EECE.2160: ECE Application Programming

Spring 2020

Syllabus

Course Meetings

Section 201: MWF 8-8:50 AM, Kitson 305
Section 202: MWF 12-12:50 PM, Ball 314

Course Website

Main page: http://mjgeiger.github.io/eece2160/sp20/

Schedule: http://mjgeiger.github.io/eece2160/sp20/schedule.html

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. 201)*
E-mail: Lin_Li@uml.edu
Office: Ball 310
Office hours: M/W/F 10:30 AM-12 PM & 1-1:50 PM, or by appointment

Dr. Michael Geiger *(Sec. 202)*
E-mail: Michael_Geiger@uml.edu
Office: Ball 301A
Phone: 978-934-3618 (x43618 on campus)
Office hours: M/W 10-11:30 AM & 1-1:50 PM; Tu 12:30-2 PM, or by appointment

Regarding instructor contact:

- During office hours, student questions are our top priority. You do not need to schedule an appointment to attend office hours—just drop in!
- Any changes to office hours will be announced on Blackboard.
- We encourage you to e-mail us questions, but please allow up to 24 hours for a response.

The Centers for Learning and Academic Support Services (CLASS) provide tutoring services, including online searchable schedules that include resources on all campuses. A tutoring request form is available if no tutors are listed for your class.

Required Textbook

Programming in C with zyLabs, EECE.2160, Spring 2020

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.

Course 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 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%	Exam 2 grade	10%
Textbook activities	10%	Exam 3 grade	10%
Exam 1 grade	10%	Exam 4 grade	10%

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

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.

General Course Information

Textbook activities

- Textbook activities (participation/challenge activities) will be assigned one chapter at a time, even though each lecture typically covers less than one chapter worth of material.
- Each set of activities will be due on the day we first cover that material in class, with the due date to be posted both on the course schedule page and Blackboard.
- **Textbook activities completed after the due date receive a grade of 0.**

Programming assignments

- Late assignments 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 submit your code through the textbook site but must access each assignment using the appropriate Blackboard link.
  - **Due to differences in the two sites, Blackboard will show the due date for on-time submissions, while zyBooks shows the last day to submit late programs.**
- Program grades consist of two parts—output (60%) and programming style (40%)
  - Output grades are available after submission—the zyBooks IDE auto-grades your output using test cases given with each program.
    - After running your program through the test cases, you must submit the results to Blackboard.
  - For programming style, an instructor manually reviews your program and assigns points according to a given rubric.
    - Each style assessment requires a brief Blackboard "assignment" that allows the grader to assign points for programming style.
    - **Failure to submit the style assessment will result in a grade of 0 for that part of the assignment, even if you submit your code to zyBooks!**
- For each program, you are allowed one penalty-free resubmission to improve your grade.
  - To request a regrade:
    - Ask an instructor to "reopen" the given program for you on zyBooks
    - Resubmit the "style assessment" assignment for the given program
    - If your output score changed, make sure you submit the new score to Blackboard.
  - 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.

Exams

- The first three exams will be in-class, while the fourth will be during the finals period.
- 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.

Student Mental Health and Well-Being

We are a campus that cares about the mental health and well-being of all individuals in our campus community. Your personal health and well-being can impact your success in this course. Students sometimes experience mental health concerns or stressful experiences that interfere with academics and have a negative impact on everyday life.

If you or someone you know are experiencing mental health challenges at UMass Lowell, please contact Counseling (information below). Their services are free and confidential, and same day appointments are available.

I am available to talk with you about stresses related to your work in my class. Furthermore, I can assist you in reaching out to any one of a wide range of campus resources, including:

- Counseling Services provides crisis intervention, assessment, referrals, short-term individual counseling, group therapy, and on-call clinicians outside of business hours. They are located at University Crossing Suite 300 and their 24/7 phone number is 978-934-6800.
- UMatter2 is a university-wide initiative to support students and promote mental health. The office may be reached at 978-934-6671.

Consider also reaching out to a friend, faculty or family member you trust for help getting connected to the support that can help.

Disability Services

Disability Services assists students with documented disabilities by providing reasonable accommodations, coordinating services and creating accessibility through assistive technology and other means. They are located in The Wellness Center at University Crossing, Suite #300 and may be called directly at 978-934-4574.

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, at a minimum, receive a grade of 0 for all parties concerned, and violations will be reported to the Provost's Office. 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 dates for the three in-class exams are fixed: Wednesday, 2/12, Wednesday, 3/4, and Monday, 4/6. The fourth exam will be held during final exams, at a date, time, and location to be determined by the registrar's office. That exam will likely be a common exam for both sections.

Week	Date (M)	Lecture Topics
1	1/20	*No Monday lecture—Martin Luther King, Jr. Day* 1. Course introduction/overview 2. Basic C program structure; IDE demo
2	1/27	*Monday, 1/27: last day to add without permission number* 3. Data types; variables 4. Operators; output with printf() 5. More on printf(); scanf() details
3	2/3	*Monday, 2/3: last day to add/drop course* 6. Debugging basics; if statements 7. Else and else if blocks 8. Range checking; switch statements
4	2/10	9. Exam 1 Preview **Wednesday, 2/12: EXAM 1** 10. While and do-while loops
5	2/17	*No Monday lecture—Presidents' Day* 11. Loop examples *(Tuesday, 2/18)* 12. For loops 13. PE1 (Conditionals, loops)
6	2/24	14. Finish PE1 15. Functions 16. Function examples
7	3/2	17. Exam 2 Preview **Wednesday, 3/4: EXAM 2** 18. Topics TBD
8	3/9	*Spring Break—no classes*
9	3/16	19. Function review; pointer arguments 20. One-dimensional and two-dimensional arrays 21. Arrays and functions

Course Schedule (continued)

Week	Date (M)	Lecture Topics
10	3/23	22. Character arrays and strings 23. String functions 24. Structures
11	3/30	25. Nested structures 26. PE2: Structures 27. Exam 3 Preview
12	4/6	**Monday, 4/6: EXAM 3** *Tuesday, 4/7: Last day to withdraw* 28. File I/O 29. Character and line input
13	4/13	30. Bitwise operators 31. Common bitwise operations: set, clear, and invert bits 32. Common bitwise operations: bit masking
14	4/20	*No Monday lecture—Patriots Day* 33. Dynamic memory allocation 34. Dynamic memory allocation (continued)
15	4/27	35-36. Topics TBD 37. Exam 3 Preview *Classes end Friday, 5/1*
	TBD	**EXAM 4: during finals; time/location TBD**