TCES 203
Programming Practicum
Assignment 4 – Classes
15 Points

This assignment tests your understanding of concepts covered in the course dealing with classes. Redo the previous homework solution to use a Class instead of a structure.

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

- A default constructor, copy constructor and an overloaded constructor that takes all three parameters.
- A function to display the information.
- Mutator functions that allow us to modify the contents of a student. Only name and hobby can be modified.
- Accessor functions that allow us to get the individual pieces of information – id, hobby or full name.

In main function, create and populate an array of student objects dynamically. All user input must be from the main function. Use constructors to pass the information and the corresponding functions to test all the member functions. Use C++ memory allocation operators, new and the corresponding delete to release memory. Use valgrind to check for leaks. Use a header file for the definition and a corresponding .cpp file for the implementation.

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 functions as well as for the class 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...
```