Instructor: Jerry C. Hamann  
Rm. 5036 Engineering, Phone 307-766-6321  
email: hamann@uwyo.edu

Office Hours: MTWF 9:30 -- 10:30, MTF 2:10 -- 3:00

Text:  

Grading:  
Midterm Exams 40%
Final Exam 25%
Homework 25%
Quizzes 10%

Grade Breakpoints:
A ≥ 93, A- ≥ 90, B+ ≥ 87, B ≥ 83, B- ≥ 80, C+ ≥ 77, C ≥ 70, D+ ≥ 67, D ≥ 60, 60 > F

Prerequisites: Students should have completed MATH 2205 (Calculus II) and either MATH 2250 or 2310 (Linear Algebra or Applied Differential Equations I) and either COSC 1010 (Introduction to Computer Science I) or ES 1060 (Introduction to Engineering Problem Solving).

Course Description: Introduces students to the fundamentals of practical engineering programming, using specific applications of numerical methods to demonstrate these principles. The use of an object-oriented approach using C++ in an efficient manner is emphasized. Other solution approaches, including C, Python and Matlab, are discussed as appropriate.

Quizzes/Exams: A brief quiz will be given in class each week, typically on Wednesday. Examinations will be given on the days noted on the Course Schedule, unless circumstances dictate a change which will be announced in class. It is not possible to make up a missed quiz or exam without a University Excused Absence.

Assignments: Homework/programming assignments will be given on a near weekly basis, with due dates as shown on the Course Schedule. Assignments must be “turned in” by the assigned time on the specified due date. No credit will be given for late assignments. The tasks involved will include analytical mathematics, design and implementation of C++ programs, programmed computations, and generation of annotated graphical plots. Solutions to the assignments will be posted on the C++ Numerical Methods Course Site available through WyoCourses.

Suggestions: Some recommendations for study which you should consider are as follows...

Don’t miss class. New material is covered each session. If you miss class, you are responsible for discovering the missed material on your own.

Read in advance. The reading assignment topic areas are detailed on the Course Schedule.

Start assignments early. Give yourself some time to consider the problems and determine whether or not you need instructor assistance. Last-minute questions are not a good idea. Start-to-finish time demands of assignments will vary from an hour to several hours and multiple debugging cycles.

Carefully read and follow all instructions. Electronic exchange of assignment solutions will be the norm in this course. You must strictly follow the folder and file naming and coding conventions requested in all assignments (including quizzes and exams, where appropriate). Don’t lose credit for failure to attend to simple things!

Don’t ignore the assignments and quizzes. They comprise 35% of your grade.

Ask questions. This includes during class and during office hours.
Objectives: At the end of the semester, students will be able to:

- Prescribe algorithms for solving computational problems with pseudocode and flow charts which apply the following structured programming constructs: SEQUENCE, IF-THEN-ELSE, CASE, FOR, WHILE, and REPEAT-UNTIL.

- Demonstrate the correctness of algorithms via “hand” computation and graphical summary of solutions for baseline problems.

- Translate pseudocode and flow-chart algorithm specifications into modular, syntactically correct C++ code which conforms to a coding style guide and accurately computes solutions for baseline problems.

- Specify classes using UML diagrams to encapsulate data structures and frequent operations typical of scientific computations.

- Translate UML class diagrams into modular, syntactically correct C++ code which conforms to a coding style guide.


To accomplish these tasks, the student will be required to apply fundamental concepts from mathematics and engineering to formulate models which accurately represent a variety of problem domains. Design, construction and testing of the computational tools will require considerable time spent in algorithm development, authoring of syntactically correct code, and validation of program-generated solutions.

Engineering Notebook Policies: To encourage student development of a personal, professional programming notebook, the quizzes and exams will be closed-book and closed-note with the exception of one reference: the student’s personal, bound engineering notebook. Specifications for an allowable notebook are as follows:

- Notebook format is permanent binding with no removable pages, page dimensions not to exceed A4 in area.

- Outside cover and front inside cover page should clearly identify (1) student owner by name and contact info, (2) course by number and title, and (3) initial date of entries.

- Each internal page should be individually and consecutively numbered, beginning with a table of contents on pages 1 thru 4. Pages should be filled successively, dated and signed.

- All entries must be made in indelible ink or be permanently affixed to each individual page (in the case of computer printouts or similar materials) by careful cutting and sizing then affixing fully, on all edges, with transparent tape (absolutely no staples allowed). Corrections should be made with simple line-outs and dated.

- It is strongly recommended that two specific reference items be placed beginning at page 5: (1) a list of C++ reserved keywords, and (2) a list of C++ operators described in sorted order of operator precedence. This material is readily available for printing from the following website: [http://en.cppreference.com/w/](http://en.cppreference.com/w/)

Notebooks which include any looseleaf (unattached) materials or non-indelible (e.g. pencil) entries may not be used on any quiz or exam (the use of such a notebook will be considered a breach of academic/professional honesty and decorum, resulting in zero credit).

Policies Regarding Assignments / Collaboration / Academic Dishonesty: Students are encouraged to discuss course topics and assignments with one another. However, the assignment solutions, quizzes and exams turned in by each student must consist of that individual’s own work. Failure to observe these expectations will result in sanctions as prescribed in University Regulation 6-802. Possible penalties include a grade of F for the course.

Disability Services: Students with a physical, learning or psychological disability requiring accommodations must register with and provide documentation to the University Disability Support Services (UDSS) in the Student Educational Opportunity Office, Knight Hall room 330. Please contact your course instructor as soon as possible to complete plans for providing accommodations.