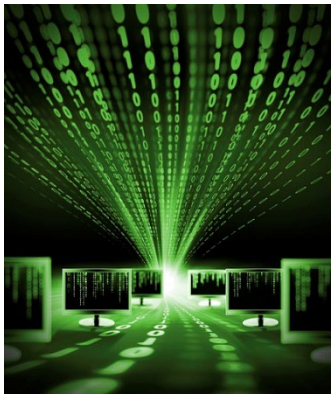
ASEN 4057 Aerospace Software

Material is preliminary and subject to change



Instructor: Alexandra Le Moine (Alexandra.LeMoine@colorado.edu)
Lecture: Monday/Wednesday 3:00-3:50pm, Aero N100 (COPILOT)
Lab: Monday/Wednesday 4:05-4:55pm, Aero N100 (COPILOT)
TA: Lynzee Hoegger

Canvas Webpage:
Github Private Repos:

Course Objectives

Aerospace engineers may go through their entire undergraduate education curriculum and have only a single formal course in *computing,* which often does not even cover formal programming, much less any details of the underlying processes by which the *computing* is accomplished. This is true despite an ever-increasing reliance on software by academia and industry for simulation and operational purposes. The purpose of this course is an attempt to fill that void.

Course Learning Goals

The goal of this course is to (1) provide aerospace engineers with an overview of key software and hardware computing concepts utilized in academia and industry and (2) give the background necessary to tackle programming projects confidently on different computing platforms with various software tools and programming languages. Students will: (A) gain deeper and broad technical computing experience including debugging, code management and optimization, documentations, and collaborative software development; (B) actively apply these technical skills to solving relevant aerospace engineering problems; and (C) develop the key skills and traits to be a good programmer and software developer in academia and industry.

Prerequisites

Students should have an extensive background in MATLAB programming and should understand programming fundamentals in C or C++. It is assumed that students in the course have taken ASEN 1320 and it is suggested that students also have taken CSCI 2270. Previous experience with C programming is recommended.

Course Textbook, References and Material

There is no required textbook for the course. Instead, reference material will be suggested throughout the course as applicable to the current lecture/assignment topics.

Course Topics

1. Fundamentals of Computer Programming and Software Design

2. MATLAB as a Tool for Software Design

a. Numerical Integration
b. Numerical Differentiation
c. Optimization
d. Debugging
e. Profiling

3. Introduction to Unix/Linux

a. Bash and Command Line Interface
b. Bash Programming

4. Version Control and Git

5. C as a Tool for Software Design
a. Review of Basic Concepts
b. Pointers and Memory Management
c. Compilation and Linking
d. Building Programs with Make
e. Defensive Programming and Debugging
f. Performance and Profiling
g. Code Optimization
h. Scientific Libraries: BLAS and LAPACK
i. Calling C within MATLAB with MEX Files

6. Moving Beyond Sequential Computing
a. Introduction to Parallel Computing Architectures
b. Parallel Computing with C and MPI
c. Parallel Computing with C and OpenMP
d. Parallel Computing with MATLAB's Parallel Computing Toolbox

Course Overview

The course will begin with a cursory overview of computer programming and software design. The course will then proceed with an overview of MATLAB as a tool for software design, reviewing basic concepts as well as exploring advanced programming techniques including debugging, profiling, handle graphics, graphical user interfaces, and numerical integration and optimization. CU-Boulder has a full MATLAB site license for students (https://oit.colorado.edu/software-hardware/software-catalog/matlab).

The course will continue with a discussion of compiled languages (C, C++, FORTRAN) versus interpreted languages (MATLAB, Python) and basic sequential computer architectures and operating systems. This will set the stage for an introduction to Unix/Linux, including bash shell programming. Understanding Unix/Linux and its environment is one of the primary goals of the course. The bash shell is the main interface with that environment, providing sophisticated configuration and programming capabilities. Another main goal of this course is to show students the similarities between programming languages and to demonstrate how it is easy to work in any programming language with an understanding of basis programming constructs.

From this point of the semester forward, the development environment will be the Unix/Linux operating system. Students will use the CSCI OpenStack Cloud Platform to work on Unix/Linux assignments. Before moving forward to the C programming language, the concept of version control will be introduced. The open-source version control system Git will be introduced for this purpose. The course will then proceed with an overview of C as a tool for software design. The course will review basic concepts, syntax, and structure before proceeding forward and discussing advanced concepts such as building programs with Make, and defensive programming and debugging. Various software tools which can dramatically improve a programmer's efficiency as well as his or her understanding of the underlying code will be introduced. These tools include debuggers, profilers, and compiler components. Several approaches will be introduced to improve code performance including compiler and memory access optimization, and various software libraries will also be introduced to emphasize that many tools have already been exhaustively developed and should not be re-implemented by the programmer.

Finally, the course will move on to parallel computation. There is currently a massive paradigm shift away from a single hardware-processing element to parallel computational units. To take advantage of state-of-the-art computer hardware architectures, much of the burden is placed on the programmer. This course will not spend any significant detail on embedded software/programming, as that is the focus on ASEN 4519/5519 – Microavionics. Nor does this course provide any instruction on the programming of applications for the popular

tablet/smartphone genre - as these are not currently used for solving traditional aerospace computational problems. Although the concepts of this class could be applicable and useful for such related applications.

Course Format

The course will follow a blend of traditional lectures with lab/computing assignments. There will be two lectures and two lab periods per week. Each lab will contain an activity that is expected to be completed during the lab time. There will be 4 individual programming/homework assignments. A final project will replace a final exam, and students may elect to work in pairs of their own choosing for the final project. Student assessment will be based on lab activities, assignments, and the final project.

Course Grading

40% Lab Activities
40% Homework Assignments
20% Final Project

Grades will be posted to the course website on Canvas.

Assignments Policy

Assignments will be submitted via GitHub. Students should try to turn in assignments that are organized with a professional appearance. Proper documentation and commenting should be used to explain programming and software concepts employed. ***Late assignments will have a 10% deduction immediately, and an additional 4% deduction will be applied for each hour the assignment is late. No assignments will be accepted 24 hours after the original due date.*** Collaboration is permitted, but collusion or plagiarism is not. Students are encouraged to discuss basic concepts related to the programming assignments, but students are **NOT** free to copy another student's assignment (except, of course, if the other student is a group member). The use of Generative AI such as ChatGPT should not be used to generate assignments but may be used to help assist with understanding concepts. ***Students who are caught in violation of the honor code will receive an "F" for the course and reported to the Dean's office for further punitive action.***

Final Project

A final project will replace a final exam and will be assigned toward the end of the semester and due before the Final Week. The final project will consist of improving performance of an existing serial code and parallelization. Students may work in pairs of their own choosing for the final project.

CU BOULDER POLICIES

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the Honor Code may include but are not limited to: plagiarism (including use of paper writing services or technology [such as essay bots]), cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. Understanding the course's syllabus is a vital part of adhering to the Honor Code.

All incidents of academic misconduct will be reported to Student Conduct & Conflict Resolution: StudentConduct@colorado.edu. Students found responsible for violating the Honor Code will be assigned resolution outcomes from Student Conduct & Conflict Resolution and will be subject to academic sanctions from the faculty member. Visit Honor Code for more information on the academic integrity policy.

Accommodation for Disabilities, Temporary Medical Conditions, and Medical Isolation

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or DSinfo@colorado.edu for further assistance. If you have a temporary medical condition, see Temporary Medical Conditions on the Disability Services website.

If you have a temporary illness, injury or required medical isolation for which you require adjustment, you must alert your professor immediately via email per the communications policy listed in section III.

Accommodation for Religious Obligations

Campus policy requires faculty to provide reasonable accommodations for students who, because of religious obligations, have conflicts with scheduled exams, assignments, or required attendance. Please communicate the need for a religious accommodation in a timely manner. In this class, you must alert your professor immediately via email per the communications policy listed in section III. See the campus policy regarding religious observances for full details.

Preferred Student Names and Pronouns

CU Boulder recognizes that students' legal information does not always align with how they identify. If you wish to have your preferred name (rather than your legal name) and/or your preferred pronouns appear on your instructors' class rosters and in Canvas, visit the Registrar's website for instructions on how to change your personal information in university systems.

Classroom Behavior

Students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote, or online. Failure to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, marital status, political affiliation, or political philosophy.

Additional classroom behavior information

- Student Classroom and Course-Related Behavior Policy.
- Student Code of Conduct.
- Office of Institutional Equity and Compliance.
- Student Code of Conduct.
- Office of Institutional Equity and Compliance.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

CU Boulder is committed to fostering an inclusive and welcoming learning, working, and living environment. University policy prohibits protected-class discrimination and harassment, sexual misconduct (harassment, exploitation, and assault), intimate partner abuse (dating or domestic violence), stalking, and related retaliation by or against members of our community on- and off-campus. The Office of Institutional Equity and Compliance (OIEC) addresses these concerns, and individuals who have been subjected to misconduct can contact OIEC at 303-492-2127 or email OIEC@colorado.edu. Information about university policies, reporting options, and OIEC support resources including confidential services can be found on the OIEC website.

Please know that faculty and graduate instructors are required to inform OIEC when they are made aware of incidents related to these concerns regardless of when or where something occurred. This is to ensure the person impacted receives outreach from OIEC about resolution options and support resources. To learn more about reporting and support for a variety of concerns, visit the Don't Ignore It page.

Mental Health and Wellness

The University of Colorado Boulder is committed to the well-being of all students. If you are struggling with personal stressors, mental health or substance use concerns that are impacting academic or daily life, please contact Counseling and Psychiatric Services (CAPS) located in C4C or call (303) 492-2277, 24/7.

Free and unlimited telehealth is also available through Academic Live Care. The Academic Live Care site also provides information about additional wellness services on campus that are available to students.