Syllabus
CMSC 140: Introduction to programming

Fall 2017

Section 1:	MWF 10am
Section 2:	MWF 11am
Location:	Ruffner 356
Website:	`http://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. 3 credits. Students may not enroll in this course if CMSC 160 has already been completed. 3 credits.

Professor:	Don Blaheta
Office:	Ruffner 337
Phone:	x2191
Email:	`blahetadp@blahedo.org` or `blahetadp@longwood.edu`
Office hours:	Mon 2–3pm, Tue 11am–noon, Thu 1–2pm, Fri 1–2pm

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 textbook is *Python for Everyone*, 2e, by Cay Horstmann and Rance Necaise (ISBN 978-1-119-05655-3). There will be regular in-class and out-of-

class assignments drawn from the book; you should bring it to every class.

The language of the course is Python 3, and you'll need to install it on your laptop, along with the IDLE editor. You will frequently be expected to bring your laptop, with Python and IDLE already installed on it, with you to class.

Course objectives

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,
5. produce descriptive statistics on numeric data, and
6. create and deliver oral messages appropriate to audience, purpose, and context.

Content

Graded work

I figure that I have about 8-10 hours of your time every week, including class 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. Half of the grade for this component will be evaluated daily; 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. **½ or 0**: Substantially late, sleeping, fussing with cellphone, etc.
- Reading quiz: Three questions, open-notes. **1**: Demonstrated that you read the assigned reading. **½**: Some correct work on the quiz.

You won't, in general, know in advance which I'll use on a particular day.

The other half will be more occasional: from time to time you will be called on to informally explain some work you've done on the board: what it does, how it works, what its result would be, and so on. Points for these explanations will not be based primarily on exact correctness (since it will be on content that is very new to you, and it's ok to get it wrong before you get it right), but on structure and flow and communication; for instance, correctly explaining part and then clearly articulating a question about a piece you don't fully understand will be just fine for this.

These points are collectively worth 10% of the grade: 5 for the point-of-the-day, and 5 for the informal explanations.

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 our assigned class periods. This work will make up 25% of the grade. Collaborative

Final project. Your final project in the course 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 late September and one during the finals period. 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 20% of the grade. Non-collaborative

Breakdown

Prep/participation	10%	
Lab/homework	25%	
Final project	25%	
Exams	40%	(20 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, 100)	[1]	
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.

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.

[1] Alas, no A+, unfortunately.

Final project

The purpose of the final project (worth a total of 100 points) is to help you connect this course content directly with a professional or personal interest of your own, and to perform a complete start-to-finish piece of data analysis using the tools and techniques we'll be developing all semester.

The first piece of this will be to find a question that is interesting and a data set that can answer it. 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.

After you've settled on a question, you'll give a very brief "elevator speech" to the class—90 seconds max—telling us what your question is and why it's interesting (to you, and maybe to us as well). 8 points.

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.

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.

After you have successfully run your algorithm to perform the data analysis, the final step is to interpret its output. You'll write a formal (but brief) summary of the 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. You'll also give a final elevator speech to the class wherein you answer your original question. The writeup is 8 points, the speech 16.

Speaking and presentation

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.

First of all, as part of the point-of-the-day 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.

The more occasional version of this—which I'll be careful to rotate around so that each student has roughly the same number of opportunities—will have you presenting these explanations individually to the whole class, and I'll have a chance to give more targeted feedback at that time. The occasional speaking is collectively worth 5% of the grade.

The first elevator speech (see "Final 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.

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.

Final exam

The final exams are scheduled at the following days and times:

Mon, 4 Dec	8–10:30am	Sec. 2 (the MWF11 section)
Wed, 6 Dec	3–5:30pm	Sec. 1 (the MWF10 section)

If you wish to take the final during the other section's time slot, contact me in advance; I will allow it on a first-come first-serve basis (but seats may be limited).

Calendar (tentative)

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

Wk	M	W	F
	August		
1	**21** — Introduction The idea of an algorithm How to read a textbook	**23 –L** 1.1–1.3 Hello world	**25** 1.6 Errors
			September
2	**28** * 1.7 Algorithms	**30** 2.1–2.2 Arithmetic and variables	**1 –L** — (continued)
3	[Labor Day **no class**]	**6** 2.4–2.5 Strings Input and output	**8 –L** — (continued)
4	**11 –L** — Ch. 2 practice	**13** — (continued)	**15** 3.1 `if`, `else` control flow
5	**18 –L** 3.2 Relational operators on numbers, strings	**20** 3.3–3.5 Nested `if`, `elif` Flowcharts	**22 –L** — (continued)
6	**25 –L** — (continued)	**27** 3.7 `and` `or` `not`	**29** ** — **Midterm exam**
	October		
7	**2** 3.6, 3.8 Test cases More strings	**4** 4.1–4.3 Loops Flowcharts	**6 –L** 4.6 Debugging loops `for`

* **28 August**: Deadline to add/drop classes (5pm)

** **29 September**: Deadline to elect pass/fail option (5pm)

Wk	M	W	F
	October		
8	**9** * 6.1–6.2 Lists and list operations	**11 –L** — (continued)	**13 –L** 7.1–7.2 Files **Project proposal due**
9	[Fall Break **no class**]	**18 –L** — Loop setup practice	**20** 4.5 Standard loop algorithms
10	**23 –L** — (continued)	**25** — **Elevator speeches**	**27** 6.3 Standard list algorithms
		November	
11	**30 –L** — (continued)	**1 –L** 4.8 Standard string algorithms	**3 –L** 4.7 Nested loops
12	**6** 8.1–8.2 Sets and dictionaries	**8 –L** — (continued)	**10** 5.1–5.4 Functions
13	**13 –L** — (continued)	**15** 5.5, 6.4 Functions on lists	**17** 6.7 Tables
14	**20** — Project work day	[Thanksgiving **no class**]	[Thanksgiving **no class**]
			December
15	**27** — **Lightning talks**	**29** 7.5–7.6 Exceptions	**1** — **Elevator speeches** **Final project due**

Final exam, Section 2 (11am section): Mon 4th, 8–11:30pm
Final exam, Section 1 (10am section): Wed 6th, 3–5:30pm

* **9 October**: Deadline to withdraw from a class (5pm)

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 website[2] for some discussion and examples of how

[2] `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 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.