CS29006: Software Engineering Lab – Class Test II
[For roll numbers ending with even digit]
Spring 2022

Important instruction for uploading your answers in moodle: The question comes with a file 'Vector.py' that contains partial code. You will be modifying this file to write your answers. After you complete writing your code, rename the file to "Vector_<ROLL_NUMBER>.py" where 'ROLL_NUMBER' should be replaced by your actual iit_kgp roll number. You should submit this file in moodle. Any deviation from these instructions will bear 0 marks.

We provide an implementation of a **Vector class** in the attached file **Vector.py**, representing the coordinates of a vector in a multidimensional space. For example, in a three-dimensional space, we might wish to represent a vector with coordinates 5, −2, 3 . Although it might be tempting to directly use a Python list to represent those coordinates, a list does not provide an appropriate abstraction for a geometric vector. In particular, if using lists, the expression [5, −2, 3] + [1, 4, 2] results in the list [5, −2, 3, 1, 4, 2]. When working with vectors, if u = 5, −2, 3 and v = 1, 4, 2 , one would expect the expression, u + v, to return a three-dimensional vector with coordinates 6, 2, 5.

We therefore try to define a Vector class, in Vector.py, that provides a better abstraction for the notion of a geometric vector. Internally, our vector should rely upon an instance of a list, named _coords, as its storage mechanism. By keeping the internal list encapsulated, we should be able to enforce the desired public interface for instances of our class.

In Vector.py, you need to define the following magic functions which should work as specified below:

Note that your code needs to check for dimensionality mismatch and out-of-bounds situations wherever applicable and display proper error messages.

Question 1. [10 Marks]

a) [1 Mark] __len__(self) :This function should be able return the dimension of the vector
b) [2 Marks] __getitem_(self,j) : Function should be able return the jth coordinate of the vector
c) [2 Marks] __setitem__(self,j,val) : Function should be able to set the jth coordinate of vector to val
d) [3 Marks] __add__(self,other) : Function should be able to return sum of two vectors
   Example : If two vectors are u =[5, -2, 3] and v = [1, 4, 2] then, output of u + v = [6,2,5]
e) [1 Mark] __eq__(self,other) : Function should be able to return True if vector has same coordinates as other
f) [1 Mark] __str__(self) : Function should be able to return the string representation of a vector. For example, the string representation of a vector with coordinates 4, 7 and 5 is "<4, 7, 5>"