**Homework 10**

**Due: Sunday (see Syllabus)**

**Points: 40**

**Instructions**

In this exercise you will create a class named **Frac** to represent fractions of the form integer/integer (aka rational numbers). **Frac** items will need to support the usual arithmetic operators. You will use a tiny demo() function (provided) to demonstrate that your Frac class functions properly. Then you will use **Frac** to approximate the value of Pi (**p**).

**Specifications**

**The Frac Class**

1. **Frac(int, int)** should yield a **Frac** object.
2. **Frac(int)** should yield a **Frac(int, 1)** object.
3. These arithmetic operators must be overloaded: **+ - \* // =**
4. The **\_\_str\_\_()** function must be overloaded so that it displays 3/4, for example for **Frac(33,44)**.
5. The **\_\_abs\_\_()** function must be overloaded so that it returns **| Frac() |**.
6. The **\_\_float\_\_()** function must be overloaded so that it converts **Frac(3,4)** into 0.75, in other words it returns the decimal equivalent of the fraction.

**Additional**

1. Define a **frac\_input(s)** function that uses input(s) to acquire a fraction specified as a string from the keyboard as either an int or int/int. It should not allow zero denominators. It should not allow illegal inputs (e.g., non-int inputs). In illegal input cases it should loop repeatedly asking for valid input.
2. Your main should use the **demo(a,b)** function (provided) to demonstrate that your **Frac** class works properly.
3. Your main should be based on the outline given in **frac\_main.py**.
4. Graphical user interface, text, application

   Description automatically generatedAfter your main has demonstrated that your **Frac** class works (using **demo()**), it needs to approximate the value of Pi (**p**):
5. You program should keep adding terms until the difference in two successive approximations is less than 1 part in 1014 (1e-14) if possible. This will provide a little over 14 digits of precision. Display your approximation after each 5,000 approximations. Compare your final computed value with **math.pi**.
6. Your output should somewhat match what is shown below.

Run your program several times using different inputs – sufficient to demonstrate that your program meets all the assignment requirements. Capture a screen shot of each run and paste them into an MS Word document. Place a caption above each image.

**Submit the Python .py file (lastname\_hw9.py) containing your program and the MS Word document** **to your instructor using the appropriate Assignment Submissions link.**

Attach: ***demo.py***

***frac\_main.py***

**Sample Output**

Text

Description automatically generated

Graphical user interface, text

Description automatically generated