Assignment 2

(Due date for assignment is Friday November 16th,4:00 p.m. ., Late date Monday November. 19th, 4:00 p.m.)

This assignment may be completed with a partner. Sorry, no groups of 3 or more.

In Machine Learning it is often required that vectors be used. These typically can represent a large array of doubles of varying sizes. Your job will be to create such a class in java (java has a vector class, we will create our own), based on the interface specification given and the defined constructors.

A vector will be non-mutable; and will define a default vector as an empty data structure, with a size of zero length.

Constructors

```
Vector()  //empty
Vector(int size, double D)  //creates a vector of size – size
                            // with elements initialized to D
Vector(double[] D) //creates a vector initialized to array D
Vector(int[] I)  //creates a vector initialized to array I
```

Methods:

```
Vector append(double[] doubleArray)
Vector append(int[] intArray)
Vector append(Vector V)
Vector append(double aDouble)
Vector clone()         // returns a clone of this
Boolean equal(Vector V) //this and V are the same
Int getLength()               //returns number of elements in this
Double getValue(int i)  //returns the value  this[i]
Vector add(Vector V)     //add this to V, returning a Vector the same size as this
Vector add(double aDouble); //add aDouble to every element of this
Vector sub(Vector V) //sub this – V
Vector subV(int l, int r)  //will return a sub vector between the
                           //indices l and r inclusive
Vector Mult(Vector V) //Multiple every element of this by corresponding element in V
Vector Mult(double aDouble)  //Multiply every element of this by aDouble
Vector Normalize()   //returns this as a normalized vector
Double EuclidianDistance(Vector V) //returns the Euclidian distance between this and V.
```

Junit testing is a testing framework for Test First Development. This is an educational assignment, in that you will research and learn the Junit test suit and apply it to the class definition above. The below link is a tutorial for Junit. Its installation, and configuration to run an automated test on a code project. All of which you will be responsible for.

https://www.guru99.com/junit-tutorial.html

As part of the assignment you will be required to code the above class, integrating the Junit testing as you go. That is, you will also develop a test class under Junit to test the framework of your class as you implement the constructors and methods. When you are finished you should be able to run the test and verify that all cases have been covered, and that your class is sound.

What this assignment is not. Turning in a working class without any test suit will earn you a big fat zero. It is not about writing a java class, it is about learning Junit testing. It also means your class has to be

correct. Thus, marks are assigned for completeness and thoroughness of the testing and the construction of the test environment.

Output should include the tests which are defined on non implemented methods and constructors. Thus initial tests should fail. You are to define this framework first, version this code as version 1, run it and save the version and output.

Implement the constructors and methods, show the tests at 1/3 complete, 2/3 complete and fully complete, along with the versions which match the tests. How about version 2.0, 3.0, and 4.0 as the final version.

Marking will be based on the quality of the submission which consists of the versions and test output as described above. That is, teams which submit a high quality assignment will be distinguished from the "night before it is due" submission.

Because of the size of this assignment you are given 3 weeks to complete it. Please start early.

Submission

- Your submission should be contained in a large (8.5 inch x 11 inch) envelope.
- **Cover Sheet** completely filled out, available from: " http://www.cosc.brocku.ca/forms/teamcover" **Note**: your assignment will not be marked unless one is submitted with the assignment on the assignment due date. This should be stapled to the outside of the envelope.
- Print out of all code and all tests which have been run on your code.
- Clearly identify the code version and test output version.

End