

TCES 203
Programming Practicum
Assignment 3 – Structures
15 Points

This assignment tests your understanding of concepts covered in the course dealing with Structures, Arrays and Pointers.

Define a student structure with a full name, hobby and student id. A student id is a 5-digit integer. The program should provide the following functions:

- A function that gets input and populates the array of students. The number of students should be passed as to the function to determine the size of dynamic allocation. Use C++ memory allocation operators, new and the corresponding delete to release memory.
- A function to display the information about one student.
- A function to display the information about an array of students. Use the same function name as the previous to show the concept of function overloading.
- Functions that allow us to modify the contents of a student given their id. Only name and hobby can be modified.

In addition, provide a main function that tests each function thoroughly. Use dynamic memory allocation and make sure that there are no memory leaks (Use valgrind to check for leaks). Use a header file for the definition and a corresponding .cpp file for the implementation. Pass structure variables by address rather than by value and use const when the function shouldn't modify the parameters. Do not use any global variables other than the structure definition.

Submission and Grading:

Submit student.h, student.cpp and main.cpp under the Assignments section of the course web page as one zip file with the name Student<LastName>.zip (Both last names in the case of pairs, exclude the <>).

There will be points taken off for not following the conventions listed in this document regarding submissions, outputs and naming conventions.

You are required to properly indent your code and will lose points if you make significant indentation mistakes. See the textbook for an explanation and examples of proper indentation.

Give meaningful names to functions and variables in your code. Localize variables whenever possible -- that is, declare them in the smallest scope in which they are needed.

Include a comment at the beginning of your program with basic information and a description of the program **and include a comment at the start of each function**. Your

comments should be written in your own words and not taken directly from this document. Write comments within functions to explain the flow or any obscure code. Provide comments for the prototypes as well as for the structure definition. Make sure that every file has a header comment including the .h and the main.cpp files.

You should include a comment at the beginning of your program with some basic information and a description of the program, as in:

```
// Menaka Abraham  
// 3/30/15  
// 203  
// Assignment #1  
//  
// This program will...
```