General Requirements for Programming Project

- ✓ The program code should be formatted for good readability and should include the usual comments. When you submit a printout of source code, make sure you don't miss some parts of lines because each line contains 80 characters at most. Also make sure blocks are indented. The output from a program must be formatted nicely by using formatting manipulators.
- ✓ Comments are used to explain what your program does. They should be useful to help the user or reader understand your program. Generally you should assume the reader or user knows C++. To insert comments into the program, you must include as a minimum: a description of the purpose of this program, due date, your name, course and section number, csci account, and file location. You do this on the top of your program. Also for each function, you must explain what the function does and include pre- and post-conditions.
- ✓ Constants should be declared in the globe declaration area, which is before the main function. By conventions, they are declared by uppercase so that they stand out.
- ✓ A user document describes how to use the program. This document should be typed or word-processed. Include an example of how to run the program, and be sure to describe any requirements that input data must satisfy. Should the grader run your program, he or she will consult this document for directions; if they are not correct, neither is the document. Note that a user should *not* be told anything about how the program is implemented.
- ✓ Test data should be verified and be appropriate. Generally, you should have at least three tests showing that the program does what it is supposed to do. You will design your own test data. On the printout, annotate each data file and test to describe what feature(s) of the program the run is testing.

An Example

```
/*This program prints the tuition at Strange College. Strange charges
$10 for registration, plus $10 per unit and a penalty of $50 for each
12 units, or fraction of 12, over 12.
Due Date: 10/31/2005
Written by: John Smith
Course and Section Number: CSCI 201 (2)
CSCI Account #: cs201301
File Location: /export/home/cs201/cs201300/EXAMPLES/prog4.cpp
*/

#include <iostream>
using namespace std;

// Constant declarations
const int REGFEE    = 10;
const int UNITFEE   = 10;
const int EXCESSFEE  = 50;

// Prototype Declarations
int calculateFee (int firstTerm,
                  int secondTerm,
                  int thirdTerm);
int termFee (int units);

int main ()
{

      cout << "Enter units for first term:  ";
      int  firstTerm;
      cin  >> firstTerm;

      cout << "Enter units for second term: ";
      int  secondTerm;
      cin  >> secondTerm;

      cout << "Enter units for third term:  ";
      int  thirdTerm;
      cin  >> thirdTerm;

      int  totalFee;
      totalFee = calculateFee
                 (firstTerm, secondTerm, thirdTerm);

      cout << "\nThe total tuition is :\t" << totalFee;

      return 0;
}     // main
/*    ================ calculateFee ================
      Calculate the total fees for the year.
         Pre  The number of units to be taken each term: firstTerm,
              secondTerm and thirdTerm
         Post Returns the annual fees
*/
int calculateFee (int firstTerm,
                  int secondTerm,
                  int thirdTerm)
{
      int fee = termFee (firstTerm)
              + termFee (secondTerm)
              + termFee (thirdTerm);
      return fee;
}     // calculateFee

/*    ================== termFee ==================
      Calculate the tuition for one term.
         Pre   units contains units to be taken in the term
         Post  The fee is calculated and returned
*/
int termFee (int units)
{
      int totalFees = REGFEE
                + ((units - 1) / 12 * EXCESSFEE)
                +  (units * UNITFEE);
      return (totalFees);
}     // termFee
```