C++ Programming Lab Problems
Chapter 5 - Looping

Turn in hardcopy of the source code, and hardcopy of output samples for each program. Turn in all programs together in ONE packet. The peer review sheet should be the first page of your packet.

1. Write a program to accept **any number** of grades. Then, print the average of the grades and print the letter grade. (Be sure to use nested if..elses when determining the letter grade.) Drop the lowest grade before you average (IF you have two or more grades.) You choose how to stop accepting grades from the user, but don't make the user tell you how many grades he will be entering.
   Run your program with the 3 data sets which follow:
   88 78 45 82 81 75 75 75 75 45
   99
   63 78 98 65 91

2. Write a program to print the following table. Each row in the table has an integer, its square, its cube, and then its square root to 2 decimal places. Print appropriate column headings. Begin the integers with 50 and end with 75, increasing each by one. Use setw and setprecision to line up the values each column. (Only the square root column should be printed with a decimal point.)

   After all rows have printed, print a "line" and then the totals for each column. **Send your output to a file.**

3. Change your binary conversion program to use the design **using iteration** we discussed in class. Do NOT write 16 if-else statements to handle each bit. This time, use the range 0 to 65,535 (range for unsigned binary, 16 bits). Be sure to check for proper range *before* you attempt to convert the number to binary. After you convert the current number, ask the user if he'd like to give another number to convert.

   Figure out how to write the binary output so that there is a space after every set of 4 binary digits.

   ```
           200 converted to binary :    0000 0000 1100 1000
   ```

   Your sample output should include the following:
   200, 63, 255, 144, -1, 1024, 65000, 100000, and 65535.

4. Write a non-interactive version of program #3. Change it to read any number of integer values from a file and to print the results to a file. Create an input file with the same values listed in #3 and add at least 5 of your own values. Your program should create an output file that gives the original base 10 number and its binary equivalent (or an error message) for each number on the input file. Turn in your source, a copy of the input file, and the output file.