Project Guidance

General: The material in this document is applicable to each of the projects assigned during this course. Please retain it for appropriate review and compliance. Your professor's instructions will make clear any exceptions to this guidance. Please ask if you need any clarification.

Grading Options: Projects will offer 2 levels of difficulty.

1. B Level: Appropriate response to requirements at this level will result in the award of a B.
2. A Level: To be awarded an A, project responses must correctly address B and A level requirements.

Any variation on these grading options will be spelled out in the project instructions. At the top of the first page in your written project report you must state clearly the grade level that you are attempting. Note that incorrect or incomplete responses at any level can result in a grade below that attempted.

Submission/Presentation Details: Projects are due at the beginning of the period specified by the instructor and will be collected at that time. Late submissions will accrue a penalty. While computer programs are an important part of projects in this course, they must be preceded and followed by appropriate foundational work and the documents that you submit must communicate this work. Use a word processor to:

1. Present a clear statement of the problem and describe what is needed in the way of a solution.
2. List all facts and indicate their source (if applicable).
3. Present each equation to be used, indicate its source (if applicable), and present sufficient detail and discussion to show that use of this "theory" is appropriate.
4. List and justify all assumptions.
5. Show sample calculations, pertinent data, and correct/consistent units that lead to a clear answer for each different case being investigated. These calculations must provide the reader a step-by-step picture of how results were achieved. Present all of this in a logical, well-organized, and legible format. (See note 1 below).
6. Include a hard copy of the C++ program developed to solve the problem. (See note 2 below).
7. Test the program to see that it is free of both syntax errors and run-time errors. You should further test with enough input to make sure that it will handle all cases being investigated and compare program results with the sample calculations generated in step 5 above in order to verify that they meet the needs associated with the original problem definition. You should test the program with a variety of other input and make a conscious decision about handling (or not handling) each situation.
8. Finally, discuss your results. Summarize your findings, evaluate them in light of your assumptions and tests, and clearly identify any results that appear questionable.

Use headings (problem statement, assumptions, sample calculations, etc) to identify that portion of your report where each of the above requirements is fulfilled. Your written presentation must show a fully documented printout of your source code.

Collaboration/Honor Code: This course is conducted under the honor code provisions outlined in the Bulletin and in the student handbook, The Lair. Projects assigned as individual work are to be accomplished without assistance from anyone. Projects that are assigned as a team effort are to be accomplished without assistance from anyone outside your team. In both individual projects and team projects you are free to discuss details with your EGR 126 instructor. Under no circumstances are you free to use solution materials created by anyone other than you or your teammate. By submitting your work for grade you are declaring that these restrictions have been honored.

Note 1. Sample calculations may be presented using legible handwriting, a word processor, MathCAD, Maple, MATLAB, Excel, Lotus 1-2-3, or any other appropriate tool or combination of tools, as long as the presentation communicates clearly the sequence of calculations specified in the algorithm and demonstrates that these steps lead to correct results. Use double underline to indicate each "answer". In those cases where results are obviously questionable, make appropriate comments.
Note 2. Many engineering problems can be solved without the use of computers and it is important to know when to write a program and when to simply solve a problem "by hand" with calculator, pencil, and paper or other tools. Some projects assigned in EGR 126 may indeed be solved without computer code and you will be penalized if you write an unnecessary program. However, in a number of cases, the learning experiences associated with this course can be enhanced by addressing somewhat simple problems with a computer. Therefore, if project instructions explicitly require you to write a program you must do so. In these instances you should include a paragraph noting the fact that a program is not necessary and you should use the "by-hand" method to verify your results.