CS5001: Lab 11. Due on Friday, Dec-1-2023.

Name(s): Xujia Qin

Email(s): qin.xuj@northeastern.edu

You can work on this lab either individually or in small group of two or three students. If working in a group, include names of all the students in the submission PDF.

Getting credit for this lab. This lab handout has several empty boxes that prompt you to answer a question. As part of the lab, you are to write the answers to these questions inside the boxes/blanks. When you are finished, you should create a PDF and upload it on Canvas. If you don't finish, you have until 11:59 PM on Friday, Dec-1-2023 to submit.

What computer to use? If your primary computer is a laptop, bring it to the lab to work on, as lab is an excellent opportunity to get started with Python on your machine. You should follow the instructions on the course website. Ask a TA for help if you have problems with your installation. If you prefer, you could also use one of the machines in the lab room to work on this lab assignment.

Lab Materials. Lab materials can always be found on Canvas under the appropriate lab posting.

For today's lab, you need: this handout, OrderDates.py, ShowFractionClass.py, ShowSimpleDateClass.py, TheFractionClass.py, TheSimpleDateClass.py~~.~~ All the files are posted on Canvas with the lab handout.

1. The Class Fraction

1.1) Study the definition of the Fraction class that is given in TheFractionClass.py. Review how it can be used by browsing through and running ShowFractionClass.py.

Recall that we can approximate square roots by repeated averaging:

```
def sqrtF(a):
    """ Returns a float that approximates the square root of a obtained by doing 5
    "rectangle averagings."

    PreC: a is a float with positive value """
    x  =  a
    half = 1./2.
    for k in range(6):
        x = (x + (a/x))*half
    return x
```

In ShowFractionSqrt.py, implement an analog of this that works when “a” and the returned approximation are Fraction objects. The intent is just for you to “translate” the mentions of floats into mentions of objects and mentions of operations on floats into method calls on objects. The module ShowFractionSqrt.py is set up for you to do this. Complete the implementation of the following function in ShowFractionSqrt.py and submit the completed program file on Canvas. **(30 points)**

```
def sqrtF(a):

    """ Returns a Fraction that approximates the square root of a obtained by doing 5
    "rectangle averagings."


    PreC: a is a Fraction with positive value """
```

1.2) In TheFractionClass.py complete the function definition of the following method() to do as described in the docstring. **(30 points)**

```
def __sub__(self,f):
    """ Returns a Fraction that is the difference of self and f.
    If f1 is a Fraction and f2 is a Fraction or an int, then
    f3 = f1-f2 is a Fraction object that represents their difference.

    PreC: f is either an int or a Fraction
    """
```

2. The Class SimpleDate

Study the definition of the SimpleDate class that is given in TheSimpleDateClass.py. Review how it can be used by browsing through and running ShowSimpleDateClass.py. Run the module OrderDates.py and observe that it generates a list of 5 randomly generated SimpleDates. Modify the module so that it prints the list three times, once with it sorted by year, once with it sorted by month index, and once with it sorted by day. You will have to write three getter functions and apply the sort method three times. Complete the implementation of the three getter functions and their three invocations in the file OrderDates.py. Submit the completed program file OrderDates.py on Canvas. **(40 points)**

What to submit on Canvas?

Submit the completed versions of the files OrderDates.py, ShowFractionSqrt.py, TheFractionClass.py that include your code for the functionality specified above in a single zip file.