# 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/Wednesday 11:00am-noon, ECOT 614

**Teaching Assistant:** Matthew Tooth (Matthew.Tooth@colorado.edu)
**Office Hour:** Tuesday 2:00-3:00pm & Thursday 10:00-11:00am, ECAE 124
**Course Assistant:** Benjamin Hutchinson

**Webpage:** Desire2Learn (https://learn.colorado.edu/)

## 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. Namely, the goal of this course is to provide aerospace engineers with an overview of key software and hardware computing concepts utilized in industry and give the background necessary to tackle programming projects confidently on different computing platforms with various software tools and programming languages.

## 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 for use exclusively for this course.***

## 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

## Class 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 of ever increasing clock frequencies to parallel computational units. This is true in everything from Field Programmable Gate Arrays (FPGAs), Digital Signal Processors (DSPs), Graphics Processing Units (GPUs), to generic microprocessors such as the Intel and AMD processor lines. In order to leverage these evolving multiple hardware units effectively, 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 Desire2Learn. Each assignment/quiz/exam will be based on a 100-point scale and weighted accordingly.

## Assignments Policy

Assignments will initially be uploaded to the 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 employed to explain programming and software concepts employed. Neatness, clarity, and completeness count. ***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. 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 a letter to me from Disability Services in a timely manner so that your needs may be addressed. Disability Services determines accommodations based on documented disabilities. Contact: 303-492-8671, Willard 322 or http://www.Colorado.EDU/disabilityservices

## Religious Observances

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 fixed in the course schedule. Please review the course schedule and let me know if certain dates conflict with your religious obligations. See policy details at http://www.colorado.edu/policies/fac_relig.html

## Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Students who fail to adhere to behavioral standards may be subject to discipline. Faculty have the professional responsibility to treat students with understanding, dignity and respect, to guide classroom discussion and to set reasonable limits on the manner in which students express opinions. See policies at https://www.colorado.edu/policies/student-classroom-and-course-related-behavior and at https://www.colorado.edu/osccr/honor-code

## Academic Honor Code

As a student at the University of Colorado you are bound by an academic code of honor. The purpose of an Honor Code at the University of Colorado at Boulder is to secure an environment where academic integrity, and the resulting behavior, can flourish. The Honor Code recognizes the importance of honesty, trust, fairness, respect, and responsibility and wishes these principles to be a defining part of the CU-Boulder campus. The Honor Code allows all students to have responsibility for, and the ability to attain, appropriate recognition for their academic and personal achievements. A student-run Honor Code is necessary because research indicates that these institutions are highly successful in alleviating indiscretions and promoting an academically honorable

community. In addressing any proven student violations regarding the Honor Code, the student leadership of the Honor Code Council applies only non-academic sanctions, and the faculty applies academic sanctions.

- Academic Dishonesty: Any of the following acts, when committed by a student at the University of Colorado at Boulder, shall constitute academic dishonesty:
  i. Plagiarism: Portrayal of another's work or ideas as one's own;
  ii. Cheating: Using unauthorized notes or study aids, allowing another party to do one's work/exam and turning in that work/exam as one's own; submitting the same or similar work in more than one course without permission from the course instructors;
  iii. Fabrication: Falsification or creation of data, research or resources, or altering a graded work without the prior consent of the course instructor;
  iv. Aid of Academic Dishonesty: Intentionally facilitating plagiarism, cheating, or fabrication;
  v. Lying: Deliberate falsification with the intent to deceive in written or verbal form as it applies to an academic submission;
  vi. Bribery: Providing, offering, or taking rewards in exchange for a grade, an assignment, or the aid of academic dishonesty;
  vii. 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.

Violations of the Honor Code are acts of academic dishonesty and include but are not limited to: plagiarism, cheating, fabrication, aid of academic dishonesty, lying to course instructors, lying to representatives of the Honor Code, bribery or threats pertaining to academic matters, or an attempt to do any of the aforementioned violations. All incidents of academic misconduct shall be reported to the Honor Code Council (honor@colorado.edu; 303-725-2273). ***Students who are found to be in violation of the academic integrity policy will be subject to both academic sanctions from the faculty member and non-academic sanctions (including but not limited to university probation, suspension, or expulsion). 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.*** For more information about the University of Colorado student honor code see https://www.colorado.edu/osccr/honor-code and at https://www.colorado.edu/policies/student-honor-code-policy

## Discrimination and Harassment

The University of Colorado at Boulder policy on Discrimination and Harassment, detailed at http://www.colorado.edu/policies/discrimination-and-harassment-policy-and-procedures, the University of Colorado policy on Sexual Harassment and the University of Colorado policy on Amorous Relationships apply to all students, staff and faculty. Any student, staff or faculty member who believes s/he has been the subject of discrimination or harassment based upon race, color, national origin, sex, age, disability, religion, sexual orientation, or veteran status should contact the Office of Discrimination and Harassment (ODH) at 303-492-2127 or the Office of Judicial Affairs at 303-492-5550. Information about the ODH and the campus resources available to assist individuals regarding discrimination or harassment can be obtained at http://hr.colorado.edu/dh/Pages/default.aspx

## 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