EECE.2160: ECE Application Programming

Fall 2019

Syllabus

Course Meetings

Section 202: MWF 1-1:50, Kitson 305
Section 203: MWF 12-12:50, Kitson 305

Course Website

Main page: http://mjgeiger.github.io/eece2160/f19/

Schedule: http://mjgeiger.github.io/eece2160/f19/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.

Instructors

Dr. Lin Li *(Sec. 203)*
E-mail: Lin_Li@uml.edu
Office: Ball 310
Office hours: Thursday/Friday 1-3 PM

Dr. Michael Geiger *(Sec. 202)*
E-mail: Michael_Geiger@uml.edu
Office: Ball 301A
Phone: 978-934-3618 (x43618 on campus)
Office hours: Monday/Wednesday 2-3 PM; Tuesday 12:30-2 PM, or by appointment

During office hours, student questions are our top priority. Feel free to stop by the office, e-mail questions, or schedule a one-on-one appointment. Office hours are subject to change.

Required Textbook

Programming in C with zyLabs, EECE.2160, Fall 2019

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.

The course Blackboard site has a link through which you can—and should—purchase this text. Accessing the text through Blackboard ensures assignment grades will be passed back to the Blackboard Grade Center.

Previous Textbook ***(recommended for anyone who wants 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 activities	10%	Highest Exam 1/Exam 2 grade	15%
		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 instructors' 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 we 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. Each chapter's activities will be due shortly after the last lecture covering that material, with the due date to be posted both on the course schedule page and Blackboard. **Activities completed after the due date receive a grade of 0.**

Programming assignments: Typically, you will have about one week to complete each assignment. Late assignments will lose 2^{n-1} points per *n* days late, including weekends and holidays. (So, -1 for 1 day late, -2 for 2 days late, -4 for 3 days late, etc.) You will submit your code through the textbook, accessing each assignment using the appropriate Blackboard link. **The Blackboard link will show the due date by which LATE submissions are due, with a note explaining what the program due date is for on-time submissions.** You must also submit a brief Blackboard "assignment" that allows the grader to assign points for programming style.

For each program, you are allowed one resubmission to improve your grade without penalty. **You request a regrade by resubmitting the "style assessment" assignment for the given program**. Late penalties on the original submission still apply—for example, an assignment that is 3 days late has a maximum score of 96 for the resubmission.

Program grades are broken into two categories—output (60%) and programming style (40%)—unless otherwise specified. Output grades are available after submission—the zyBooks IDE auto-grades your output using test cases given with each program. For programming style, an instructor manually reviews your program and assigns points according to a given rubric.

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) will also be treated as an instance of cheating.

Students may discuss assignments in general terms and may 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 and the due dates for each assignment.

Please note that several days are denoted as "PE#"—these classes will contain 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.

The exam dates will be fixed shortly after the start of the semester. Tentative dates for the first two exams are shown below (during weeks 5 and 10), and the third exam will be held during final exams, at a date and time to be determined by the registrar's office.

Week	Date (M)	Lecture Topics
1	9/2	*No Monday lecture—Labor Day* 1. Course introduction/overview 2. Basic C program structure; IDE demo
2	9/9	3. Data types; variables *Tuesday, 9/10: last day to add without permission number* 4. Operators; output with printf() 5. Input with scanf()
3	9/16	6. PE1 (Flowcharts, debugging) *Tuesday, 9/17: last day to add/drop course* 7. If statements 8. Range checking
4	9/23	9. Switch statements; while loops 10. Do-while loops; loop examples 11. PE2 (Conditionals, while loops)
5	9/30	12. For loops *(Tuesday, 2/19)* 13. Exam 1 Preview ***Exam 1 likely late in week 5 or early in week 6***
6	10/7	14. Exam 1 Review -or- Functions 15. Functions -or- Exam 1 Review 16. Function examples
7	10/14	*No Monday lecture—Columbus Day* 17. Pointer arguments *(Tuesday, 10/15)* 18. Pointer argument examples 19. PE3 (Functions)
8	10/21	20. One dimensional and two dimensional arrays 21. Arrays and functions 22. Character arrays and strings
9	10/28	23. String functions and examples 24. Topics TBD 25. Exam 2 Preview

Course Schedule (continued)

Week	Date (M)	Lecture Topics
10	11/4	***Exam 2 likely early in Week 10*** 26. Structures 27. Nested structures
11	11/11	*No Monday lecture—Veterans Day* 28. PE4: Structures *Thursday, 11/14: Last day to withdraw* 29. File I/O
12	11/18	30. Character and line I/O 31. Bitwise operators 32. Common bitwise operations
13	11/25	33. Dynamic memory allocation *No Wednesday, Friday lectures—Thanksgiving Break*
14	12/2	34. Dynamic memory allocation (continued) 35-36. Topics TBD
15	12/9	37. Topics TBD 38. Exam 3 Preview *Classes end Thursday, 12/12*
	TBD	**EXAM 3: during finals; time/location TBD**