you deem necessary.
- `compute(self, n) → int`
  - **Note:** There are a variety of ways to calculate the  $n^{\text{th}}$  Fibonacci number (golden ratio, generating functions, linear brute force, recursion, etc). Implement the one that you think is best.
- `display(self, step, num) → None`

- This function outputs the calculated number to the console. The output must take the following format (also seen in the example below):
  - **Format:** The <step><suffix> Fibonacci number is: <num>
  - **Note:** The suffixes differ on the number according to the English language (e.g. 3rd not 3th).

### 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'.



```
Administrator: C:\Windows\System32\cmd.exe
C:\Users\Faust\Google Drive\JH-Repo\U0\GE\CIS 313 - Winter 2019\Labs>python lab0.py
Enter a digit or 'q' to quit: 4
The 4th Fibonacci number is: 2
Enter a digit or 'q' to quit: 5
The 5th Fibonacci number is: 3
Enter a digit or 'q' to quit: 6
The 6th Fibonacci number is: 5
Enter a digit or 'q' to quit: 7
The 7th Fibonacci number is: 8
Enter a digit or 'q' to quit: 14
The 14th Fibonacci number is: 233
Enter a digit or 'q' to quit: 27
The 27th Fibonacci number is: 121393
Enter a digit or 'q' to quit: q
C:\Users\Faust\Google Drive\JH-Repo\U0\GE\CIS 313 - Winter 2019\Labs>_
```

Figure 1: I/O Example

---

### 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). All of the programming projects will be tested on a VM running Linux Mint 20 (See the installation video for how to set up your machine).

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/> □ Most recommended! Many thumbs up!
2. Python documentation: <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 you 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 ***only*** the python file onto Canvas.

That's it! Make sure that you test your code on the VM 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 the given I/O specifications.
- Your program is robust and properly handles all of the corner cases.

#### Completeness (25% of total grade)

- You completed the main function and it does not have any syntax errors.
- You completed the class and it does not have any syntax errors.

#### Elegance: (15% 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 for this class is a bit unique.** Make sure to read the info below carefully:

Your homework is always due on a Tuesday at 11:59 pm. On Wednesday, I will take 10% off of the total points available. On Thursday, I will take 20% off. On Friday 30%, no homework will be accepted after Friday. I do not start grading the homework until the following Monday. Keep this in mind when submitting your assignment: It may be better for you to seek help and submit the homework a day late than to submit on time and fail.

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.