

Syllabus

CMSC 160: Intro to algorithmic design I

Fall 2019

Section 1: MWF 10am, Ruffner 354; Thu 9:30am, Ruffner G56
Section 2: MWF 11am, Ruffner 354; Thu 2:00pm, Ruffner G56
Website: <http://cs.longwood.edu/courses/cmsc160>

An introduction to problem solving and algorithmic design using an object-oriented programming language. Topics include programming logic, iteration, functions, recursion, arrays, memory management, user-defined data types, abstraction, and complexity analysis. 4 credits.

Professor: Don Blaheta
Office: Ruffner 334
Phone: x2191
Email: blahetadp@blahedo.org or blahetadp@longwood.edu
Office hours: Mondays 3–4pm; Tuesdays 2:30–3:30pm;
Thursdays 11am–noon; Fridays 1–2pm

Overview

Edsger Dijkstra¹ famously said that “computer science is no more about computers than astronomy is about telescopes.” What did he mean?

In this course, we will indeed spend plenty of time with computers, but my focus is ultimately to teach a different way of thinking about problems. The key skills you will develop are analytical problem solving and thinking in terms of abstractions. Should you choose to continue in computer science, you will use these skills in every CS course you take, whether you are writing programs in C++, or in some other language, or not at all. And if I really do my job, the precision of thought you learn here will help to shape how you think about problems in your other courses, even if you never take another CS course again.

Textbook and resources

The textbook is EITHER *C++ for everyone 2e*, by Cay Horstmann (ISBN 978-0-470-92713-7) OR *Brief C++ 3e*, by Cay Horstmann (no ISBN, order through bookstore). Despite the name difference, they are different editions of the same book; and despite

¹A pioneer of computer science. More info available on Wikipedia.

the edition difference, they're actually almost identical. The second edition has the advantage of being a physical book, and older (thus cheaper); the third is electronic only but has some end-of-section exercises that are integrated into the book. There will be regular assignments drawn from the book, but sharing with another student is possible; which edition and format you use is up to you.

You will be given an account on the department's computer systems, where we'll use the Linux operating system and a compile system built on the gcc compiler. You may also be interested in installing Linux on your own machine; if so, contact me and I'll help you get it set up. Some of the assignments, especially early in the semester, will use the classroom software at `repl.it`, so you will get an account there too (which we'll walk through how to sign up for it).

Course objectives

At the end of this course, the successful student will be able to:

1. implement algorithms using C++;
2. read, write, and debug programs that use standard control constructs such as if/else, loops, and functions;
3. appropriately choose among the fundamental programming data types; and
4. use standard design strategies for ensuring that programs are robust and readable.

Content

Graded work

I figure that I have on average about 12 hours of your time every week, including class and lab time as well as reading, practice, homework, and projects. If you find you're regularly spending substantially more time than this, please do come discuss it with me, so that we can ensure you're making the most effective use of your time. The work you do for this course will be evaluated as follows:

Preparation and participation. You need to be an active participant in this class: present, prepared, and on-task. The point for each day will be assigned using one of the following rubrics:

- Basic attendance: If you're there, you get the point!
- Participation: **1**: Attentive and on-task. **1/2** or **0**: Substantially late, sleeping, fussing with cellphone, etc.

- Reading quiz: Three questions, open-notes. **1**: Demonstrated that you read the assigned reading. $\frac{1}{2}$: At least some correct work on the quiz.

I will not, in general, tell you in advance which one I'll use on a particular day. These points are collectively worth 10% of the grade.

Lab work. The central goal of the course is that you learn to program, so the bulk of the work you do will be programming work before, during, and after our assigned lab periods. Except where otherwise noted, work associated with a particular lab is due at 4pm on the following Wednesday. Lab work will make up 40% of the grade.

Exams. There will be two exams, one in October and one during the finals period. Each will have a take-home component and a sit-down portion. The final will not be explicitly cumulative, though of course the material from the second half of the course builds on the earlier stuff. **You are not permitted to discuss the exams *at all*, with anyone other than me.** Each exam is worth 25% of the grade.

The sit-down portion of the final exams will be 8–10:30am on Wednesday, 11 December (for Section 2) and 3–5:30pm on Thursday, 12 December (for Section 1). You may choose to take your exam with the other section BUT you must notify me *in advance* if you plan to do so.

Breakdown

Prep/participation	10%	
Lab work	40%	
Exams	50%	(25 each)

Grading scale

I tend to grade hard on individual assignments, but compensate for this in the final grades. The grading scale will be approximately as follows:

A–	[85, 90)	A	[90, 95)	A+	[95, 100]
B–	[70, 75)	B	[75, 80)	B+	[80, 85)
C–	[55, 60)	C	[60, 65)	C+	[65, 70)
D–	[40, 45)	D	[45, 50)	D+	[50, 55)

While there will be no “curve” in the statistical sense, I may slightly adjust the scale at the end of the term if it turns out some of the assignments were too difficult. Final grades of A+ are recorded as an A in the grading system. Final grades below the minimum for D– are recorded as an F.

Calendar (tentative)

Wk	M	W	R	F
August				
1	26 §§1.4–1.5 How to read a textbook Variables Input	28 §§2.1, 2.3.1, 2.5.1 Account setup Hello, world!	29 Lab 1 drill Mad Libs	30 §§1.6, 2.2.4; pp34–35* Errors Numeric range Type conversion
September				
2	[Labor Day no class]	** 4 §§3.1–3.3 if/else, if alone else if Flowcharts Comparisons	5 Lab 2 drill Test cases Conditional execution	6 §§3.4–3.5; pp67–69* Comparison errors Blocks Nested if Flowcharts cont'd
3	9 §§2.2, 2.5.2 Arithmetic expressions String concatenation Math functions bring laptops	11 §§1.7, 2.4 Order of operations Algorithm design Do it by hand	12 Lab 3 drill Tinkerblocks	13 §2.5.4 String functions Characters
4	16 — Vectors	18 — Vectors and loops	19 Lab 4 drill Strings and vectors	20 (p79*, §4.2) Tracing Debugging
5	23 §§2.1.4, 2.2.2 Assignment, increment Accumulation const, constexpr	25 §§4.7.1, 4.7.2, 4.7.4 Sum/average Counting things Min/max	26 Lab 5 drill Standard loops Reading and modifying code	27 §§4.1, 4.2 while General loops
October				
6	30 §§4.5, 6.7.2 Input .push_back	2 p153 Loop-and-a-half break	3 Lab 6 drill Loop practice	4 *** §4.4 do More loop practice Exam 1 TH out
7	7 — Exam 1	9 §5.1 (& p162*) Functions Testing Stubs	10 Lab 7 drill Unit testing Debugging	11 §§5.2–5.4, §4.7.3 Parameters Return values Find first match Early return

* pp34–35 (3e) = p38–39 (2e); pp67–69 (3e) = pp85–86 (2e); p79 (3e) = p97 (2e); p162 (3e) = p217 (2e)

** **3 September:** Deadline to add/drop classes (5pm)

*** **4 October:** Deadline to elect pass/fail option (5pm)

Wk	M	W	R	F
	October			
8	[Fall Break no class]	16 §§4.10*, 5.5 Random numbers Void functions Recursion	17 Lab 8 drill Writing functions	18 §3.4 Nested conditionals Boolean variables Boolean functions
9	21 §§3.5, 3.7 Flowcharts Boolean operations Truth tables	23 pp87–89*, §3.6 Complex booleans DeMorgan's Law Test case coverage	24 Lab 9 drill Boolean logic	25 — Bundling data struct
10	28 — Struct parameters Struct return values	30 — Vectors of structs	31 Lab 10 drill Reading and debugging code	November 1 §§5.9, 6.7.3, 6.7.4 Reference parameters Functions on vectors Vector algorithms
11	4 §6.2.7 Filtering data Removing data Mapping data	6 ** — Producing vectors of structs	7 Lab 11 drill Weather statistics	8 §§4.3, 4.7 C-style for Nested loops
12	11 — Sorting	13 §6.6 2D arrays Vectors of vectors	14 Lab 12 drill Lijnenspel	15 §§2.3.2, 8.3 Formatted output newline and tab
13	18 §§8.2, 8.4 Line input Streams	20 §§6.1, 7.1, 7.2 Arrays, pointers Arrays as pointers	21 — Lab work day	22 §7.3 C strings .c_str printf
14	25 §9.6* Constructors	[Thanksgiving no class]	[Thanksgiving no class]	[Thanksgiving no class]
	December			
15	2 — Algorithm design Modeling and simulation	4 — Modeling and simulation	5 Lab 13 drill Modeling and simulation	6 — Modeling and simulation <i>Exam 2 TH out</i>
	<i>Section 2 Exam 2: Wed 11 Dec, 8–10:30am</i> <i>Section 1 Exam 2: Thu 12 Dec, 3–5:30pm</i>			

* §4.10 (3e) = §4.9 (2e); pp87–89 (3e) = pp107-08 (2e); §9.6 (3e) = §9.5 (2e)

** **6 November**: Deadline to withdraw from a class (5pm)

Policies

You can find several university-wide course policies at <http://www.longwood.edu/academicaffairs/syllabus-statements/> .

Support

This is an introductory course. That means that what is covered is an important basis for other work in the field, *not* that it is supposed to be obvious, or easy. So don't feel bad if something doesn't click right away. Never hesitate to ask me a question; I'll usually at least give you a hint as to where to look next.

I'm in my office a lot (not just during posted office hours). Feel free to come in and ask questions (or just to talk). If you can't catch me in my office, email is probably your best bet.

You should also make use of your fellow students as resources. While you can't copy each other's work (see the collaboration policy), studying together is a great idea, and asking and answering questions of other students is actively encouraged.

Accommodations

If you have any special need that I can accommodate, I'm happy to do so; come speak to me early in the term so we can set things up. If you have a documented disability, you should also contact Longwood's Office of Disability Resources (Brock Hall, x2391) to discuss some of the support the college can offer you. All such conversations are confidential.

Honor code policy

Above all, I ask and expect that you will conduct yourself with honesty and integrity—and not to ignore the other ten points of the Honor Code, either. Take pride in what you are capable of, and have the humility to give credit where it is due.

The two main forms of academic dishonesty are “cheating” and “plagiarism”. “Cheating” is getting help from someplace you shouldn't, and “plagiarism” is presenting someone else's idea as if it's your own. If you ever find yourself inclined towards either of these, know that there are always other, better options. Persevere! See my website² for some discussion and examples of how to steer clear of these problems, and feel free to come talk to me if you need help finding some of those other options (even if it's for another course).

²<http://cs.longwood.edu/~dblaheta/collab.html>

Cheating or plagiarism (on any assignment) will normally receive a *minimum* penalty of a lowered *course* grade, ranging up to an F in the course. Cases will also be turned in to the Honor Board. But: I believe in your potential, and I hope that you will, or will grow to, observe this policy not simply to evade punishment but positively as a matter of character.

Attendance and late policy

Attendance is required, and assignments must be turned in on time. That said, if you have a good reason to miss class or hand something in late, I tend to be fairly liberal with extensions if you ask in advance. (Good reasons do include assignments due for other classes.) (And medical and family emergencies are exempted from the “in advance” part, of course. But contact me ASAP.)

Frequent absence will result in a lowered participation grade; habitual absence may in extreme cases result in a failing grade for the class. *Unexcused* late assignments will normally be given a zero.

Inclement weather policy

I don't plan to cancel class for weather unless the entire college shuts down. If you are commuting or are otherwise significantly affected by a weather event, use your own best judgement; and if you do miss class for this reason, contact me as soon as possible to make up missed work.

Early bird policy

Nobody's perfect, and on occasion an assignment gets written a little unclearly (or, once in a while, with an actual error in it). If you catch one and bring it to my attention early, so that I can issue a clarification or correction to the rest of the class, there'll be some extra credit in it for you.