**Lab: Vector class (operator overloading)**

**The Problem**

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

**Some Background**

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.

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 the square root of (x2 + y2) .

**Your Tasks**

Make a vector class. Provide the operators

* \_\_init\_\_ # constructor, takes 3 args: self, x, y. No return
* \_\_str\_\_ # for printing, takes 1 arg self. Returns a string
* \_\_add\_\_ # vector + vector. Takes 2 args, self and vector. Returns a new vector
* \_\_sub\_\_ # vector – vector. Takes 2 args, self and vector. Returns a new vector
* \_\_mul\_\_ # two possibilities. vector\*integer or vector\*vector (dot product). Get it to do just one of the two at first,

# then see if you can use introspection to do both

* magnitude # magnitude of the vector. One arg, self. Returns a float