CSE 4256

Due Sunday Feb 20 11:59 PM EST

Dr. Alan Weide

Spring Term 2022

Homework 5

Assignment

This assignment should be completed as a pair of Python source (`.py`) files with the following names:

- `main.py`
- `hw5_fraction.py` modified from `hw5_fraction.py` (avaialble on Carmen)

Each question (including the Challenge Activities) has an associated `TODO` comment in the template files.

(1) Implement the `Fraction` class in `hw5_fraction.py`.

(2) Create a file called `main.py` in the same directory as `hw5_fraction.py`. In the new file, import `Fraction` into `main.py` in two ways:
   - By importing the entire module `hw5_fraction` into `main.py`
   - By importing only the class `Fraction` into `main.py`

(3) Pick one of those imports and give the imported entity a shorter name.

(4) Write some code in `main.py` to test your implementation of the `Fraction` class. Here are a few suggested classes of test cases:
   - Initialization and printing of a fraction with value 0. **Hint**: Such a fraction should have denominator 1 in reduced form.
   - Initialization and printing of a fraction with numerator and denominator that are relatively prime.
   - Initialization and printing of a fraction with nonzero numerator and denominator that are *not* relatively prime.
   - The sum of two fractions with the same denominator.
   - The sum of two fractions with different denominators.

(5) Augment the `Fraction` class with a method to enable `Fractions` to be compared with the less-than operator <.

(6) Use `functools.total_ordering` to enable `Fractions` to be compared with *any* of the comparison operators in Python.

(7) Add code to test this new functionality of `Fraction`.

(8) Move your test code from `main.py` to the end of `hw5_fraction.py` such that they are only executed if `hw5_fraction.py` is invoked as the main program by the Python interpreter. At this point, the file `main.py` should only have import statements in it.

Challenge Activities

Some homeworks (such as this one) will have additional challenge activities. These activities **do not contribute to your grade**, but they are problems that I find interesting or challenging.

(9) Augment the `Fraction` class with methods to support integer exponentiation with the `**` operator and absolute value with the built-in **`abs`** function.

(10) Augment the `Fraction` class with a method to support conversion to **`bool`**. **Hint**: by Python convention for numeric types, the value 0 should evaluate to `False`, and any other value should evaluate to `True`.

(11) Augment the `Fraction` class with a class method called `from_str(cls, str_rep: str) -> Fraction` with the following definition.

```
@classmethod
def from_str(cls, str_rep: str) -> 'Fraction':
    """Produces a fraction from string str_rep.

    Requires str_rep is in one of two forms.
    Either str_rep is the string representation of a fraction
        (e.g., '5/3' or '-18/36'),
    or str_rep is the string representation of a decimal number
```

```
        (e.g., '47.625' or '-8.3333').
    """
```

(12) Modify the `__init__` method to take either two **`int`** arguments or a single **`str`** argument, treated as it is in `from_str`.

(13) Does your implementation of `Fraction` "gracefully" handle negative numbers? In particular, the statement `Fraction(2, -4)` should produce the fraction `-1/2` and `Fraction(-5,3).mixed_number()` should return `"-1 2/3"`. If this is the output that you get: great! If not, fix your `Fraction` class to handle it.

Submission

To submit this assignment, upload a **`.zip`** file containing both Python files to the "Homework 5" assignment on Carmen. As always, be sure to note all group members who contributed to the assignment and what those contributions were.