Lab9 – Introduction to OOP

Lab Objectives

The aim of this worksheet is to introduce you to Object Oriented Programming in Python by getting you to:

- Define your own class
- Write code for class attributes and methods
- Use the class to create and manipulate objects

You will develop a package that can manipulate fractions symbolically in a robust manner by appropriate use of Exception handling. We need to provide a user interface i.e. functions that can be usefully used by a third-party user of the code.

Problem Statement

Create a class called **Fraction** for performing basic arithmetic with fractions. Use integer variables to represent the state attributes of the class – the **numerator** and the **denominator.**

We have started solving this problem in this week's lectures. You may want to read the slides again and consult the following documentation (scroll down at Section 3.3.) https://docs.python.org/3/reference/datamodel.html# for special operators in Python object-oriented programming. Answer the following questions:

1. Write a constructor method `__init__` that enables an object of this class to be initialized when it is declared. Your constructor should be able to create an object initialized using default attribute values. For that purpose, you may use optional arguments to the method `__init__`. Use exceptions handling as deemed appropriate. Moreover, your constructor should store the fraction in reduced form, i.e., the fraction 2/4 should be stored as ½. To this end, implement the method `gcd(a, b)` that calculates the greatest common divisor of two integers a and b and use it to simplify the fraction.

Here is a pseudocode for the iterative gcd function for your convenience:

```
function gcd(a, b)
   while (a != b)
       if (a > b)
          a = a-b
   else
      b = b-a
return a
```

Hint: *This constructor is a bit complex, go step by step. First, implement the gcd method and then use the calculated gcd to simplify the fraction. Also, consider raising an exception when the denominator is zero*.

2. Write the method `__str__` to print a given Fraction.
3. Write the method `__eq__` to test the equality of two Fractions
4. Write the method `__add__` to add two Fractions. The result of the addition should be stored in reduced form.
5. Write the method `__sub__` to subtract two Fractions. The result of the subtraction should be stored in reduced form.
6. Write the method `__mul__` to multiply two Fractions. The result of the multiplication should be stored in reduced form.
7. Write the method `__truediv__` to divide two Fractions. The result of the division should be stored in reduced form.
8. Write the method `scale` that takes as inputs a list of Fraction `objects [x1, x2, …, xn]` and an integer factor and returns the list of simplified Fraction objects `[x1/factor, x2/factor, …, xn/factor].`
9. Write a method `printListOfFractions` that takes in a list of Fraction objects and print them on the screen.