

16.216: ECE Application Programming

Fall 2011

Syllabus

Course Meetings

Section 201: MWF 12:00-12:50 PM, Ball Hall 210

Section 202: MWF 1:00-1:50 PM, Ball Hall 313

Instructor

Dr. Michael Geiger

E-mail: Michael_Geiger@uml.edu

Office: Engineering Building 118A

Phone: 978-934-3618 (x3618 on campus)

Office hours: W 2-4, Th 11-1, F 2-3

The above hours are the minimum that I will be available—I will (usually) be on campus five days a week. Feel free to stop by my office (and knock if the door is closed!), e-mail me questions, or schedule a one-on-one appointment. These hours are subject to change.

Teaching Assistant

Sai Chalamalasetti

E-mail: Sairahul_Chalamalasetti@student.uml.edu

Office hours: M/T/Th 2-4, Ball Hall 304 (Virtual Instrumentation Lab)

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.

Credits: 3

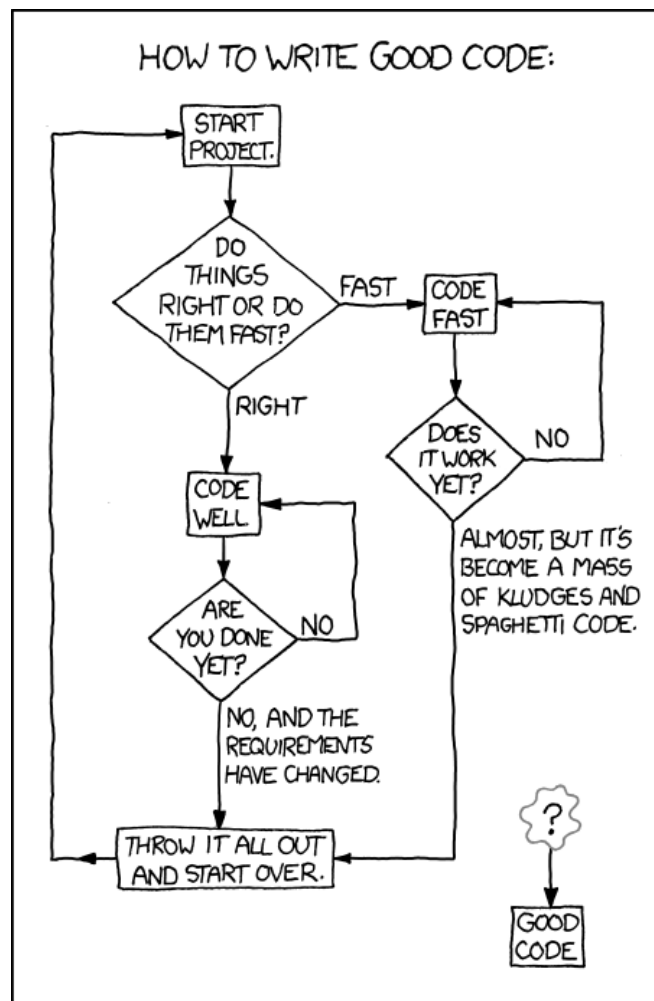
Prerequisites: 25.108 (Intro to Engineering II) and ECE major

Course Overview (cont.)

Course Objectives: By the end of this course, you should understand and be able to use all of 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

What you'll really learn in this course ... ? (<http://xkcd.com/844>)



Course Overview (cont.)

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:

Weekly programming assignments	40%
Project	20%
Exam 1	10%
Exam 2	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.

Programming assignments: Typically, you will have 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 via e-mail directly to Dr. Geiger.

For each assignment, you will be allowed one resubmission to improve your grade without penalty. However, you must resubmit your assignment within 24 hours of your grade being posted. Note that late penalties still apply to resubmitted assignments for which the original submission is late. See the grading guidelines for more details.

Exams: Make-up exams will only be offered in exceptional circumstances. You must notify Dr. Geiger 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 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—examples include compiler errors or output formatting. In this case, students are required to note that they received assistance from a classmate by listing that person's name and the nature of their assistance as part of their assignment header.

Any assignment or portion of an assignment that violates this policy will receive a grade of zero for all parties concerned. Depending on the severity of the infraction, or in cases of repeat violations, additional penalties may be given at the instructor's discretion, up to and including a failing grade in the course.

Further information on the UMass Lowell Academic Integrity Policy can be found at: http://www.uml.edu/catalog/undergraduate/policies/academic_dishonesty.htm

Course Schedule

This schedule contains a tentative schedule of topics we will cover throughout the term; this schedule is subject to change. Note that most lectures are associated with some section(s) of the textbook, which I suggest you read prior to class.

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 suggest anyone with a laptop bring it to class on these days.

Please note that the exam dates are fixed—the first exam will be held on **Wednesday, October 5 in class**, the second exam will be held on **Wednesday, November 9 in class**, and the third exam will be held during final exams (date TBD).

Week	Date (M)	Lecture Topics	Reading	Programs
1	9/2 (F)	Lec. 1: Course introduction/overview; C programming cycle	F: None	Program 1 (due 9/9)
2	9/5	<i>No Monday lecture—Labor Day</i> Lec. 2: Basic C program structure; representing data in C (number systems, data types, constants, variables) Lec. 3: Using data in C (operators, expressions, assignments)	W: 2.1-2.3, 2.8 F: 2.4, 2.6-2.7, 4.1-4.5, 7.1-7.3, 20.1	Program 2 (due 9/16)
3	9/12	Lec. 4: Numeric I/O (<code>printf()</code> and <code>scanf()</code>); output formatting intro Lec. 5: Output formatting (cont.) Lec. 6: PE1 (Expressions & I/O)	M: 2.4-2.5, 3.1-3.2 W: None F: None	Program 3 (due 9/23)
4	9/19	Lec. 7: Conditional statements: <code>if</code> Lec. 8: Conditional statements: <code>switch</code> Lec. 9: PE2 (Conditional statements)	M: 5.1-5.2 W: 5.3 F: None	Program 4 (due 9/30)
5	9/26	Lec. 10: Loops: <code>while/do-while</code> Lec. 11: Loops: <code>for</code> , <code>break/continue</code> Lec. 12: PE3 (Loops)	M: 6.1-6.2 W: 6.3-6.5 F: None	Program 5 (due 10/14)
6	10/3	Lec. 13: Exam 1 Preview Wednesday, 10/5: EXAM 1 Lec. 14: Exam 1 Review	None	<i>No new assignment</i>
7	10/10	<i>No Monday lecture—Columbus Day</i> Lec. 15: Functions: arguments, return values Lec. 16: Pointers; pointer arguments	W: 9.1-9.4 F: 11.1-11.5	Program 6 (due 10/21)
8	10/17	Lec. 17: ANSI library functions Lec. 18: PE4 (Functions) Lec. 19: One dimensional arrays	M: 21.1-21.2, 23.3 W: None F: 8.1	Program 7 (due 10/28)

Course Schedule (cont.)

Week	Date (M)	Lecture Topics	Reading	Programs
9	10/24	Lec. 20: Pointer arithmetic Lec. 21: PE5 (Arrays and strings) Lec. 22: Character arrays and strings	M: 12.1-12.3, W: None F: 13.1-13.4	Program 8 (due 11/4)
10	10/31	Lec. 23: C string functions Lec. 24: PE6 (Strings) Lec. 25: Two-dimensional arrays	M: 13.5-13.6 W: None F: 8.2, 12.4	Program 9 (due 11/14)
11	11/7	Lec. 26: Exam 2 Preview Wednesday, 11/9: Exam 2 <i>No Friday lecture—Veterans Day</i>	None	Project work
12	11/14	Lec. 27: Exam 2 Review <i>Tuesday, 11/15: Last day to withdraw</i> Lec. 28: File I/O Lec. 29: PE7 (2-D arrays, File I/O)	M: None W: 22.1-22.3 F: None	Project work
13	11/21	Lec. 30: Structures Lec. 31: Structures (cont.) <i>No Friday lecture—Thanksgiving Break</i>	M: 16.1-16.3 W: None	Project work
14	11/28	Lec. 32: PE8 (Structures) Lec. 33: Dynamic memory allocation Lec. 34: PE9 (Dynamic memory allocation)	M: None W: 17.1-17.4 F: None	Project work
15	12/5	Project work; additional lectures as needed	None	Project work
16	12/12	Lec. 35: Exam 3 Preview <i>Classes end Monday 12/12</i>	None	Project code due 12/12
	TBD	EXAM 3		