**CS 010 - Introduction to Computer Science I**

**Lab 9 - Vectors: Standard Deviation**

**Before tackling this lab, you should have completed:**

Zyante Chapter 6 and at least some corresponding Codelab exercises

Video tutorials: [Module 9 playlist](http://www.youtube.com/playlist?list=PLTTJbxrH72A3-ixZ747kMbMs07AX2zr4I)

**Collaboration policy:**

Collaboration on these lab exercises is strongly ENCOURAGED.  
Read the full policy at: [Full Collaboration Policy](https://docs.google.com/document/d/1WyzL3qvKLrC1UCRf178b_wYWQmEZlhDObFNFb79U63I/edit?usp=sharing)

**Lab Objectives**

* Use vectors in functions

You may need to review the concept (and the equation!) for calculating the mean and the standard deviation of a set of values.

* Lengthy (but entertaining!) YouTube video on standard deviation: <http://www.youtube.com/watch?v=dq_D30kyR1A>
* Basic examples: <http://en.wikipedia.org/wiki/Standard_deviation#Basic_examples>
* Quick Reference: <http://standard-deviation.appspot.com/>

You will be given two functions and must build the third to get your lab point. From within your cs010\_practice directory, type "git pull" and hit enter in a terminal or download+upload the lab9.cpp starter file from the lab 9 Google drive directory.

The three functions for this lab: fillVector, average and standardDeviation.

1. Each of these three functions will perform the task implied by the name.
2. Each will take a vector of doubles (by value? by reference? by const reference?)
3. None of the functions should contain output.

**fillVector** populates the vector with values acquired from input. Input redirection should be utilized in testing as you will not know how many values will exist, simply that each will be separated by whitespace.

**average** calculates and returns the mean of all the values in the vector.

**standardDeviation** calculates and returns the standard deviation of the values in the vector.

We recommend that you build a simple test harness that demos the functionality of each function to your satisfaction. R’Sub will use its own test harness, but **you still need to understand how to test!**