Syllabus
CMSC 140: Introduction to programming

Spring 2020

Section 1:	MWF 10am, Ruffner G54
Section 2:	MWF 1pm, Ruffner 352
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. Students may not enroll in this course if CMSC 160 has already been completed. 3 credits. FQRC, SI.

Professor:	Don Blaheta
Office:	Ruffner 334
Phone:	x2191
Email:	`blahetadp@blahedo.org` or `blahetadp@longwood.edu`
Office hours:	Tuesdays 1:30–3, Wednesdays 2–3:30, Fridays 11–12

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 *Automate the boring stuff with Python*, 2e, by Al Sweigart (ISBN 978-1-59327-992-9). There will be regular in-class and out-of-class as-

signments drawn from the book; you should bring it to every class. It is available online for free at `http://automatetheboringstuff.com` and if you use that version you can "bring it to class" by having a suitable internet-connected device.

The language of the course is Python 3, and you'll need to install it on your laptop, along with the Mu editor. You will frequently be expected to bring your laptop, with Python and Mu already installed on it, with you to class. Python can be downloaded and installed from `https://www.python.org/downloads/`, and Mu can be downloaded and installed from `https://codewith.mu/`.

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.

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:

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

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 late February and one during the finals period. The final will not be explicitly cumulative, though of Non-collaborative

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.

Breakdown

Prep/participation	10%	
Lab/homework	25%	
Course 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, 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.

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

SLO 7

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.

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 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 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. SLO 11, 12

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. 8 points is 8% of the project grade, and thus 2% of the course grade. 16 points is 16% of the project grade, and thus 4% of the course grade. 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 each stage, you will receive explicit instruction in each speaking style used in this course. SLO 7, 8; FO 13–15

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. SLO 7, 8; FO 13, 15

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. SLO 7, 8; FO 14, 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

Final exam

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

Mon, 4 Dec	3–5:30pm	Sec. 2 (the MWF1pm section)
Tue, 5 Dec	3–5:30pm	Sec. 1 (the MWF10am 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 (topic/content)

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

Wk	M	W	F
	January		
1		**15** — Introduction The idea of an algorithm How to read a textbook	**17 –L** pp3–13 Hello world
2	[MLK Day **no class**]	**22** * pp13–20 Parts of a program Comments Input / output	**24** — Arithmetic and operations
3	**27** pp21–28 `if` and blocks Comparisons and booleans `and` `or` `not`	**29 –L** — (continued)	**31** pp28–35 `else`, `elif` Flowcharts
	February		
4	**3 –L** — (continued)	**5** pp35–43 `while` loops `break`, `continue`	**7 –L** pp51–55 (continued)
5	**10** pp44–51 `for` loops `range`	**12 –L** — more loop practice	**14** pp77–85 Lists List operations
6	**17** pp85–88 Multiple assignment Random choice Looping over lists	**19 –L** — Standard list-loop algorithms Test cases	**21** ** pp88–93 Adding and removing items Other list operations
7	**24 –L** — Loops with adding and removing	**26** — More list-loop practice	**28** — **Midterm exam**

* **22 January**: Deadline to add/drop classes (5pm)

** **21 February**: Deadline to elect pass/fail option (5pm)

Wk	M	W	F
	March		
		SPRING BREAK	
8	**9** pp57–63 Function basics	**11** pp65–68 Scope More function practice	**13** pp201–207, 215–219 Files and paths Reading and writing
9	**16 –L** — Working with functions and files	**18** pp371–382 CSV files Project overview	**20** — CSV files, continued **Project proposal due**
10	**23** pp129–138 More string operations	**25** pp138–140 Joining and splitting strings	**27** pp140–153 Strings continued
		April	
11	**30** * — Speech practice Project work day	**1** — **Elevator speeches**	**3** pp267–279 Getting web files Bare basics of HTML
12	**6** pp279–291 Web scraping	**8** pp111–128 Dictionaries	**10 –L** — Practice with dicts
13	**13** pp447–458 Image manipulation	**15** pp459–472 Pixel-level control Shapes and text	**17 –L** — Practice with images
14	**20** — Project work day	[Research day **no class**]	**24** — **Lightning talks**
			May
15	**27** pp358-367 Manipulating `.docx` files	**29** — **Elevator speeches** **Project writeup due**	

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

* **31 March**: Deadline to withdraw from a class (5pm)

Calendar (structural, narrative)

Week 1

During Week 1 we introduce the course and the idea of an algorithm, and make a particular point to discuss how reading a textbook and studying CS, math, or other technical course is not the same as reading a news article or studying for other classes. We also introduce our classroom model for communicating about programs that we develop in-class. (CCO 6, 7; FO 14)

Week 2

Here we write our first non-trivial programs (where we're not just typing in someone else's code). (SLO 1)

Week 3

In the third week, students begin writing programs using if/else conditional constructs and choosing when to use them to model a problem. We introduce word problems into the mix, which motivates discussion and work on how to formulate the problem computationally or algorithmically. (SLO 1, 2; QRO 9, 12)

Week 4

We continue practicing how to express simple computational algorithms, adding repetition (loop) constructs to our toolbox. (SLO 1, 2; QRO 9)

Week 5

In the fifth week, students expand their loop skills and we introduce lists. (SLO 2)

Weeks 6–7

We now begin to write and design loop-based programs to process sequential data, choosing from among several standard algorithms the one(s) most appropriate to the task. We also formalise the concept of the "test case" is

formalized as a way to express, in advance, reasonable and expected outcomes from the solution of a problem that has been modelled computationally. The website CodingBat is introduced as a way to practice and study outside the immediately supervised context of the classroom and conventional homework. (SLO 2, 4, 5; CCO 7; QRO 10, 11)

Week 8

Several strands of programming are deepened; for the first time, we write programs that can read their data from a text file. (SLO 1, 2, 3; CCO 6; QRO 9)

Week 9

Both flat and structured text files are explored. I will explain what is expected during the upcoming "elevator speeches" to be occurring in the following week. (SLO 1, 2, 3, 4, 5; QRO 9; FO 14)

Week 10

We revisit the essential string data type and the many ways to manipulate them. (SLO 1)

Week 11

Students develop and present brief speeches about the problem they have chosen for their course project, for which I will send them individualized feedback. (CCO 6, 8; QRO 9; FO 13, 14, 15)

Week 12

We continue to explore applications of the earlier concepts; and we move on from sequential (list) data to more complex data structures. (SLO 1, 2; CCO 6; QRO 9, 10)

Week 13

Image manipulation forms a case study for the week, to explore an application and apply earlier concepts. Motivated students will be able to use this as a foundation for independent or course-driven future problem-solving using algorithmic means. (SLO 2; CCO 6; QRO 10)

Week 14

This week is dedicated to ongoing inquiry into the problem they have chosen for their course project. (SLO 1, 4, 5; CCO 6; QRO 9, 10, 11)

Week 15

In the final week of the class, the focus is on finishing and presenting the interpretation and analysis of the results they have gotten on the project. Feedback after the first presentation of the week will help them prepare and polish their second presentation. (SLO 4, 5; CCO 7, 8; QRO 11, 12; FO 13)

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 website[1] 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 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.

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