CSCE 240 – Programming Assignment Four

Due: 11:59pm on Wednesday, November 1

Purpose – Implement two classes

Class 1

Create a *Weight* class that holds the value and units of a weight in private double and string data members, respectively. The class should allow for the units to be ounces, pounds, grams, or kilograms. The class will contain a constructor, a *ConvertUnits* member function, and accessor and mutator functions for the private data members. You will also need to overload the stream insertion operator (<<) for the *Weight* class.

Read the comments in the supplied *weight.h* header file for more detailed requirements. Also, review initial tests for constructor, mutators, and accessors in *testweight1.cc*. Review initial tests for the *ConvertUnits* function in *testweight2.cc*. And review initial tests for the stream insertion operator in *testweight3.cc*.

If you place all of the attached files in the same directory, you can run the initial tests with the commands

make testweight1
make testweight2
make testweight3

You are strongly encouraged to create more rigorous tests.

Class 2

Create a *WeightRange* class that has a smallest *Weight* object and a largest *Weight* object as private data members. The class will include two constructors, an *InRange* member function, a *Width* member function, and accessor and mutator functions for the private data members.

Read the comments in the supplied *weightrange.h* header file fore more detailed requirements. Also, review initial tests for constructor and accessors in *testweightrange1.cc*. Review initial tests for the mutator functions in *testweightrange2.cc*. Review initial tests for the *InRange* function in *testweightrange3.cc*. And review initial tests for the *Width* function in *testweightrange4.cc*.

If you place all of the attached files in the same directory, you can run the initial tests with the commands

make testweightrange1
make testweightrange2
make testweightrange3
make testweightrange4

You are strongly encouraged to create more rigorous tests.

Specifications

- Add all code for the definition of the *Weight* class to the attached *weight.h* header file.

- Include all of the necessary code for the *Weight* class, including the implementations of the public member functions and the overloaded stream insertion operator, in the attached *weight.cc* source file.
- Add all code for the definition of the *WeightRange* class to the attached *weightrange.h* header file.
- Include all of the necessary code for the *WeightRange* class, including the implementations of the public member functions, in the attached *weightrange.cc* source file.
- You will submit a zip file (only a zip file will be accepted) containing *weight.h, weight.cc, weightrange.h* and *weightrange.cc* to the assignment in Blackboard.
- Source files must compile and run on a computer of the instructor's choosing in the Linux lab (see your course syllabus for additional details).
- Your programming assignment will be graded with modified versions of the test files *testweight1.cc, testweight2.cc, testweight3.cc, testweightrange1.cc, testweightrange2.cc, testweightrange3.cc,* and *testweightrange4.cc*

Grade Breakdown

Style *weight.h*: 0.25 points

Style *weight.cc*: 0.25 points

Style *weightrange.h*: 0.25 points

Style *weightrange.cc*: 0.25 points

Documentation: 1 point

Clean compilation of *weight.cc*: 0.5 points

Clean compilation of *weightrange.cc*: 0.5 points

Weight class passes instructor's modified *testweight1.cc* tests: 1 point

Weight class passes instructor's modified *testweight2.cc* tests: 1 point

Weight class passes instructor's modified *testweight3.cc* tests: 1 point

WeightRange class passes instructor's modified *testweightrange1.cc* tests: 1 point

WeightRange class passes instructor's modified *testweightrange2.cc* tests: 1 point

WeightRange class passes instructor's modified *testweightrange3.cc* tests: 1 point

WeightRange class passes instructor's modified *testweightrange4.cc* tests: 1 point

The penalty for late program submissions is 10% per day, with no submission accepted after 3 days.