MTRE 2610 – Engineering Algorithms and Visualization – Spring 2016

Instructor: Kevin McFall, PhD

Office Phone: 678-915-3004

Cell Phone: 610-573-6242

Office Address: Q 320

Office Hours: MTWRF 2:00-3:00 pm or by appointment

E-mail: kmcfall@kennesaw.edu

Location: Lecture Q 310 / Laboratory Q 245

Meeting times: Lecture 10:00-10:50 am MW, Laboratory 11:00-1:50 pm M or W

Start Date: 01/11/2016

Pre-requisites: MATH 1190 (Calculus I) and CSE 1311 or equivalent (procedural programming)

Textbooks: How to Think Like a Computer Scientist, C++ version[1] by Allen B. Downey, C++ Annotations[2] by Frank B. Brokken, and An Introduction to MATLAB[3] by Krister Ahlersten all available for free online. Even more important than the textbooks are the required software MATLAB, available free of charge for KSU students[4], and the free Visual Studio Community[5].

Course Catalog Description: This course covers the development of algorithms to solve mechatronics related problems, using the tools C++, MATLAB, and Simulink. Object-oriented programming will be covered including classes, inheritance, and operator overloading. Basic numerical methods topics include matrix operations, solving linear systems, and curve fitting. Visualizing data in two and three dimensions with parametric curve plots, histograms, surface plots, and contour plots will be introduced. The laboratory component will focus on assignments relevant to mechatronics including robotics, controls, sensors, pneumatics, etc.

Learning Outcomes:

- Increase proficiency with programming skills
- Understand functional and data encapsulation using object oriented programming
- Interface members functions from with third-party software libraries
- Apply numerical methods to solve mechatronics-related problems
- Visualize and manipulate data in two and three dimensions
- Use sensor readings to control actuation

Topics Covered Include:

- Basics of object oriented programming including classes, operator overloading, and inheritance.
- Basic numerical methods such as curve fitting, root finding, solving linear systems, and numerical differentiation/integration.
- Simple sensors such as switches, potentiometers, and infrared range finders
- Basic actuators such as DC motors, servos, and pneumatic cylinders

[1] http://greenteapress.com/thinkcpp/
[2] http://cppannotations.sourceforge.net/annotations/html/
[3] http://bookboon.com/en/an-introduction-to-matlab-ebook
[4] https://apps.kennesaw.edu/files/pr_app_uni_cdoc/doc/Matlab_DownloadInstructions.pdf
[5] https://www.visualstudio.com/downloads/download-visual-studio-vs

Grading Policy

Homework (15%): Several programming assignments will be collected which can be completed individually or in groups of two. Source code will be submitted in a D2L Dropbox with filenames beginning with “HW” followed by 2 digits for the homework number, followed by the last name(s) of the student(s). Homework written in MATLAB will be submitted as a single .m file with the filename as described. C++ homework is submitted instead as a .zip file with the required file name. The .zip file will contain a folder for each problem containing the Visual Studio project. Each folder will contain all the required files and subfolders so that the project can be opened and the code executed. Only a single submission for groups of two is required. When executed, the program will output to the screen the problem numbers and answers. Problems with graphical output will display which figure number contains the solution.

- Submission requirements satisfied (1 point)
- Significant progress made toward solution (2 points)
- Correct result obtained (1 point)

Laboratory exercises (25%): Laboratory exercises will generally be graded during the scheduled time and groups not completing in time will be assessed on what is finished. Longer laboratories may be allowed to submit as an informal lab report. These reports *do not* require an introduction, procedure, etc. but rather simply document the source code developed, any relevant output obtained, and answers to any questions posed in the lab description. Submissions will be submitted before the beginning of the next lab meeting in D2L dropboxes in the same manner as homework, but where “HW” is replaced with “Lab” in the filename. Grades will be assigned according to the fraction of the lab content correctly addressed in the report.

Tests (3x15%): Three in-class tests will be administered where students have access to internet-connected computers to compile and execute programs. Test grades may be curved in an attempt to maintain the class average in the C range. No assistance from any human may be solicited during the test period. Source code solving each problem will be uploaded to a D2L Dropbox before the end of the test in the same manner as homework but with “HW” replaced with “Test”. Problems will be graded according to the following rubric:

0	No content relevant to solving the problem
30 (F)	Some relevant content but no indication of how to solve the problem
60 (D)	Some indication that the correct solution method is being followed
70 (C)	Significant work showing understanding of how to approach the problem
80 (B)	Primary details of solving the problem are complete but significant mistakes are made
90 (A)	Problem is answered correctly except for minor mistakes such as sign or algebra errors
100	Problem is answered completely with the correct answer

Final exam (15%): A cumulative final exam will follow the same format as the tests with “Final” in the submitted filename.

The scale for the final course grade is as follows:

A	90-100
B	80-89
C	70-79
D	60-69
F	0-59

Attendance Policy

Forcing everyone to come to every class is not practical. Each student bears responsibility for material covered in class. If students choose to miss class, that is their decision. Class time will be spent explaining the day's content and working problems, under the assumption that all students have read and understood the reading assignment. In general, late assignments are not accepted nor can make-up tests be administered. Laboratory attendance is mandatory; students will receive a grade of zero for missed lab sessions. Extenuating circumstances can result in exceptions to these rules, but agreement must be reached with the instructor in advance of the assignment, test, or lab that will be missed.

Academic Misconduct

The Code of Academic Integrity at KSU states that

> *As a member of the Kennesaw State University community of scholars, I understand that my actions are not only a reflection on myself, but also a reflection on the University and the larger body of scholars of which it is a part. Acting unethically, no matter how minor the offense, will be detrimental to my academic progress and self-image. It will also adversely affect all students, faculty, staff, the reputation of this University, and the value of the degrees it awards. Whether on campus or online, I understand that it is not only my personal responsibility, but also a duty to the entire KSU community that I act in a manner consistent with the highest level of academic integrity. Therefore, I promise that as a member of the Kennesaw State* University community, I will not participate in any form of academic misconduct.

All acts of academic misconduct will be documented with the Student Academic Misconduct Incident form and included on the student's academic record.

Disability Statement

Any student who, because of a disabling condition, may require some special arrangements in order to meet the course requirements should contact the instructor as soon as possible to arrange the necessary accommodations. Students should present appropriate verification from KSU Student Disability Services. No requirement exists that accommodations be made prior to completion of this approved University process.

Communication

Course material will be disseminated in D2L including lecture notes, homework solutions, old tests, etc. All official course announcements, including instructions when class may be cancelled, will be posted in the D2L course news. Be sure to check D2L regularly.

Course Schedule

Day	Date	Description	Reading	HW due	Lab exercise
Mon	01/11	MATLAB environment	1, 2, 3[6], 8, 10, 13		C++ review
Wed	01/13	Vector and matrix indexing	4		
Wed	01/20	2D plotting and searching	7[6]	HW01	C++/MATLAB comparison
Mon	01/25	User defined functions	9[6]		Arduino basics
Wed	01/27	Debugging	12[6]	HW02	
Mon	02/01	Surface and contour plotting			H-bridge motor drivers
Wed	02/03	Recitation		HW03	
Mon	02/08	Structures and cell arrays			Controlling motor motion
Wed	02/10	File I/O	6[6]		
Mon	02/15	Review		HW04	Pneumatics and active light sensors
Wed	02/17	Test 01			
Mon	02/22	Curve fitting	5.5, 11.4[6]		Image processing
Wed	02/24	Root finding and equation solving	11.1-11.2[6]		
Mon	02/29	Matrix operations	5.1-5.4[6]	HW05	Machine vision
Wed	03/02	Numerical differentiation/integration	11.3[6]		
Mon	03/07	Recitation			Distance sensor
Wed	03/09	Review		HW06	
Mon	03/14	Test 02			PI control
Wed	03/16	Dynamic array allocation			
Mon	03/21	Structure data and functions	8, 11[7], 3.2[8]		Acceleration
Wed	03/23	Classes	14[7], 7,9[8]		
Mon	03/28	Encapsulation and libraries		HW07	Serial communication
Wed	03/30	Recitation			
Mon	04/11	Function/operator overloading	11[8]	HW08	Complex class
Wed	04/13	Inheritance	13[8]		
Mon	04/18	Constructors/destructors		HW09	Mandelbrot
Wed	04/20	Recitation			
Mon	04/25	Review		HW10	Mandelbrot con’t
Wed	04/27	Test 03			
Mon	05/02	Review			

[6] *An Introduction to MATLAB*

[7] *How to Think Like a Computer Scientist*

[8] *C++ Annotations Version 10.2.0*