CS 112 Assignment 4a

Submit your finished code to the Dropbox. You will need to pass off the assignments with the TA or tutor during their office hours (their information is in the CS 112 Course Information section of the Table of Contents), or the instructor in class. Follow the formatting guide which is also found in the CS 112 Course Information section.

Read the instructions carefully. Make sure your output matches the example run.

Program: Loops, Input Validation and the Fibonacci sequence.

Loops are a powerful construct in all programming languages. One very common use for loops is in the validation of user input.

Investigate and Ponder: Run the following code in Python and discuss with your partner why you got an error. Failure to run the code and discuss the results with your partner before you attempt to pass off your program will result in a failed pass off.

Experiment #1
Write the following code in Python:

```
num = input("enter a number: ")
num2 = num + 10
print(num2)
```

Why did the code cause an error? What can you do to fix the error?

Experiment #2
Write the following code in Python:

```
num = int(input("enter a number: "))
```

Run this code in Python but instead of entering a number, enter the word ‘cow’ and hit return.

Why did entering ‘cow’ instead of a number cause an error? What can you do to fix the error?

Time to create your program!
Fibonacci numbers can be found everywhere in nature; rabbit populations grow very closely to the Fibonacci series. Art and music make use of Fibonacci numbers. Flower petals, pinecones, sea shells, tree branches, hurricanes and spiral galaxies all follow this amazing pattern. You can learn more here http://jwilson.coe.uga.edu/emat6680/parveen/fib_nature.htm. A Fibonacci sequence of 6 numbers looks like this: 1, 1, 2, 3, 5, 8. Can you figure out the pattern?

In this program you are going ask the user how many numbers in the Fibonacci sequence they would like to display. Your program will validate the input the user enters to ensure they entered a whole number. Your program will then print out the numbers in the Fibonacci sequence as specified by the user.

The overall design of your program will have two loops. The first loop will take care of input validation. The second loop will calculate and print the Fibonacci sequence.

First we take care of input validation. Follow the step by step guide below. In **Step 1 – Step 5** you will create a loop that takes care of input validation. In other words, those steps will ensure that the user entered a whole number and not a string or a float. If the user enters anything other than a whole number, the program will ask the user to try again. Important – The program will NOT continue until the user enters valid input.

Next we build the Fibonacci sequence. In **Step 7** you will initialize some variables that are needed to generate the Fibonacci sequence. In **Step 8 – Step 9** you will create a loop that generates and prints the Fibonacci sequence. Make sure you thoroughly discuss all Experiment results and Discussion questions with your partner BEFORE passing off your program.

Step 1
Open the CS 112 Python template and save it as Assignment 4a.py.

Step 2
In the main() function create a loop that will ask the user a question and validate that the user entered an integer. Indent using the TAB key and type the following:

```
while True:
```

Step 3
Use an input() statement to ask the user the following question:
“How many Fibonacci numbers do you want to generate (Enter a number > 2)? ”

Step 4
Assign the user's answer to a variable called numFibLoop. Do NOT cast the user's answer to an integer in this step! (To understand why, think about the error in Exercise #2 above.)

Since we are now writing a new block of code within the while statement above, you will need to indent again. Your code should now look something like this:

```
def main(): #main function need in all programs for automated testing

    #Validate that the user enters an integer
    while True:
        numFibLoop = input…….
```

Step 5
Now use the isdigit() method on your numFibLoop variable to validate that the input the user entered is indeed an integer. If it is a digit (all numbers and no letters), then cast the variable to an int and use a break to exit the while loop. If it is not a digit, then use an else statement to give the user an error message. Pay special attention to your indentation. The following code should be indented inside the while loop and on an equal level to the input() statement you wrote above.

```
 if numFibLoop.isdigit():
        numFibLoop = int(numFibLoop)
        break
else:
        print("Invalid response.  Plea……
```

That takes care of input validation. The program will run in an infinite loop, asking the user to enter a number. If the user enters anything other than a number (a float or a string for example) the program will ask the user to correct the problem and try again. Only when the user enters valid input (in this case a whole number) will the program move on.

Ponder – Discuss the following questions with your partner. Failure to discuss these questions before you attempt to pass off your program will result in a failed pass off.

- Discussion question 1: Since you code will now successfully perform input validation on your numFibLoop variable, what can you know with certainty about your numFibLoop variable?

- Discussion question 2: How will this benefit you in writing the code for the rest of your program?

Step 6
Output two empty lines AFTER the code has exited the while loop.

Step 7
Finishing Fibonacci.
Type the following code:

```
prevFib=0
currentFib=1
print(1)
```

Step 8
Create a while loop that uses `numFibLoop-1 > 0` as the test (this is a truthy test)

Step 9
Use the following code INSIDE the while loop

```
    newFib= currentFib +prevFib
    print(newFib)
    prevFib = currentFib
    currentFib = newFib
    numFibLoop =numFibLoop-1
```

Ponder – Discuss the following question with your partner. Failure to discuss this question before you attempt to pass off your program will result in a failed pass off.

- Discussion question 3: In programming, there are often many right ways (and an infinite number of wrong ways) to make a program work. Discuss with your partner another way you could have designed the Fibonacci while loop. In other words, what could you have used as the test other than `numFibLoop-1 > 0`?

Step 10

Run your code, correct any problems and make sure you understand why it works before passing off.

Have you figured out the formula for the Fibonacci sequence yet?

Example Run

```
How many Fibonacci numbers do you want to generate (Enter a number > 2)? cow
Invalid response.  Please enter a whole number > 2

How many Fibonacci numbers do you want to generate (Enter a number > 2)? 8.5
Invalid response.  Please enter a whole number > 2

How many Fibonacci numbers do you want to generate (Enter a number > 2)? 8


1
1
2
3
5
8
13
21
>>>
```