EECE.2160: ECE Application Programming

Fall 2018

Syllabus

Course Meetings

Section 201: MWF 8-8:50, Ball 326
Section 202: MWF 1-1:50, Ball 208
Section 203: MWF 12-12:50, Ball 208

Course Website

Main page: http://mjgeiger.github.io/eece2160/f18/

Schedule: http://mjgeiger.github.io/eece2160/f18/schedule.htm

All course announcements will be posted on the course Blackboard page. You are responsible for checking that site, as well as the sites listed above, on a regular basis.

Instructor

Dr. Michael Geiger
E-mail: Michael_Geiger@uml.edu
Office: Ball 301A
Phone: 978-934-3618 (x43618 on campus)
Office hours: Monday 9-10:30 AM, Wednesday 9-10:30 AM, Thursday 1:30-3 PM

During the above hours, student questions are my top priority. I am available by appointment at other times. Feel free to stop by my office, e-mail me questions, or schedule a one-on-one appointment. Office hours are subject to change.

Teaching Assistant: TBD *(TA contact info and office hours to be posted ASAP)*

Required Textbook

Programming in C with zyLabs, EECE.2160, Fall 2018

The text is required because (a) part of your grade depends on completing its interactive examples, and (b) you will submit all programming assignments through the textbook IDE.

To obtain access to the textbook, please follow the instructions below:

1. **Sign in or create account at learn.zybooks.com**
2. **Enter zyBook code: UMLEECE2160GeigerFall2018**
3. **Subscribe ($77 for this term; lasts until 1/4/19)**
   - You must use a *student.uml.edu* e-mail address to subscribe and select the course section (201, 202, 203) for which you are registered

Previous Textbook *(recommended for anyone who prefers a traditional, hardcopy textbook)*

K.N. King, *C Programming: A Modern Approach*, 2nd edition, 2008, W.W. Norton.
ISBN: 978-0-393-97950-3

Course Overview

Catalog Description: Introduces C programming for engineers. Covers fundamentals of procedural programming with applications in electrical and computer engineering and embedded systems. Topics include variables, expressions and statements, console input/output, modularization and functions, arrays, pointers and strings, algorithms, structures, and file input/output. Introduces working with C at the bit manipulation level. Laboratories include designing and programming engineering applications. 3 credits.

Course Objectives: By the end of this course, you should understand how to use the following:

1. **Basic C Language Concepts:** constants, variables, operators, expressions and assignment statements
2. **Input and Output:** Reading data from the keyboard and displaying formatted results on the screen
3. **Flow of Control 1 – Decisions and selection:** `if` and `switch` statements
4. **Flow of Control 2 – Repetition:** `while`, `do-while`, and `for` loops
5. **Functions**: Defining and calling functions. Using arguments to pass data to a function. Using arguments to obtain results from a function. Return values.
6. **Data Structures 1**: One and two-dimensional arrays. Character strings.
7. **Data Structures 2**: Structures, collections of data components of differing types.
8. **File Input / Output**: Writing programs which obtain input from a file rather than the keyboard, and which write results to a file rather than to the screen

Grading: Grades will be computed on an A to F scale; no A+ grades will be assigned, in accordance with UMass Lowell policy. The weights assigned to the various items are:

Programming assignments	50%	Lowest Exam 1/Exam 2 grade	10%
Textbook participation activities	5%	Highest Exam 1/Exam 2 grade	15%
Textbook challenge activities	5%	Exam 3	15%

Incomplete grades will only be given in exceptional situations, and the student must be passing the class at the time the grade is requested.

The following rubric describes how grades will be assigned if no grading curve is applied. A grading curve may be used at the instructor's discretion, depending on the overall course average at the end of the term. Grades will not be curved down, meaning that the table below describes the minimum letter grade you will earn for a final average in each of the ranges shown:

Range	Grade	Range	Grade
> 92	A	78-79	C+
90-92	A-	73-77	C
88-89	B+	70-72	C-
83-87	B	68-69	D+
80-82	B-	60-67	D
		< 60	F

Your grade is based strictly on the work you do during the semester. Please do not ask for extra credit work to improve your grade—any extra credit work I give is available to the whole class, not just the students who ask for it.

Textbook activities: Each lecture has a related reading assignment, which contains a set of participation and/or challenge activities. These activities must be completed no more than three days after the associated lecture, including weekends and holidays, to receive credit. (For example, activities related to the lecture on Friday, 9/7 must be completed by the end of the day (11:59:59 PM) Monday, 9/10.) Activities completed after the due date receive a grade of 0.

Programming assignments: Typically, you will have about one week to complete each assignment. All assignments will be graded according to the program grading guidelines, to be distributed separately. Late assignments will lose 2^{n-1} points per day, including weekends and holidays. You will submit your work directly through the class zyBook.

For each program, grades are broken into two categories—output (60%) and programming style (40%)—unless otherwise specified in the assignment. Output grades are immediately available after program submission—the zyBooks IDE auto-grades your output, based on a series of test cases given with each program, and displays your score. For programming style, an instructor will manually review your program and assign points according to a predetermined rubric.

For each program, you will be allowed one resubmission to improve your grade without penalty. You must resubmit your code by the given deadline for that assignment; late penalties apply to late resubmissions. Note that this policy does not allow you to avoid penalties when the original submission is late (e.g., an assignment losing 4 points for a late initial submission has a maximum possible score of 96 for the resubmission). See the grading guidelines for more details.

Exams: Make-up exams will only be offered in exceptional circumstances. You must notify your instructor as early as possible in order to determine an appropriate make-up date.

Class participation: You are responsible for all material discussed or announced in class. You are expected to attend class regularly and participate in any in-class discussions, as such exercises are essential to your learning. Although lecture attendance is not explicitly required, regular attendance will improve your understanding of the course concepts.

Academic Honesty

All assignments and exams must be completed individually unless otherwise specified. You may discuss concepts or material covered in class, but may not share any details of your solutions to assigned problems, including algorithms and code. Plagiarism (in this course, copying code from an outside source) is also unacceptable and will be treated as an instance of cheating.

Students are allowed to discuss assignments in general terms and to help one another fix specific errors, such as compiler errors or output formatting. In this case, students must note in their program header that they received assistance from a classmate. However, any code sharing—even if used only to help a classmate solve a specific error—is an academic honesty violation.

Any assignment or portion of an assignment violating this policy will receive a grade of 0 for all parties concerned. Depending on the severity of the infraction, or in cases of repeat violations, the instructor may give additional penalties, up to and including a failing grade in the course.

Further information on the University Academic Integrity policy can be found at:

https://www.uml.edu/Catalog/Undergraduate/Policies/Academic-Policies/Academic-Integrity.aspx

Course Schedule

This schedule contains a tentative schedule of topics we will cover throughout the term; the course website will contain the most up-to-date version. The web page will also describe which section(s) of the textbook are associated with each lecture, as well as the due date for each programming assignment. **You should expect to complete approximately 10 programming assignments this term.**

Please note that several days are denoted as "PE#"—in these classes, we will do an in-class programming exercise. While students will be able to participate even if they do not have a computer, I encourage anyone with a laptop to bring it to class on these days.

Please note that the exam dates are fixed—the first exam will be held on **Friday, October 5 in class**, the second exam will be held on **Monday, November 5 in class**, and the third exam will be held on **Monday, December 17, from 3-6 PM (room TBD)**.

Week	Date (M)	Lecture Topics	Programs
1	9/3	*No Monday lecture—classes start 9/5* 1. Course introduction/overview 2. Basic C program structure; IDE demo	Program 1 (due 9/12)
2	9/10	3. Data types; variables 4. Operators; output with printf() 5. Input with scanf()	Program 2 (due 9/21)
3	9/17	6. PE1 (Flowcharts, debugging) 7. If statements 8. Range checking	Program 3 (due 10/1)
4	9/24	9. Switch statements; while loops 10. Do-while loops; loop examples 11. PE2 (Conditionals, while loops)	
5	10/1	12. For loops 13. Exam 1 Preview **Friday, 10/5: EXAM 1**	Program 4 (due 10/12)
6	10/8	*No Monday lecture—Columbus Day* 14. Exam 1 Review 15. Functions *(Thursday, 10/11)* 16. Function examples	Program 5 (due 10/22)
7	10/15	17. Pointer arguments 18. Pointer argument examples 19. PE3 (Functions)	
8	10/22	20. One dimensional and two dimensional arrays 21. Arrays and functions 22. Character arrays and strings	Program 6 (due 10/31)
9	10/29	23. String functions and examples 24. Topics TBD 25. Exam 2 Preview	Program 7 (due 11/13)

Course Schedule (continued)

Week	Date (M)	Lecture Topics	Programs
10	11/5	**Monday, 11/5: EXAM 2** 26. Structures 27. Nested structures	
11	11/12	*No Monday lecture—Veterans Day* 28. PE4: Structures *Wednesday, 11/14: Last day to withdraw* 29. File I/O	Program 8 (due 11/26)
12	11/19	30. Character and line I/O *No Wednesday, Friday lecture—Thanksgiving*	
13	11/26	31. Bitwise operators 32. Common bitwise operations 33. Dynamic memory allocation	Program 9 (due 12/5)
14	12/3	34. Dynamic memory allocation (continued) 35-36. Topics TBD	Program 10 (due 12/13)
15	12/10	37. Topics TBD 38. Exam 3 Preview *Classes end Thursday, 12/13*	
	12/17	**EXAM 3: MONDAY, 12/17, 3-6 PM (room TBD)**	