CS 210 – Fundamentals of Programming I
Fall 2006 – In-class Exercise for 11/9/06

Name(s):

(15 points) Complete this exercise in pairs. Write the answers to the written part in this assignment sheet and turn it in with a print out the program when you are done. The purpose of this exercise is to work with overloaded and friend operators. Here is a problem statement, analysis, and design for today’s exercise. It is a driver program that tests a full implementation of the Temperature class introduced in lecture. Much of the analysis and design of the Temperature class is in the handout An Analysis and Design Guideline for Classes, so it is omitted here, as is the analysis and design for the function PrintMenu.

Problem Statement
Write a program that tests the operations of the Temperature class. It should be menu-driven and allow the user to test the operations until they want to quit.

Main Program
Analysis - what data is constant, input, computed, or output?

<table>
<thead>
<tr>
<th>Objects</th>
<th>Type</th>
<th>Kind</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A temperature</td>
<td>Temperature</td>
<td>variable</td>
<td>theTemperature</td>
</tr>
<tr>
<td>Second temperature</td>
<td>Temperature</td>
<td>variable</td>
<td>secondTemperature</td>
</tr>
<tr>
<td>Menu choice</td>
<td>char</td>
<td>variable</td>
<td>choice</td>
</tr>
<tr>
<td>Degree input</td>
<td>double</td>
<td>variable</td>
<td>degrees</td>
</tr>
<tr>
<td>Scale input</td>
<td>char</td>
<td>variable</td>
<td>scale</td>
</tr>
</tbody>
</table>

Design - what are the steps to solve this problem?

1. Read in theTemperature
2. Print menu using PrintMenu()
3. Read in choice
4. While choice is not 'Q' and is not 'q' do
   4.1. Select on choice
      4.1.1. Case 'A', 'a'
        4.1.1.1. Construct theTemperature = Temperature()
4.1.2. Case 'B', 'b'
   4.1.2.1. Read in degrees and scale
   4.1.2.2. Construct theTemperature = Temperature(degrees, scale)
4.1.3. Case 'C', 'c'
   4.1.3.1. Display theTemperature.GetDegrees()
4.1.4. Case 'D', 'd'
   4.1.4.1. Display theTemperature.GetScale()
4.1.5. Case 'E', 'e'
   4.1.5.1. Display theTemperature.Fahrenheit()
4.1.6. rest of cases are similar
4.2. Print menu using PrintMenu()
4.3. Read in choice

0. Create a new project, then download files inclass22.cpp, temperature.h, and temperature.cpp from the course webpage under today's date and add it to your project. Build and run this program. Test it out on various data until you are satisfied that you understand how it works.

1. (6 points) Study the Temperature class definition in temperature.h. Answer the following questions.

   a. What is the purpose of making a function a friend of a class?

   b. How/where are friend functions declared?

   c. List the overloaded operators defined for the Temperature class. How is operator- different from the rest of the operators?
d. What is the purpose of the `const` at the end of member function prototypes (e.g., like that at the end of `Fahrenheit`)?

e. What is the purpose of the `const` in a parameter list (e.g., `operator== (const Temperature & leftOperand, const Temperature & rightOperand)`)?

f. Suppose we wanted to add `operator>` to the Temperature class as a friend function. What would its prototype be?

2. (1 points) Study the source file `temperature.cpp`. Answer the following question:

   a. The function headers for operators `==`, `+`, `<`, and `>` do not have the `Temperature::` prefix before their function names. Why not?

3. (2 points) Study the main program in `inclass22.cpp`. This program is a menu-driven program that can be used to test the Temperature class. Answer the following questions:

   a. Which variables in the main program are Temperature objects?

   b. How is the function `operator==` called?
4. (6 points) Modify this program in the following ways:

   a. Add `operator>` to the Temperature class as a friend function. This operation should behave as
      one would expect. (You need to provide a prototype in the class definition in `temperature.h`
      and a definition in `temperature.cpp`.)

   b. Add code to the main program to test this new operation. I.e., add a new case in the menu and
      switch statement to exercise this new function appropriately.

Rebuild and test your program until you are confident that it computes the correct result.

When you have completed this exercise, print out all three files and turn them in with one copy of this
exercise sheet with your answers to the questions.