

Syllabus

CMSC 160: Intro to algorithmic design I

Fall 2015

Lecture: MWF 12 noon, Ruffner 356
Lab: Tue 2:00pm, Ruffner G56
Website: <http://cs.longwood.edu/courses/cmsc160/3>

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.

Professor: Don Blaheta
Office: Ruffner 337
Phone: x2191
Email: blahetadp@blahedo.org ¹
Office hours: Mon/Wed 4–5:30pm; Tue/Fri 11am–noon

Overview

Edsger Dijkstra² 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.

¹Or blahetadp@longwood.edu if you’d rather, but I prefer the off-campus one and check it more frequently.

²A pioneer of computer science. More info available on Wikipedia.

Textbook and resources

The textbook is *Programming: principles and practice using C++* 2e, by Bjarne Stroustrup (ISBN 978-0-321-99278-9). There will be regular assignments drawn from the book, but sharing with another student is possible.

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.

Course objectives

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

1. write and debug algorithms using standard control constructs such as `if/else`, loops, and functions,
2. appropriately choose among the fundamental programming data types and explain some of their limitations,
3. process data that is stored in sequential (list) form,
4. read and comfortably work with the fundamental syntax and semantics of a C++ program, and
5. ensure that programs are robust, readable, and efficient, using standard design strategies.

Content

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, 100)	³
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.

³Alas, no A+, unfortunately.

Calendar (tentative)

Wk	M	T	W	F
	August			
1	24 — Introductions Overview	25 §§1.1–1.5 Parts of a computer Account setup	26 §1.6 Stages of design How to read textbooks	28 §§2.1–2.2 Hello, world
2	31 * §§2.3–2.5 Compiling and linking	September		
		1 Ch.2 Drill Lab 1: Hello, world	2 §§3.1–3.2 Input and variables	4 §3.3 Types
3	[Labor Day no class]	8 Ch.3 Drill Lab 2: Mad Libs	9 §3.4 Operations	11 §§3.5–3.6 Initialisation and assignment
4	14 §3.7 Variable naming	15 Lab 3 Drill Lab 3: Test cases and string operations	16 §§3.8–3.9 Types and type conversion	18 §§4.1–4.2 What is computation? Objectives
5	21 §4.3 Expressions	22 Lab 4 Drill Lab 4: Design process	23 §§4.4–4.4.1 Statements if, else, else if	25 — Boolean logic and, or, not
6	28 §4.4.2 Iteration while and for	29 Lab 5 Drill Lab 5: Loops	30 §4.5 Functions Exam 1 TH out	October
				2 ** — Exam 1
7	5 §§4.6–4.7 Vectors	6 Ch.4 Drill Lab 6: Processing lists: Stats package	7 §§24.1–24.2 Numeric precision Overflow	9 §§18.6–18.6.3 Arrays Arrays vs vectors

* **31 August:** Deadline to add/drop courses (5pm)

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

Wk	M	T	W	F
October				
8	[Fall Break no class]		14* §§5.1–5.4 Compiler and linker errors	16 §§5.5–5.7 Run-time errors Exceptions and logic errors
9	19 §§5.8–5.10 Specification Debugging	20 Ch.5 drill Lab 8: Code reading and debugging: Divisors	21 — Arrays revisited	23 §§27.5–27.6 C strings stdio
10	26 §§24.3–24.5 2D arrays Matrix	27 Lab 7 drill Lab 7: Lijnspeel	28 §§6.1–6.3.1 Case study: calculator	30 §§6.3.2–6.3.5 Tokens class and struct
November				
11	2 §6.4 Tokens, cont'd Grammars	3 Ch.6 Drill Lab 9: Code reading and debugging: Calculator	4 §§10.1–10.4 File I/O bring laptops	6 §§11.1–11.2 Formatting bring laptops
12	9 §§11.4–11.5 String streams Line-based input	10 Lab 10 drill Lab 10: Processing and formatting structs: Stats package/weather	11 §§7.1–7.2 I/O revisited	13 §7.3 Error handling
13	16 §§7.4–7.5 Negative numbers and remainders	17 Lab 11 drill Lab 11: Sorting	18 §§8.1–8.4, 8.7 Declarations vs definitions Header files	20 §§8.5.1–8.5.7 Function call specifics
14	23 §22.1 History of CS Ideals	24 — Lab 12: Code reading and refactoring: Lijnspeel	[Thanksgiving no class]	
December				
15	30 §22.2 Programming languages	1 — Lab 13: Lijnspeel continued	2	4 — TBA Exam 2 TH out
7 11:30am–2:00pm Exam 2				

* **14 October:** Deadline to withdraw from courses (5pm)

Graded work

I figure that I have about 10–15 hours of your time every week, including class and lab time as well as reading, practice, homework, and projects. If you find you're spending more time than this, please do come discuss it with me, and we'll see what we can work out. 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. $\frac{1}{2}$ **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 Monday. Lab work will make up 40% of the grade.

Exams. There will be two exams, one in early 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.

Breakdown

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

Policies

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 (Graham 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 website⁴ for some discussion and examples of how

⁴<http://cs.longwood.edu/~dblaheta/collab.html>

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).

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 get a note from a dean.)

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.