Syllabus
CMSC 140: Introduction to programming

Fall 2024

Section 3:	MWF noon, Rotunda G56
Websites:	`https://longwood.instructure.com/courses/1314215`
	`http://www.cs.longwood.edu/courses/cmsc140`

A first course in computer programming, intended for students with no previous experience in writing computer programs. Emphasis will be placed on practical programming skills; assignments will primarily use the Python programming language. Students will cover the fundamental control structures and will learn to process real data stored in sequential lists and in key-value pairs. Students may not enroll in this course if CMSC 160 has already been completed. 3 credits. FQRC, SI.

Professor:	Don Blaheta
Office:	Rotunda 334
Phone:	x2191
Email:	`blahetadp@longwood.edu`
100% Office hours:	Mondays 3–4pm; Tuesdays 11am–noon;
	Wednesdays 10:30–11:30am; Fridays 2–3pm

Overview

In the last decade, we have seen huge increases in the computational thinking required in a wide variety of disciplines; across the natural and social sciences, the arts, and even in the humanities, researchers and practitioners are finding that they have data, they need to process it, and no completely off-the-shelf solution will do. They need to write a program to solve a problem.

In this course, we will introduce the concepts and ways of thinking required to write straightforward programs to process a variety of kinds of data.

Textbook and resources

The primary textbook is *A Practical Introduction to Python Programming*, by Brian Heinold. It is free and can be found online.[1] There will be regular in-class and out-of-class assignments drawn from the book.

There will also be readings assigned from *The Little QR Booklet*, by me. This is also free and will be available on the Canvas page.

You will be expected to have a computer that can connect to the internet and various websites, including one called `codeboard.io`, which is where you will be writing your programs. Some days are designated lab/laptop days, and if you are attending in-person you'll be expected to bring your laptop with you on those days. A phone is not sufficient for this; a tablet might be, as long as you have a decent keyboard for it.

AI Policy

My general feeling about AI is this: AI is a tool. Use it when it's helpful, don't use it when you could do it better or faster yourself.

That said, there are certain skills that programmers and computer scientists will need to develop and execute without the help of AI, slightly because AI might not *always* be available but mostly because you'll need to be able to evaluate and debug the code that the AI (or other programmers) have given you. Thus for assignments that are about *developing* your programming skills (labs, homeworks, projects), I'm going to discourage use of AI until you've given a few solid attempts without. For assignments that are *assessing* your skills (exams) I'll have specific instructions on whether you are or are not allowed to use generative AI to assist. *In general* tasks that you're doing on your own time will permit use of AI, but please attend to specific instructions on each assignment.

I will expect that when you *do* use generative AI, you will document it: say which AI system you used and what help it gave you. Some assignments will have more detailed instructions on what and how to document this.

(Note that although Longwood's Honor Code does not inherently ban the use of AI, some other professors seem to think it does, so for your safety you should check with each professor before using it in their class.)

[1] `https://www.brianheinold.net/python/python_book.html`

The section formerly known as "Covid-19 notes"

I have a few policies that originally evolved in response to the pandemic but I've decided they're just good policy so I kept them. Here's the gist: It's really easy to keep zoom open for every class, and it's not nearly as good as in-person attendance but way better than total absence. So I open Zoom every day and ask you to make good choices.

Attending class. There are two ways you can attend class: in person, or via Zoom link. Either mode of attendance is sufficient to mark you as "attending" (not necessarily engaging or participating). If you attend via Zoom link,

- you must have a reason, and
- you must say what it is,

but I don't need any medical detail and if it's not directly covid-related I'm not going to police that. (Again: be an adult and make good choices.) The Zoom experience is nowhere near equivalent to the in-person experience and is not a replacement for it; but if you are quarantined, or otherwise just can't attend in person on a particular day, zooming is better than total absence.

Zooming vs masking. Although we've moved from "pandemic" to "endemic" on Covid-19, I'd just like to remind everyone that masking is still a tool in our toolkit: if you're feeling a bit sniffly, you can still wear a mask. We all have masks, we all got really good at wearing them, and it's a courtesy to your classmates to take this easy step to decrease the likelihood of spreading anything. (Including colds and other stuff! Masks help us not spread *lots* of things.)

What if the professor gets sick? Same as for students: if I'm feeling a little sniffly, I'll wear a mask, and if I am more seriously sick (but well enough to teach), I'll zoom myself into the class. If necessary I can teach from a zoom window on the projector screen (and have done so!); I'll post to Canvas and send an email with instructions as soon as I know I need to do this.

Course objectives / Student learning outcomes

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

1. write programs to handle both numeric and textual data,

2. use standard programming control constructs such as if/else, loops, and functions,
3. read and process real-world data using flat text files,
4. count, filter, and transform data presented in sequential (list) form, and
5. produce descriptive statistics on numeric data.

Core curriculum objectives. In addition to the course-specific objectives above, this course shares the objectives of the core curriculum as a whole. During this course, the successful student will:

6. engage in creative inquiry and cultivate curiosity,
7. develop foundational knowledge and skills in the discipline (e.g. how to communicate, study, read, etc.), and
8. create and deliver oral messages appropriate to audience, purpose, and context.

Quantitative reasoning objectives. As a Quantitative Reasoning course in the core curriculum, this course shares the following objectives as well. At the end of this course, the successful student will be able to:

9. formulate a question/issue using appropriate mathematical, algorithmic, and/or statistical terms, and explain the decision process behind the choices made in that formulation;
10. use mathematical, algorithmic, and/or statistical methods to gather and/or analyze data—justification of the methods chosen should be included;
11. determine the reasonableness of an answer and/or evaluate the explanations of data for reasonableness, and understand the limitations behind the methods used in the previous outcome; and
12. interpret the results of a mathematical, algorithmic, and/or statistical analysis, and present the interpretation in a context appropriate for a broader audience.

Faculty objectives

Per section II-O-III-F-3 in the faculty manual, faculty teaching this speaking-infused course are expected to:

13. integrate speaking opportunities, exercises, and/or assignments with Core outcomes and individual course objectives, so that students may simultaneously master course content and develop and improve their oral communication skills;

14. provide explicit instruction to aid student understanding of speaking appropriately for audiences in the relevant context or discipline; and

15. provide appropriate and timely peer and/or instructor feedback on student oral communication to allow opportunities for students to improve their performance on subsequent assignments.

The Pillar Objective

16. Analyze which quantitative reasoning methods best address different types of questions and apply them to various problems in context.

Content

Graded work

I figure that I have, on average, about 9 hours of your time every week, including class time as well as reading, practice, homework, and project work. 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:

Engagement. You need to be actively engaged in this class. Engagement comes in many forms, but I expect that you will be interacting with your classmates, and with me, in class. General engagement will be evaluated in two-week blocks—so you don't need to artificially say a thing every day—and it's ok if most of your engagement is via the discussion boards as long as *some* of it is spoken out loud. In addition, there will be

occasional required interactions via Canvas that will be considered part of the engagement grade. Engagement makes up 10% of the course grade, and more than half of that ($> 5\%$) is in the fortnightly general participation blocks.

Lab work (and homework). The central goal of the course is that you learn to program, so the bulk of the work you do will be "lab" work before, during, and after class. On the graded assignments, a correct answer by the initial duedate is required for full credit, but up to 90% of the points can be recovered in later revisions by fixing errors in the first version—which is also an excellent way to study and learn. Graded homework will make up 25% of the grade. Collaborative

Course project. Your course project will involve working with a data set in your area, building a program to process that data, and writing up your results. Evaluation will be based on the program code itself as well as your written and verbal proposals and conclusions based on the results. The project will be worth a total of 25% of the final grade. Collaborative

Exams. There will be two exams, one in early October and one in early December. **You are not permitted to discuss the exams *at all,* with anyone other than me.** Each exam is worth 20% of the grade. (40% total) Non-collaborative

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.

Note that *individual* grades recorded in Canvas should be accurate (and you should let me know if there's a data entry error!), but *averages* as computed by Canvas sometimes are not, if the averaging is complex or (especially) if an individual student has a special case scenario. The reference gradebook is

my own spreadsheet, and while I will try to make Canvas reflect it (including averages) as well as I can, Canvas can't always handle it.

Study and practice

As in many disciplines, the most effective way to learn this material is to *do* it—the first attempt being difficult and often not fully correct, but improving with continued effort and practice. Trying to "study" simply by *reading about* it is not going to be enough (though it's the right place to start). This course is structured to help you study how to program (and to *learn* to study how to program): in general a concept will be first introduced in a reading, after which we'll have some low-stakes in-class practice where the point is to try some things and see what works (and analyse what doesn't). After these supervised attempts, you'll try some more problems on your own, outside of class, with feedback and a grade. When we get to the major assessments, like the exams or the project, you will have been studying for far longer than the day or two leading up to the exam, and you'll be familiar with the kinds of practice you need in order to be successful. SLO 7

Course project

The purpose of the course project (worth a total of 100 points) is for you to connect this course content directly with a professional or personal interest of your own, to perform a complete start-to-finish piece of data analysis using the tools and techniques we'll be developing all semester, and to communicate the intent and results of your inquiry to others. In that sense, it serves to integrate a number of course goals; the technical component (addressing SLOs 1–5) is central but only 40% of the grade, so it's important not to lose sight of the others. An explicit goal of this project is to help you decide what *you* might be curious about, and to encourage you to indulge and pursue that curiosity, and to engage in creative inquiry to answer and feed that curiosity. Another explicit goal of the project is to serve as content about which you'll need to communicate; this kind of longer-running project gives me a vehicle for a more formal kind of feedback on your communication (and a well-defined place to assess it).

Breakdown

The first piece of this will be to find a core question that is interesting and a data set that can answer it. What makes a question "interesting"? That's largely up to you! You'll submit an initial proposal writeup that includes (a link to) the data set and some explanation of how the data represents information of interest and formulates your (proposed) question in terms of the algorithm you intend to use to solve it; I'll respond to it and (if necessary) help you refine it into something workable. This writeup is worth 12 points. 12 points is 12% of the project grade, and thus 3% of the course grade. SLO 6, 7, 9

After you've settled on a question, you'll give a very brief "elevator speech" to the class—90 second max—telling us what your question is and why it's interesting (to you, and maybe to us as well). 8 points. 8 points is 8% of the project grade, and thus 2% of the course grade. SLO 6, 8, 9

The core of the assignment will of course be to write a program that implements your chosen algorithm to perform the data analysis you need. The program itself will be worth 40 points. 40 points is 40% of the project grade, and thus 10% of the course grade. SLO 1–5, 6, 10

In the last week or so of the course, after you've done most or all of the implementation work, you'll give a "lightning talk" (same as an elevator speech, but more technical) that explains to us what your algorithm will do and how it will answer your question. 16 points. 16 points is 16% of the project grade, and thus 4% of the course grade. SLO 7, 8

After you have successfully run your algorithm to perform the data analysis, the final step is to interpret its output. You'll also give a final elevator speech that presents your result, explaining why it is reasonable (i.e. not obviously the result of a bug!), and what the numbers mean in the context of the problem—answering your original core question. You'll also distill the entire QR process into a written summary of the result. The speech is 16 points; the writeup is 8. 16 points is 16% of the project grade, and thus 4% of the course grade. 8 points is 8% of the project grade, and thus 2% of the course grade. SLO 11, 12 SLO 8, 12

The speaking infusion

This course offers many opportunities to develop your speaking skills, and provides a sort of ramp from low-stakes speaking opportunities, with feedback, leading up to the speaking worth more points at the end of the semester. At SLO 7, 8; FO 13–15

each stage, you will receive explicit instruction in each speaking style used in this course.

As part of the participation grade, there will be frequent and regular chances to explain problems and answers in small groups with your classmates. These points will be nearly automatic (if you're present and participating), and as I circulate around the room to answer questions I can also give some light feedback. SLO 7, 8; FO 13, 15

The first elevator speech (see "Course project" above) will be the first of the speaking opportunities you can specifically plan for, and thus will be held to a somewhat higher standard; but it is still only worth 2% of the final grade, to give a chance for feedback on this format. I will send you comments by email or via Canvas that will be geared to help you improve your performance on the later speeches. SLO 7, 8; FO 13, 14, 15

The last two speeches will follow basically the same format as the first elevator speech, but will be worth 4% of the final grade each. SLO 7, 8

Calendar (topic/content)

Days marked **–L** mean you should bring your laptop that day, as we will be doing lab work.

Wk	M	W	F
	August		
1	**26** — Introduction The idea of an algorithm	**28 –L** — How to read/use a textbook Hello world	**30** Ch. 1 Input/output, comments
	September		
2	[Labor Day **no class**]	* **4** Ch. 2 for loops `range`	**6 –L** — (continued)
3	**9** — Quantitative Reasoning	**11** — What makes a workable quantitative question?	**13** Ch. 3 Arithmetic Order of operations
4	**16 –L** — Random, Math	**18** — Limitations of a QR process	**20** Ch. 4 `if` and blocks Comparisons and booleans `and or not`
5	**23 –L** — Blocks and nesting Flowcharts	**25** — `else`, `elif`	[prof absent **no class**]
		October	
6	**30** — Interpreting results of a quant question	**2** — Test cases Practicing with codingbat	**4** ** — **Exam 1**
7	**7** Ch. 6 Strings	**9 –L** — String slicing More string practice	[Fall Break **no class**]

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

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

Wk	M	W	F
	October		
8	**14** Ch. 7 Lists List operations Looping over lists	**16** Ch. 5 Standard list-loop algorithms	**18** — QR: Checking results for reasonability Resolving limitations Putting it together
9	**21 –L** — `split`, `join` Multiple assignment	**23** — Adding/removing items Writing the quant program Project overview	**25 –L** — List-loop practice
			November
10	**28 –L** Ch. 12 Text files	**30 –L** — CSV files **Project proposal due**	**1** Ch. 11 Dictionaries
11	**4** Ch. 8 Dictionary practice List comprehensions	**6** * 8.5 2D lists	**8** Ch. 13 Function basics
12	**11 –L** — Scope Using functions practically	**13 –L** — Speech practice Project work day	**15** — Functions, cont'd
13	**18** — **Elevator speeches**	[Research Day **no class**]	**22** Ch. 9 `while` loops `break`, `continue`
14	**25** — **Lightning talks**	[Thanksgiving **no class**]	[Thanksgiving **no class**]
	December		
15	**2** — Review	**4** — **Exam 2**	**6** — **Elevator speeches**

Project program and writeup due Thu, 12 December, 5:30pm
Exam time reserved for (online) speech overflow if needed: Thu, 12 Dec 3–5:30pm

* **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.

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.

"Office hours"

If I'm in my office and my door is open, that means I'm available for you to drop in and ask questions, and I'm happy to turn on my "office hours" zoom link so you can join me that way instead. At least four hours a week I've designated as 100% hours, i.e. there's a nearly 100% chance I'm available at those times.

But I'm in my office a lot and I'd like to effectively communicate to you when you're most likely to catch me, so if you look at my office schedule on my website or linked from Canvas, you'll also see many hours listed with other percentages like 60% or 40 or 10, as informal estimates of the probability I'll have office hours in that block for drop-in questions. (If you want more certainty, you can always give me advance notice and be extra sure I'll be there at whatever time!)

If you can't catch me in my office, email or Slack is probably your best bet.

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 Accessibility Resources Office (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 website[2] 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).

Cheating or plagiarism (on any assignment) will normally receive a *minimum* penalty of lowering the *course* grade by a full letter, and may range at my discretion 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.

[2] `http://cs.longwood.edu/~dblaheta/collab.html`

Inclement weather policy

I don't plan to cancel class for weather unless the entire college shuts down; and if the campus closes, I'm likely to hold class in some form by zoom instead (check your email). If you are commuting or are otherwise significantly affected by a weather event, use your own best judgement (and remember that zoom is an option); and if you do miss class for this reason (e.g.: power's out too), 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.

This document was written and prepared without the use of generative AI.