

420-LCU-05 Programming in Python - Lab Exercise 12

April 28, 2022

1 Goals for this lab

- More practice with basic Object Oriented Concepts
- Understanding and extending a class definition.
- Creating objects and calling class methods.

2 General Instructions

- Your submission for this lab should be in 1 python file.
- Code for class methods **to be added inside the class**. Include a doc string for each method you define.
- The code for the 8 items of **section 5** is written outside the class (no indentation). In a comment line include the number for each item.

3 class vector

Consider the following basic class definition intended to model a vector in n dimensions. a vector can have 0 to n components. Carefully, examine the class definition given before moving on to the next section.

```
class vector(object):
    '''A class that represents a simple vector in 0 or more dimensions.'''
    def __init__(self, comp=[]):
        self.components = comp
        self.size = len(comp)

    def __repr__(self):
        vector_info = "vector("
        if len(self.components) > 0:
            vector_info += str(self.components) #create a string from list
        return (vector_info + ')')
```

4 Class Methods

1. Provide the code for a method, such that 2 vectors can be compared correctly using the == operator. Two vectors a and b are equal if they have the same number of components and all of their components are equal. This method returns True if both vectors are equal and False otherwise. **Hint:** This method has to have a special name in Python
2. Modify the method __repr__ to print the size of the vector (in addition to the other info).
3. Write the code for a method named, magnitude to calculate the magnitude of a vector v with any number of components. Magnitude of a vector = square root of (x*x + y*y + z*z + ...). Your code should work for a vector of any length.
4. Write the code for a method named, anyZero. For a given vector, anyZero returns True if the vector has any component with value zero. The method returns False otherwise.
5. Write the code for a special method that can add 2 vectors: v1 & v2. The method returns a new vector. Note that the length of the new vector will be equal to the length of the longer vector (between v1 & v2).

5 Writing code to use the class vector

1. Write code to create a vector: `v0` with no components. Watch for default parameter. Write code to print `v0`.
2. Write code to create 3 vectors: `v1` with components 3,5. `v2` with components 1,2,0. `v3` with components 2,4,5,6.
3. Write code to print `v0`, `v1`, `v2` and `v3`.
4. write code to print the magnitude for `v3`.
5. Using the class method in 1 above, write the code to check if `v1` and `v2` are equal.
6. Write code to check if `v2` and `v3` have the same magnitude.
7. Write code to check if `v2` has any zero components.
8. Write code to add the 2 vectors `v2` and `v3`.