Course objectives
This course provides an introduction to programming in C++ and the software package Matlab.

Instructor
Johan Larsson (jola@umd.edu) Office hours: Mon 1500-1630 in EGR3149

Teaching Assistants and Fellows (TAs and TFs)
David Alcantara (david.a.rmb@gmail.com) Office hours: Mon 1400-1500
Jon Tedeschi (jtedesch@gmail.com) Office hours: Mon 1700-1800
Akash Chawla (achawla5@terpmail.umd.edu) Office hours: Tue 1100-1200
Jenna Marcus (jlmarcus@umd.edu) Office hours: Tue 1200-1300
Evan Winter (evanjwinter97@gmail.com) Office hours: Tue 1500-1600
Siavash Toosi (stoosi@umd.edu) Office hours: Tue 1700-1800

All office hours by TAs/TFs are held in EGR3109.

Lectures/Studios
There will be two traditional lectures per week (Mon/Wed 1100-1150, in JMP3201) during which the main concepts will be discussed together with sample code that illustrates these concepts.

The studios will be focused on programming exercises for the students. A set of exercises/problems will be given for each studio, which the students must work through during the studio. The time, place, and responsible TAs/TFs for the studios are:
- Section 0101 Fri 0800-0950 KEB2111 Toosi, Alcantara, Winter/Chawla
- Section 0102 Thu 1500-1650 KEB2111 Toosi, Marcus, Chawla
- Section 0103 Thu 1500-1650 EGR0312 Tedeschi, Winter

Textbook
The primary source of reference materials for the course will be the lecture notes. All homework and exam questions will be drawn from these notes; no other reference is required. Annotated example code developed and discussed during each lecture will be posted on ELMS for subsequent review and experimentation. Additionally, Matlab itself has an extensive built-in tutorial/help system, and there is a vast amount of information freely available on the web covering C++. Additional pointers to suggested websites for supplemental C++ information will be given when we reach that part of the course.

So you do not need a regular textbook – but if you really want one, two suggestions are:

- Etter and Ingber, “Engineering Problem Solving with C++”, Pearson Education Inc.

Course management
Assignments, notifications and supplementary material will be posted on the ELMS site. Please check this site regularly for updates. Note that the numerical grade computed by ELMS is not representative of your actual grade – it does not assign the correct weights to different components.
Assessment

The course grade will be calculated approximately as:

- Exams 65%
- Homework 25%
- Studios 10%

Studios

Every week you will practice the techniques shown in lectures on example problems during your assigned studio sections. The studios will be run as a kind of “super office hours”. TAs will circulate through the room as you work on the problems, answering your questions and checking your results.

**Attendance in studio is mandatory.** Studios are graded on attendance and participation. Lack of attendance, or minimal to no participation in the studio (measured in terms of attempted/completed problems), will earn a grade of 0 for that studio. Full marks are obtained for successfully completing all of the problems, or for showing honest, even if ultimately unsuccessful, attempts at all problems. In the event that you finish all problems before the end of your studio session, you may either leave early or else use the time to experiment further with the example code developed in lectures (recommended).

The problems for each week’s studios will be posted on ELMS prior to the first (Thursday) studios. While students with later (Friday) studios may be tempted to treat these as homework problems, completing them before their actual studio time, this is actively discouraged. Studios are meant to be guided, allowing you to ask questions as you get familiar with the material. The goal is not to finish the studio, the goal is to understand the techniques being used.

Reference solutions for the studio exercises will be available on ELMS shortly after the last studio session of the week (i.e. Fri evening or Sat morning).

Homework

After the last weekly studio session, the homework problem for that week will be released on ELMS. This will be a short program which you must independently complete. Your solution code must be uploaded to ELMS (not e-mailed to any course staff!) for grading before class starts on the following Wed.

Reference solutions for each week’s homework will be posted shortly after the due date (i.e. Wed afternoon or evening).

Exams

All exams are of the pen-and-paper kind, i.e., without a computer. The only aids allowed are pencil and eraser – no notes, no book, no calculator, etc.

Midterm exams will typically consist of 4-5 problems, while the final will have 8-9 problems. There will be a mix of “write code to do this” and “what does this code do?” problems.

Don’t worry (too much) about neatness – we understand that you may be rushing during exams. Having said this, you can’t provide multiple answers to a question – so if you start writing something and then change your mind, make sure you cross that out clearly and unambiguously.

Honor policy

All materials (homework and studio solutions) submitted for evaluation must be original work completed by you alone. You may discuss general approaches to the problems with other students (as well as the staff), but you may not under any circumstances share computer code with your classmates, or submit code written, in whole or in part, by someone else. You may not “jointly” write code with another student, and submit the result as your own solution. You may not download or copy, then submit as your own solutions, code from any other source (e.g., a book, web site, email,
text message, etc). The sole exception to this latter, when applicable, is code provided as part of the assignment itself; this is permissible to include as part of your solution.

**Policy on missing midterm exams**

Allowable reasons for missing the midterm exams include sickness (doctor’s note required), University-related travel (provided it is of sufficient importance) and religious observances (following the University policy, *students should notify the instructor within the first 2 weeks of the semester* to allow for adjustments to the scheduling of the midterm exams).

In case of a missed exam, the instructor will either: (a) provide a make-up exam; or (b) adjust the final grade calculation to effectively ignore the missed exam.

The University-wide policies can be found at http://www.ugst.umd.edu/courserelatedpolicies.html.

**Course software**

This course will require use of both Matlab as well as a C++ compiler. Both are available in the engineering computing clusters, where the labs will be held. You can always use the computers in these clusters to complete your homework, or you can install both pieces of software on your personal computer.

1. Accessing Matlab:

   Configure your computer to use Matlab through the VCL ("virtual computer lab") following the instructions at http://eit.umd.edu/vcl.

   Alternately, download and install Matlab (free student version) directly to your machine following the directions at https://terpware.umd.edu/Mac. (Follow link to your machine type–Mac or PC–look under “Analysis and Modeling”, then “Matlab for Students.” Login with your usual UMD username/password and follow instructions from there.)

2. Accessing C++:

   We will begin using C++ around the time of the first midterm exam, and additional information about accessing and using a C++ compiler will be given at that time.
Topics and tentative weekly schedule

**Unit I: Matlab**

1. Matlab arithmetic; variables; native functions; scripts (“m-files”)
2. Arrays; complex numbers; polynomials; plotting
3. Matrix and vector operations
4. User-defined functions; functions vs scripts
5. Logical expressions; conditionals and branching
6. Loops (for and while)

**Unit II: C++**

7. Program structure; compilation and execution; screen output
8. Keyboard input; mathematical functions; variables and data types
9. Conditionals and loops
10. User-defined functions; argument passing conventions
11. Arrays; static and dynamic memory allocation; pointers
12. File input and output
13. Data structures; structure arrays
14. Function and operator overloading
15. Classes, objects, and member functions

**Exams**

1. Matlab midterm tentatively during class on Monday Oct 16.
2. C++ midterm tentatively during class on Monday Nov 20.
3. Final exam tentatively at 0800-1000 on Friday Dec 15.