The Five Step Process
"You shall code no line before its time" Joe Allen, Ph.D.

This is a description of the five step process that will be used in designing software for scientific and engineering analysis. It is essential that the first four steps be completed before a single line of source code is typed.

The five steps are

(Baseline Analysis replaces the Pseudocode previously included in this process. It is very important to have a comprehensive list of baseline cases to identify any deficiencies in the solution.)

1 Problem
Each problem statement should be analyzed against the following questions.
   a) What do we need to Find?
   b) What information is Given?
   c) What do we need to Solve?

Find: Determine the specific data or information that is requested.

Given: Identify the information and data that is provided in the problem statement. Is there enough information to generate answers to the request in Find? If not, list the missing information. The Given portion of this three step process breaks into two categories
   Explicit Given: the data and information are stated explicitly in the problem statement
   Implicit Given: Data and information are "well known" so it is not included in the statement. Or, a problem is complex and multiple steps determined by intermediate problems are not explicitly given in the Problem statement (you are expected to provide this as part of your solution so you must break this complex problem into smaller problems).

Solve: The solution to the complete problem may require formulas to calculate numerical data. It might require a logical path for making decisions (i.e. a control structure). Before attempting to define the Solution in the next step, it is essential to clarify the problem complexity by identifying any special cases that will require particular solution procedures.

2 Solution
With the information from 1, define a complete solution. Complete means that all data, information, and procedures required to generate the requested output are explicitly stated. All special cases should be identified. At this point, Baseline Data for the special cases should be documented. Although it will not be analyzed until step 4, it needs to be identified in the Solution.
3 Algorithm
This is the heart and soul of the 5 step process. The solution needs to be broken down into a series of detailed steps. Using the algorithm, someone with no technical background in the problem should be able to generate the output for this problem.

   a) The algorithm should contain sufficient detail to ensure accurate output is generated in the Baseline Analysis.
   b) The algorithm should be clearly stated so that it can be translated into C++ Source Code. The only challenges in the translation process should be knowledge of C++ language and syntax.

4 Baseline Analysis
The baseline analysis generates data that can be used to check for errors in the Source Code when the program is run. If the algorithm is accurate, then well-defined examples will bring to light translation errors from algorithm to source code. In addition, this analysis will demonstrate the completeness of the Solution to the Problem.

5 Source Code
Translating the Algorithm into C++ Source Code will be much easier given steps 1 – 4. Type, compile, link, and run the program. The Baseline Analysis contains the data for the test runs that will be used to validate the code.