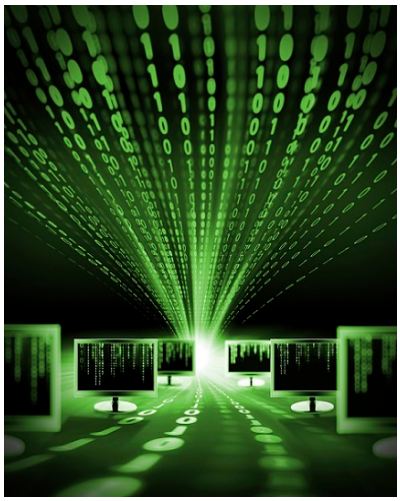
ASEN 4057 Aerospace Software

Material is preliminary and subject to change

Instructor: Prof. Tomoko Matsuo (Tomoko.Matsuo@colorado.edu)
Lecture: Monday/Wednesday 9:00-9:50am, ECCE 141 (CadLab)
Lab: Monday/Wednesday 10:00-10:50am, ECCE 141 (CadLab)
Office Hour: Monday 11:00am-11:50am, ECOT 614
Thursday 3:00pm-4:50pm, ECOT 614

Course Assistant: Madison Junker, Michael Labarge, Branden Solt
Office Hour: TBD, ECAE 124

Canvas Webpage: https://canvas.colorado.edu/
Github Private Repos: https://github.com/Aerospace-Software



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. It is assumed that students in the course have taken GEEN 1300, COEN 1300, ECEN 1030, ECEN 1310, CSCI 1300, CSCI 1310, or CSCI 1320. Previous experience with C programming is highly 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. ***Students will be required to have a 32 GB USB drive for use exclusively for this course.*** The drive is to store and transfer virtual machine files between your personal machine and a CadLab machine.

Course Topics

1. Fundamentals of Computer Programming and Software Design

2. MATLAB as a Tool for Software Design

- a. Review of Basic Concepts
  - i. Classes, Arrays, Computations
  - ii. If Statements
  - iii. While and For Loops
  - iv. Plotting
  - v. Scripts and Functions
- b. Advanced MATALB Concepts
  - i. Debugging
  - ii. Profiling
  - iii. Numerical Integration and Optimization
  - iv. Handle Graphics
  - v. GUI Development

3. Moving Beyond MATLAB

- a. Compiled Languages versus Interpreted Languages
- b. Basic Sequential Computer Architectures and Operating Systems

4. Introduction to Unix/Linux

- a. Bash and Command Line Interface
- b. Bash Programming

5. Version Control and Git

6. 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

7. 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-downloads-and-licensing/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 fairly 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 Linux operating system. A course installation will be available to be installed on your own computers via the virtualization product VirtualBox (https://www.virtualbox.org). 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. A number of 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. In order 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/smartphones genre - as these are not currently used for solving traditional aerospace computational problems. Although the concepts of this class could definitely 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. There will be six to eight individual quizzes throughout the semester, based on material covered in the lectures. There will be also six to eight programming assignments throughout the semester. For some of these programming assignments, you will need to work in groups, and for other assignments, you will need to work individually. For the group assignments, groups will be *randomly* determined. A midterm exam will be given in approximately the 12th week. There will be both in-class and take-home portions associated with this midterm exam. The in-class portion will test concepts, while the take-home portion will require programming. 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 individual quizzes, group assignments, the individual midterm exam, and the group final project.

Course Grading

30% Assignments
30% Quizzes
20% Midterm
20% Final Project

Grades will be posted to the course website on Canvas. Each assignment/quiz/exam will be based on a 100-point scale and weighted accordingly.

Assignments Policy

Assignments will initially be uploaded to the Canvas course website. After version control is introduced, students will employ Git to turn in assignments. Students should make an effort 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 5% deduction will be applied for each hour the assignment is late. No assignments will be accepted 18 hours after the original due date.*** Students will work individually or in *randomly* assigned groups for each programming assignment. Collaboration is permitted, even between groups. Students may 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). ***Students who are caught copying (or providing his or her assignment to another) will receive an "F" for the course and reported to the Dean's office for further punitive action.***

Quizzes Policy

Quizzes will take place at the beginning of class, and no make-up quizzes will be allowed. The lowest quiz grade will be dropped in computing the average to account for any missed quizzes. Quizzes will be announced approximately one week in advance. Quizzes will be closed book, and collaboration on the quizzes will not be tolerated. ***Students who are caught in these activities will receive an "F" for the course and reported to the Dean's office for further punitive action***

Examination Policy

The midterm examination will cover all material in the course including lecture, discussions, assignments, and quizzes. There will be two portions to the midterm exam, an in-class portion and a take-home portion. The in-class portion will cover concepts and be closed book. It will be very similar in style to the quizzes, and collaboration will not be tolerated. The take-home portion will require programming and be open book. Students may use class notes, homework solutions, and any additional material included on the course website for this portion of the exam. Students may not, however, consult with classmates or reference code from external sources. Submissions will be compared with submissions from other classmates as well as an online repository to ensure this policy is followed. ***Students who are caught consulting with classmates or referencing external sources 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.

Disabilities

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to me 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 or injury, see Temporary Medical Conditions under the Students tab on the Disability Services website.

Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail 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, political affiliation or political philosophy. Class rosters are provided to the instructor with the student's legal name. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records. For more information, see the policies on classroom behavior and the Student Code of Conduct.

Academic 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 policy may include: plagiarism, 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. All incidents of academic misconduct will be reported to the Honor Code (honor@colorado.edu); 303-492-5550). Students who are found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code as well as academic sanctions from the faculty member. Additional information regarding the Honor Code academic integrity policy can be found at the Honor Code Office website. Violations of the Honor Code include any act of **academic dishonesty** which is defined as follows. Any act in which a student gains, or attempts to gain, an unfair academic advantage over other students. These acts may include, but are not limited to:

I. **Plagiarism:** Portrayal of another's work or ideas as one's own.
II. **Cheating:** Using prohibited notes or study aids, allowing another party to do one's work/exam and turning in that work/exam as one's own, copying another student's course work, and collaborating on course work when prohibited.
III. **Fabrication:** Falsification or creation of data, research, or resources, altering a graded work without the prior consent of the course instructor.
IV. **Lying:** Deliberate falsification with the intent to deceive in written or verbal form as applied to an academic submission.
V. **Bribery:** Providing, offering, or taking rewards in exchange for a grade, or, an assignment, or in the aid of Academic Dishonesty.
VI. **Threat:** An attempt to intimidate a student, staff, or faculty member for the purpose of receiving an unearned grade or in an effort to prevent the reporting of an Honor Code violation, or in connection with any other form of Academic Dishonesty.
VII. **Unauthorized Access:** Gaining unauthorized access to protected academic information including, but not limited to: CU-SIS; a faculty member's computer, files, and/or office; or secure information on an online server.
VIII. **Clicker Fraud:** Using, or having someone else use, clicker technology fraudulently in an effort to receive academic credit.
IX. **Resubmission:** Submitting the same or similar work for credit more than once without permission from all course instructors involved.
X. **Aiding Academic Dishonesty:** Intentionally facilitating any act which may help a student to gain an unfair academic advantage including, but not limited to, any of the aforementioned acts.

Any act of academic dishonesty will result in an F for this course and will become a permanent part of the student's academic record.

Plagiarism

This course includes a research project and final written report. In constructing the research paper it is expected that ideas and concepts will come from specific reference material. It must be demonstrated that this material supports the original premise of your research project and is properly referenced. Please examine the following guidelines to avoid committing plagiarism:

How to avoid Plagiarism, Northwestern University
Plagiarism: What it is and how to recognize and avoid it, Indiana University

Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to fostering a positive and welcoming learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct (including sexual assault, exploitation, harassment, dating or domestic violence, and stalking), discrimination, and harassment by members of our community. Individuals who believe they have been subject to misconduct or retaliatory actions for reporting a concern should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127 or cureport@colorado.edu. Information about the OIEC, university policies, anonymous reporting, and the campus resources can be found on the OIEC website. Please know that faculty and instructors have a responsibility to inform OIEC when made aware of incidents of sexual misconduct, discrimination, harassment and/or related retaliation, to ensure that individuals impacted receive information about options for reporting and support resources.

Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to reasonably and fairly deal with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, all dates for exams, assignments and presentations are listed in the course schedule. Please review the course schedule and let me know if certain dates conflict with your religious obligations. See the campus policy regarding religious observances for full details.

Other Policies

Please be respectful of others during class time. This includes turning off your cell phone before class and not talking during class unless you have the floor. Details about all of the university policies can be found on the web at http://www.colorado.edu/policies/index.htm