Lab Exercise #12 Operator Overloading

Assignment Overview
We are going to work with overloaded operators and making our own class. We are going to make a 2D vector class.

Some Background

So if you don't remember, here is a little background on two-dimensional vectors.
A vector is basically an arrow that has a magnitude (a length) and a direction (an angle with respect to typically the x axis). It usually is represented as an x,y pair, where the origin of the vector is a 0,0 and the head of the vector is at the listed pair.

| A vector | V + W | V-W | V*2 |
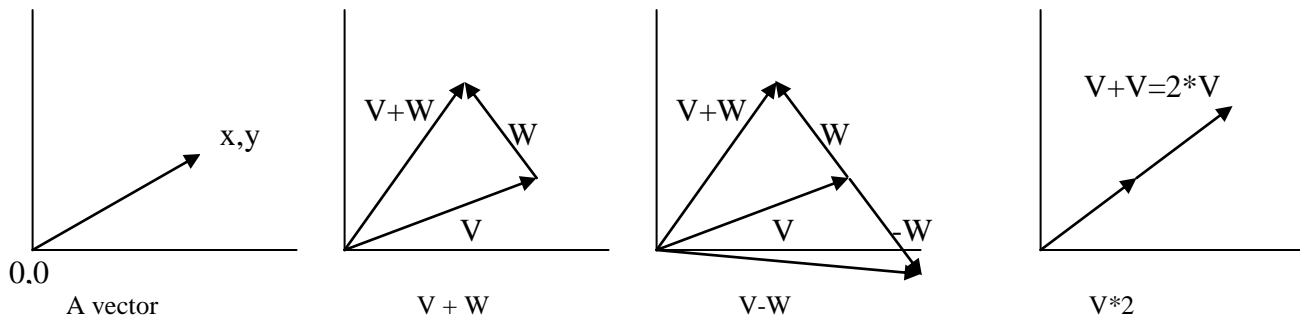
Here are some of the operations you can perform on a vector.
- vector addition. If V1 is (x,y) and V2 is (a,b), the V+W is (x+a,y+b), a vector
- vector multiplication by a scalar. if V1 is (x,y), the V*n is (x*n,y*n), a vector
- vector subtraction V-W is the same as V+(W*-1), a vector
- vector multiplication with another vector. There are two possibilities, dot product or cross product. We'll do dot product. If V=(x,y) and W=(a,b), then V*W = x*a + y*b, a scalar. Thus the dot product yields a scalar, not a vector
- vector magnitude. The magnitude based on the Pythagorean theorem for a V=(x,y) says that the magnitude is sqrt(x**2 + y**2). You might look at math.hypot for this

Your Tasks

Make a vector class. Provide the operators

__init__	# constructor. Takes 3 args: self,x,y. Default for both x and y is 0. No return
__str__	# for printing. Takes 1 arg: self. Returns a string (precise to 2 decimals).
__repr__	# for displaying in the shell. Takes 1 arg: self. Returns a string.
__add__	# vector + vector. Takes 2 args: self and a vector. Returns a new vector
__sub__	# vector – vector. Takes 2 args: self and a vector. Returns a new vector
__mul__,	
__rmul__	# possibilities: vector*float or float*vector (scalar product) or vector*vector (dot product).
	# Get it to do just one of the product's first, then use introspection to do both
__eq__	# vector == vector. Takes 2 args: self and a vector. Returns True or False.
magnitude	# magnitude of the vector. Takes 1 arg, self. Returns a float
unit	# conversion to a unit vector. Takes 1 arg, self. Scales the vector by the inverse of
	# the magnitude (1/magnitude). No return value. Raises ValueError if magnitude is 0
	# with message "Cannot convert zero vector to a unit vector".

Write a main function that tests the functionality of all methods in your vector class. Be sure to include a test showing that the unit method raises an exception when applied to a zero vector.