Project 2 – Grade Report STL (Part B)

For this part of the project, you will add an additional function that prints out all students in alphabetical order by last name.

IMPLEMENTATION

Your project should be working correctly by now, and you certainly want to avoid any changes that could cause your application to crash. For this reason, you are going to create a class, **StudentListChild**, that **inherits** from **StudentList**, and you will write your function in the new class.

How to proceed:

- **StudentList.h**
    - Change the member variable of the class to **protected**, so you can have **direct access from the derived class.**
- Create the new class **StudentListChild** that **inherits** from **StudentList**
    - **StudentListChild.h**
        - There are **no** member variables in this class, but you still need to add a **constructor** and a **destructor**.
        - Function **printStudentsInOrder**
            - This is a **void** function with **no parameters**.
    - **StudentListChild.cpp**
        - Both constructor and destructor will remain **empty**.
        - **Function printStudentsInOrder**
            - There are several algorithms that would make this function efficient (an O(log n) versus the actual O(n) you will have here), but the idea is to play with the STL library and its classes. The algorithm you will use is as follows

```
Algorithm printStudentsInOrder()
        For each student in the vector
                Insert the last/first name, and ID into a map
        For each item in the map
                Print out the last name, first name, and ID
```

            - Since you are creating a **map**, you will need to **include the map class** in your **StudentListChild.h** file.
            - A **map** contains **two pieces of data**, where the order is determined by the first piece of data. You want to print the roster in **alphabetical order** by **last name**; therefore, you should **combine last and first name** and insert it in the map as the **first parameter**, and then insert the **ID** as the **second parameter**.
            - The format of your output should be as the one shown below, which prints the ID first.

```
145632 - Bear, Baloo
654321 - Cricket, Jiminy
765345 - Duck, Daisy
789456 - Duck, Donald
123456 - Mouse, Mickey
159753 - Mouse, Minnie
345678 - Pan, Peter
112233 - Pelekai, Lilo
645915 - Pistoles, Panchito
324598 - Rabitt, Roger
```

MAIN.CPP

You will need to change the **Main.cpp** file to add a new selection in your menu:

`7: Print roster`

Obviously, the selection to **exit** the program will change to 8.

OBJECTS OF THE CHILD CLASS

To use your new function, you cannot use objects of the **StudentList** class; therefore, you will need to change a few sections so that you use only **StudentListChild** class objects. The only files that will need modifications are:

- Main.cpp
- InputHandler.h
- OutputHandler.h

DUE DATE

You will turn in your project at the beginning of class. I will go over the details again, at the beginning of class, in case you do not remember how to turn in a project in the **Q drive**.

Due date:

- **MW class:** Monday, November 23
- **TTh class:** Tuesday, November 24