**CSCI 4957/5957-002 – C++**

**Fall 2014 - Phil Pfeiffer – Assignment 9 – introduction to classes and inheritance, part 2**

**Due: Thursday, 4 December, BEFORE class (HARD deadline)**

**Terms:**

* **graduate students, as individuals**
* **undergraduates (Eric and Christopher) may submit together or as individuals, as they choose**

**Value: 6 points, as shown below.**

***As usual, I recommend using*** [***www.cplusplus.com***](http://www.cplusplus.com) ***'s search function to obtain information on STL functions.***

**Deliverables:**

* Your codes, along with screenshots of your main program's output.

***Background.*** This assignment asks you to reframe the student record and student record collection codes, using the reframed grade line and grade line summary codes and some additional classes. Your starter code for this assignment will be your grade line and grade line summary codes from assignment 8, along with your student-record-related codes from assignment 7 and lab 11. I've posted my codes for assignment 7 and lab 11 to the website. You may, however, use your own codes for these problem sets, if you wish.

1. (2 points) Redo the logic for ***studentRecord.h*** and ***studentRecord.cpp.***  as follows:

* (1 point) Create a first class, a ***MeasurableOptions*** class, to replace the ***StudentMeasurableTypes*** namespace. This class should consist of the following entities:
* Two public definitions:
* An enum class, ***measurable\_t,*** for defining the three types of measurables to support: i.e., homeworks, quizzes, and tests.
* A ***typedef*** that makes ***measurable\_name\_t*** a synonym for ***std::string***.
* One private data member:
* An entity of type ***vector<measurable\_t>***
* One constructor:
* A default constructor that pushes each ***measurable\_t*** in turn into ***vector<measurable\_t>***
* One accessor:
* A function of type ***const summary\_option\_t* 🡪 *measurable\_name\_t*** that returns a name string for each enum in ***measurable\_t***. Note: I implemented this function using a locally defined, 3-element static map from measurable enums to strings.
* One pair of ***const*** iterators:
* A cbegin()- and a cend()- style iterator for the vector of measurable options.
* A null destructor
* (part of above) A ***static const*** instance of an object of type ***MeasurableOptions,*** in ***StudentRecord.cpp,*** for use by the module's instances of ***operator<<.***
* (part of above) An overloaded ***operator<<*** for outputting a single value of type ***MeasurableOptions::measurable\_t.***
* (1 point) Redo the ***StudentRecord*** class to use the ***MeasurableOptions*** class. Your revisions, which should on the whole be modest, should incorporate the following changes:
* Use the ***= delete*** idiom to disable the default constructor and single argument forms of the ***StudentRecord*** constructors;
* Use the overloaded ***operator<<*** to output ***MeasurableOptions::measurable\_t*** values in the overloaded ***operator<<*** for ***StudentRecord*** objects.

1. (2 points) Redo the logic for ***studentRecordCollection.h*** and ***studentRecordCollection.cpp.***  as follows:

* (1 point) Create a first class, a ***SortOptions*** class, to replace the ***\_sort\_direction\_to\_name*** map. This class should consist of the following entities:
* Two public definitions:
* An enum class,***sort\_option\_t,*** for defining the two sort options to support: i.e., ascending and descending.
* A ***typedef*** that makes ***sort\_option\_name\_t*** a synonym for ***std::string***.
* One private data member:
* An entity of type ***vector<sort\_option\_t>***
* One constructor:
* A default constructor that pushes each ***sort\_option\_name\_t*** in turn into ***vector<sort\_option\_t>***
* One accessor:
* A function of type ***const sort\_option\_t* 🡪 *sort\_option\_name\_t*** that returns a name string for each enum in ***measurable\_t***. Note: I implemented this function using a locally defined, 2-element static map from sort option enums to strings.
* One pair of ***const*** iterators:
* A cbegin()- and a cend()- style iterator for the vector of sort options.
* A null destructor
* (part of above) Create an overloaded ***operator<<*** for outputting values of type ***SortOptions*** ***::sort\_options\_t***. Implement this function using a ***static const*** instance of an object of type ***SortOptions.***
* (1 point) Redo the ***StudentRecordComparator*** class, according to the following requirements:
* Use the ***= delete*** idiom to disable the default constructor and single argument forms of the ***StudentRecordComparator*** constructors;
* Redo ***StudentRecordComparator***'s other functions using the ***SortOptions*** class.
* Reimplement ***operator<<*** for ***MeasurableOptions::measurable\_t*** and ***GradeLineSummaryOptions::summary\_option\_t.*** For this problem, I ***require*** that ***operator<<*** be reimplemented without additional copies of the options classes, let alone direct calls to their name() methods. Rather, do the following:
* Allocate two local variables of type ***std::strstream*** and ***std::string***;
* Stream the ***StudentRecordComparator*** object's sort direction, measurement type, and summary type member data items into the ***strstream*** object;
* Stream the ***strstream*** object's content into the ***string*** object;
* Return the ***string*** object.
* (no further credit) The ***StudentRecordCollection*** class should not require revisions. I recommend, however, that you confirm this.

1. (2 points)

Complete the program by revising ***main.cpp*** and getting it to run with the updated codes. The only changes that should be required involve creating instances of MeasurableOptions, GradeLineSummaryOptions, and SortOptions objects and using these to iterate over measurables, summary options, and sort directions.