Object-Oriented Programming in C++

Lab Exercise 2 (2%)

Objective

The objective of this exercise is twofold: first to get a basic understanding null-terminated strings (also referred to as c-style-strings or simply c-strings) and, second, to get a hands-on practice in writing well-behaved object that use pointers internally. *You can work on this exercise in a group of up to two people.*

Description

There are two parts to this lab exercise. In the former you will acquire fundamental knowledge of null-terminated C-style strings and, in the latter, you build on your new-found knowledge and implements a well-behaved C++ class that implements a C++ string-like objects using C-style strings internally.

PART I

C++ has two ways of handling strings: either as a *string* object or as a null-terminated c-style string. It is almost always preferable to use former way of handling strings, however, there are occasional uses of the latter. An all-around programmer needs therefore to understand how they work.

Read the discussion of null-terminated c-style strings at:

- https://www.tutorialspoint.com/cprogramming/c_strings.htm
- https://www.prismnet.com/~mcmahon/Notes/strings.html

When you are done call the TA over and answer the following questions?

- How do you find the length of a c-string?
- How many bytes does the c-string "T-528-HLUT" occupy in memory (assuming each character can be stored in one byte)?
- How do you test if two c-style strings are identical?
- Name two scenarios in C++ when c-style strings cannot be avoided?

PART II

A file *my_str.h* is provided, with a declaration for a class *MyStr*. You are **not** allowed to modify this file. Your task is to provide the definition of the class (in a file named *my_vector.cpp*), that is, the implementation of the asked for functionality. There is also a main program provided for testing purposes, which you can modify as you see fit. In MySchool, hand in the *my_str.cpp* file (only) by the end of the lab.

Good luck! ☺