CST8219 – C++ Programming

Lab 8

Introduction:

The goal of this lab is to creating a Java GUI, and calling a C++ function using the GUI.

Reference:

Week 8 Powerpoint materials on Brightspace. There are many reference websites at the end of the powerpoint slides.

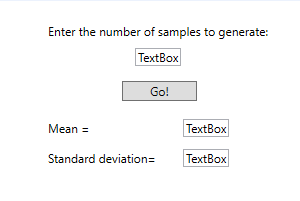
Steps:

1. Create a new directory called Week8 for this lab. Create a Java GUI that has a text field saying: “Enter the number of samples to generate:”. You should then also have an edit text that lets users type in an integer, followed by a button that says “Go!”
2. When the user clicks on the Go button, read the number that the user has typed, and create an array of int[] in Java, and initialize it with random numbers. Use Java.util.Random as your random number generator. It has a nextInt() function that you should call repeatedly in a for loop. Once the array is generated, you will pass the array to two C++ functions to calculate the mean, and standard deviation of the array. The function signatures in Java should be:

***public native double calculateSTDDev( int [] numbers );***

***public native double calculateMean(int [] numbers );***

1. You should then have two text fields that show the results of what the functions returned:



1. If you use Java 1.8, use the **javac** command to compile your Java class file, and then use the **javah** command to generate the .h header file. If you use Java 1.9 or newer, use **javac -h** to generate the header file. You must then create the .cpp file that implements the function declarations that are created.
2. Then create a CMakeLists.txt file that will compile a library instead of an executable. Normally, you would write a line: add\_executable(MyProjectName file1.h file1.cpp file2.h file2.cpp …etc). This time, change that line to add\_library(Week8 headerFile.h headerFile.cpp) , but change the filename from headerFile to whatever file the **javah** program generates.

***Submission***: Create a zip file containing everything in your week8 directory and submit it on Brightspace. Make sure it includes your Java gui class, the .cpp and .h, and your CMakeLists.txt file

Marks: (total of 10)

The Java Gui has a text input field for the number of samples, and a button to start the computation +1

Your Java class has 2 public native functions declared +2

Your C++ header file has 2 functions generated by javah +1

Your C++ implementation (.cpp) file computes the mean and stddev +2

The results are correctly displayed on the gui +2

Your CMakeLists.txt file builds a library instead of an excutable file +1

Your Java class loads the correct library generated by your CMakeLists +1