CST8219 – C++ Programming
Lab 8

Introduction:
The goal of this lab is to creating a Java Text Menu Interface, and calling a C++ function.

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 Text Menu that asks the user to type in an integer: "Enter the number of samples to generate:". When the user enters in a number, read the number and create an array of int[] in Java, and initialize it with random numbers between 0 and 100. Use Java.util.Random as your random number generator. Use the int nextInt(int upperLimit) function to generate a number between 0 and 100 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);

2. You should then output the mean and standard deviation that you calculated to the System.out console. Since the numbers are uniformly distributed between 0 and 100, the mean should be close to 50, and the standard deviation should be close to sqrt(10000/12), or 28.8.

3. 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.

4. 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