Lab Exercise #13

Assignment Overview

This lab exercise provides practice with classes and operator overloading in Python.

You will work with a partner on this exercise during your lab session. Two people should work at one computer. Occasionally switch the person who is typing. Talk to each other about what you are doing and why so that both of you understand each step.

Part A: Class Fraction

1. Examine the files named "fraction.py" (which contains the definition of class "Fraction") and "lab13.parta.py" (which contains a simple test bed), then execute the test bed.

    a) Under what circumstances will the constructor (method "__init__") create an object with 0 as the numerator and 1 as the denominator?

    b) What is the difference in appearance of an object displayed using "__repr__" and that same object displayed using "__str__"?

2. Revise the test bed to use the fraction 4/5 as both arguments to function "display", and then execute the test bed.

3. Revise "fraction.py" so that variable "debug" is initialized to True (instead of False), and then execute the test bed.

4. Revise the test bed to use the fraction 3/4 and the integer 5 as the arguments to function "display", and then execute the test bed.

    a) What function is called by function "display" when the expression "arg1 - arg2" is evaluated?

    b) What function is called by function "display" when the expression "arg1 < arg2" is evaluated?

5. Revise the test bed to use the integer 7 and the fraction 1/4 as the arguments to function "display", and then execute the test bed.

    a) What function is called by function "display" when the expression "arg1 - arg2" is evaluated?

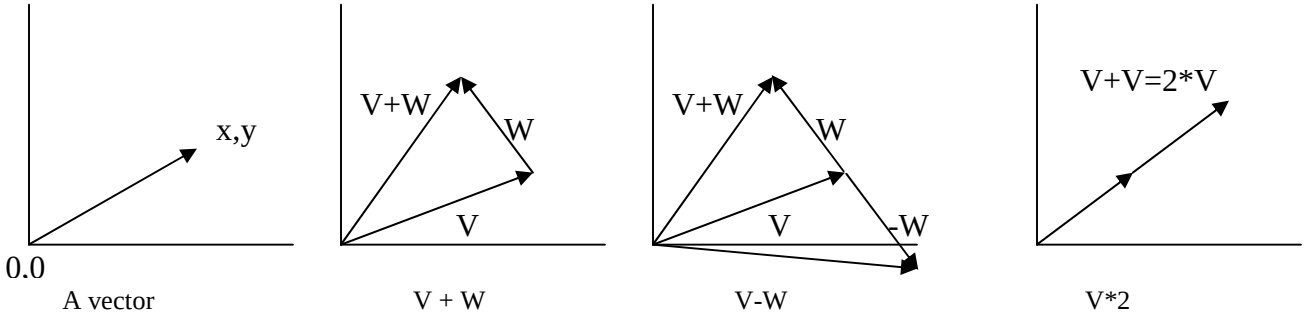
    b) What function is called by function "display" when the expression "arg1 < arg2" is evaluated?

★ **Demonstrate your completed program to your TA. On-line students should submit the completed program (named "lab13.parta.py") for grading via the CSE handin system.**

Part B: Class Vector

Develop class "Vector", which supports the manipulation of two-dimensional vectors.

A two-dimensional vector is basically an arrow which has a magnitude (a length) and a direction (an angle with respect to the x axis). It usually is represented as an (x,y) pair, where the origin of the vector is at (0,0) and the head of the vector is at the listed pair.



Operations which can be performed on a vector include:

- Vector addition: if V is (x,y) and W is (a,b), then V+W is the vector (x+a,y+b).
- Vector subtraction: V-W is the same as V+(W*-1).
- Vector multiplication by a scalar: if V is (x,y), then V*n (and n*V) is the vector (x*n,y*n).
- Vector multiplication by another vector (dot product): if V=(x,y) and W=(a,b), then V*W = x*a + y*b, which is a scalar.
- Vector magnitude: the magnitude for V=(x,y) is sqrt(x**2 + y**2), which is a scalar. You might wish to look at function "hypot" in the math library.

Define class "Vector" and a test bed in the file named "lab13.partb.py". Be sure to demonstrate that each method in class "Vector" is implemented correctly.

Your class "Vector" will include the following methods:

__init__	The constructor; accepts three arguments (self, x and y), where x and y default to zero.
__repr__	Used for displaying in the shell; returns a string representing the vector.
__str__	Used for printing; returns a string (with two digits to the right of the decimal point).
magnitude	Calculates the magnitude of a vector; accepts one argument (self), returns a float.
__eq__	Equality comparison; accepts two arguments (self and another vector), returns a Boolean.

__add__ Vector addition; accepts two arguments (self and another vector), returns a new vector.

__sub__ Vector subtraction; accepts two arguments (self and another vector), returns a new vector.

__mul__ Vector multiplication; accepts two arguments, supports three possibilities:

vector * float – returns a new vector (scalar multiplication)
float * vector – returns a new vector (scalar multiplication)
vector * vector – returns a scalar (dot product)

★ **Demonstrate your completed program to your TA. On-line students should submit the completed program (named "lab13.partb.py") for grading via the CSE handin system.**