

**CS 1160 Introduction to Computer Science and Programming Methods**  
**Programming Assignment 1**  
**Due Friday, January 14, 2005**

**Three Easy Pieces**

For this assignment you will write, compile, and test three small programs. This exercise should allow you to become familiar with the use of a C++ compiler in addition to giving you practice in elementary aspects of programming in C++.

**The Programs**

1. **C\_to\_F**: This program should prompt the user to enter a temperature in *degrees Celsius*, and then it should compute and report the corresponding temperature in *degrees Fahrenheit*.

[NOTE: The conversion formula is  $F = \frac{9}{5}C + 32$ .]

2. **F\_to\_C**: This program should prompt the user to enter a temperature in *degrees Fahrenheit*, and then it should compute and report the corresponding temperature in *degrees Celsius*.

[NOTE: The conversion formula is  $C = \frac{5}{9}(F - 32)$ .]

3. **meter\_reader**: This program will emulate the process of computing an electricity bill. It will prompt the user for three items of input

- i. The *current* meter reading (a positive integer),
- ii. The *previous* meter reading (a positive integer), and
- iii. The *number of days* in the billing period (a positive integer).

The *units* represented by the meter readings are **kwh** (kilowatt-hours). Your program will assume that the *cost* of electricity is 8 cents per kwh for the first 300 kwh and 10 cents per kwh for usage in excess of 300 kwh.

Your program should then report

- i. The total usage (in kwh) for the billing period,
- ii. The average daily usage (in kwh per day) for the period, and
- iii. The total cost (in dollars) for the period.

In addition to the basic calculations this program needs to do it should also *validate its input*. That is, it should *check* that the input provided by the user is reasonable. This means

- i. All three integer inputs should be *positive*, and
- ii. The *current reading* should be *greater than* the *previous reading*.

If the input is *bad*, your program should issue an *error message* for the user and *exit*. One good way to exit is to use the command `exit(1)`, which is a statement that *immediately* terminates the program. [Note: In order to use this statement it might be necessary to include the header file `cstdlib` via the command `#include <cstdlib>` at the beginning of your program.

**What to turn in**

Your programs should each include block comments at the beginning in which you report your name, the course number, the programming assignment, the due date, and the nature of the program.

You should turn in *paper* copies of your program source code along with copies of actual runs of the program. The runs of the program must show that you *tested* your program in a manner that would *convince* someone reading the output that the program is working correctly.

In order to prepare the paper copies, it may be necessary to use cut-and-paste into a simple editor, e.g., **notepad**. The paper copy can then be made by printing from the editor. Note that you are on your honor not to use the capabilities of the editor to correct any erroneous output from your program. If such action is detected, it will be considered academic dishonesty.