CS264 Laboratory Session 2

Dr. Edgar Galván

Tuesday October 9th 2018

Deadline: All solutions to be submitted by 6pm Tuesday 16th October 2018

1 Lab objectives

In this lab you will continue on from last week by solving (slightly) more complex problems in C++. In particular, you will be required to write programs that make use of the features of C++ that we covered since the last lab.

Learning Outcomes
By the end of this lab sheet you should be able to write C++ programs that make use of (i) vectors, (ii) template classes, (iii) typedef, (iv) sort, (v) setprecision.

2 Exercises

For each of the problems given below write a C++ program that provides a solution. Each box provides a filename to use (or in certain cases multiple filenames). Please ensure that you use those filenames.

> **Step 1.1:** For this week's exercises you should create a sub-directory off of the top level `Labs` directory from last week called `Lab2`. Once you have created your first file within that directory you should add it to your repository. All your work for this assignment should go into that directory. Be sure to commit at least after each exercise with appropriate commit messages.

Remember:

- **Add source files to your git repository and commit changes regularly.**
- **All commits should be accompanied by messages that would allow a lecturer or demonstrator to understand the purpose of the commit.**
- **Comment your code.**
- **Use proper indentation for function and control structures.**

Recall that you can compile a cpp program as follows: g++ -o nameOfProgram.o nameOfProgram.cpp

Reference Accelerated C++ Practical Programming by Example. Koenig and Moo.*

Exercise 1: *Compile, execute and test the last program of W2,L2 ("Computing median"). You should save the source in a file called `exercise1.cpp`.

Exercise 2: *Write a program to report how many times each word appears in its input. Hint: use the sort function. E.g.,
Input:
maynooth university UNIVERSITY maynooth such
Output:
UNIVERSITY appears 1 times
maynooth appears 2 times
such appears 1 times
university appears 1 times
You should save the source in a file called `exercise2.cpp`.

Exercise 3: Write a program to report the length of the longest and shortest string in its input. You should also display the word e.g.,
Input:
as e asd edasdasd ekdiaisaskdL DKSAAA
Output:
smallest word: e of size 1
largest word: ekdiaisaskdL of size 12.
You should save the source in a file called `exercise3.cpp`.

Exercise 4: Write a program to report, in reversed order, strings in its input along with its sizes. E.g.,
Input:
maynooth university a zebra bat
Output:
zebra of size: 5
university of size: 10
maynooth of size: 8
bat of size: 3
a of size: 1
You should save the source in a file called `exercise4.cpp`.

Exercise 5: *The average-grade computation discussed in W2, L2 (Batches of data - average) might divide by zero if the student didn't enter any grades. Division by zero is undefined in C++, which means that the implementation is permitted to do anything it likes. What does your C++ implementation do in this case? Rewrite the program so that its behavior does not depend on how the implementation treats division by zero. You should save the source in a file called `exercise5.cpp`.