

Course Title: CS1210: Computer Science I: Fundamentals

...come to class...do the work...ask for help...

Term: Fall 2024

Time: 8:30am - 9:20am MWF

Location: 100 Phillips Hall

Department: Computer Science

Instructor: Alberto Maria Segre

14G MacLean Hall

alberto-segre@uiowa.edu

Office Hours: 9:30-11:00 Fridays (or by appointment)

Course Description

Welcome to CS1210! This is the introduction to computer science course for CS majors and minors, Data Science majors, CS&E majors, and students from other disciplines (*e.g.*, mathematics, statistics) where a solid foundation in computer science combined with fluency in computer programming is required.

CS1210 presents computational concepts through working examples and exercises. Topics include data representation and manipulation, control strategies, functions, objects and classes. And while the course will be taught in Python (more specifically, Python 3), the emphasis is on learning general problem solving techniques, applying these to new problems, and analyzing the quality and correctness of the solution rather than simply learning to program in some specific language. In short: computer science is not the same as computer programming!

All students are expected to have competency with basic arithmetic and algebra: any student ready to take Calculus I is ready to take CS1210. Although there are no explicit computing prerequisites for this class and no previous programming experience is required, we assume you have strong basic computing skills (*e.g.*, understanding files and folders, difference between file types, simple network concepts, using email, attachments, web basics, understanding URLs, etc.) is assumed.

That said, I do realize that students enrolled in this class may likely come from a broad range of backgrounds, from those who have never seen a program before, to self-taught programmers, to those who have already had some exposure to programming in high school, to those who consider themselves already proficient programmers with several years of experience (albeit perhaps not in Python). Surprisingly, experience shows that prior programming experience does not necessarily correlate with a higher grade at end of term! Yes, your classmates may seem to have more programming experience than you, may believe they already know exactly how to approach a specific problem, but remember: (*i*) this class is about more than programming, and (*ii*) prior programming experience doesn't always result in well-conceived and well-executed programmatic solutions.

On the other hand, CS1210 does cover a lot of material at an aggressive pace: if you feel that CS1210 may be too overwhelming, alternative introductory computing courses are CS1020 *Principles of Computing* (for all students) and CS1110 *Introduction to Computer Science* (for pre-CS, Informatics, social sciences, and other data-oriented majors). Both of these other courses, like CS1210, satisfy the quantitative reasoning general education requirement. CS1020 generally has little or no programming content, while CS1110 includes a gentle introduction to Python, and is therefore also suitable for those exploring the CS major but who may wish to have more exposure to computing before attempting CS1210 (in other words, "pre-CS" majors).

Course Format

The course is presented in lecture/discussion form. The lecture section meets from 8:30am to 9:20am on Monday, Wednesday and Friday in the 100 Phillips Hall, right at the NE corner of where Iowa Avenue meets Clinton Street, directly in front of the Old Capitol building.

Each lecture is recorded during delivery and later posted to the ICON site. These recordings, which capture the screen and my voice, are meant to allow you to go back and watch specific parts of a lecture in order to help clear up any questions you might have. The ICON site provides an index for each lecture so you can focus in on the specific part you care about, and you can actually choose to watch/listen at higher or lower speeds. Think of these recordings primarily as a fail safe mechanism in the event something in class needs clarification, or if you are sick and cannot attend.¹

In addition to the lecture, all students will also attend a smaller 75 minute discussion/lab section every Tuesday starting the second week of classes (*i.e.*, watch for an announcement in class). Each lab/discussion sessions is led by a TA and meets in small (less than 25 seat) computer classrooms.

A01	8:00am - 9:15am T	Charles Tobler
	301 MLH	<i>charles-tobler@uiowa.edu</i>
A02	9:30am - 10:45am T	Amy Gilhoi
	301 MLH	<i>amy-gilhoi@uiowa.edu</i>
A03	11:00am - 12:15pm T	Amy Gilhoi
	301 MLH	<i>amy-gilhoi@uiowa.edu</i>
A04	12:30pm - 1:45pm T	Charles Tobler
	301 MLH	<i>charles-tobler@uiowa.edu</i>
A05	2:00pm - 3:15pm T	Amy Gilhoi
	301 MLH	<i>amy-gilhoi@uiowa.edu</i>
A06	3:30pm - 4:45pm T	Alexander Guo
	301 MLH	<i>alexander-guo@uiowa.edu</i>
A07	5:00pm - 6:15pm T	Alexander Guo
	301 MLH	<i>alexander-guo@uiowa.edu</i>
A08	6:30pm - 7:45pm T	Alexander Guo
	301 MLH	<i>alexander-guo@uiowa.edu</i>
A10	9:30am - 10:45am T	Charles Tobler
	W240 AJB	<i>charles-tobler@uiowa.edu</i>

On arrival each Tuesday, you will be randomly paired with another classmate (you may occasionally be paired with two classmates in the event your section has an odd number of students) to complete a small programming assignment, generally consisting of a few function definitions. You and your partner(s) will upload a single (joint) solution for grading, and identical grades will be assigned to each partner. Note that *only* work done in person during your assigned lab section with your assigned partner(s) will be graded. The lab problems exercise important problem solving, programming, and collaboration skills; working with a partner encourages discussion and reinforces your own understanding of the concepts (and working collaboratively is an essential skill in this discipline). As a bonus, if you and your partner get stuck, the

¹ One cautionary note: because the lectures are recorded and posted, some students assume that they are optional: **this is not the case**. Skipping a lecture and promising yourself that you will watch it later is a provably bad strategy, because (i) recorded lectures are always less engaging than interactive lectures, (ii) you can't ask questions of a video recording, and (iii) experience confirms that students who elect not to attend obtain systematically lower grades and are far more likely to fail or withdraw from the class.

TA will be there to help you successfully move ahead.

Course Materials

Your primary reference for all course-related material is the ICON website. You are responsible for the material posted there, and you should visit it regularly — at least once a day is not an unreasonable ask. On the website, you will find course announcements, a copy of this syllabus, video recordings of the lectures, references to all of the homework assignments, the class calendar (including reading assignments), and a link to the online class discussion board, "Piazza" (more on this later).

There are many reasonable textbooks to help beginning programmers with Python and the other concepts introduced in this course. Because books can be expensive, over the years I've learned to prefer free, open source, textbooks. The book I have selected and from which I will assign most readings is *Python for Everybody* (denoted [P4E]), by Charles Severance from the University of Michigan, which is available here (in multiple languages):

<https://www.py4e.com/book>

Note that the book's website has a bunch of additional material which you may find useful, including lectures, extra exercises, and so on. More to the point, there are literally tons of resources on the web to help with this course. Indeed, even the book we use is itself an open-source remix of the popular open-source text *Think Python: How to Think like a Computer Scientist* (denoted [TP2]), by Alan Downey from Olin College. The 2015 2nd edition is available in PDF here (in multiple languages):

<https://greenteapress.com/wp/think-python-2e/>.

Note that, on occasion, I may assign readings from this second text as well, so be sure to check whether. For your convenience, PDFs of both text books are available directly from the the CS1210 ICON website.

One more: just last month, Alan Downey released the 3rd edition of Think Python (denoted [TP3]):

<https://greenteapress.com/wp/think-python-3rd-edition/>

Unfortunately, there is no free PDF for the 3rd edition (you can buy a bound paper copy on Amazon if you must), but there is a really cool online interactive version of the book:

<https://alldowney.github.io/ThinkPython/>

that lets you interact with the material and provides many online problems to help you better understand the material. When I assign a reading from [TP3], I'm suggesting you use the interactive edition. Besides, it has a nifty AI assistant which can answer your questions as you go.

You should plan to complete any assigned readings as soon as they are assigned, but most will also find rereading them once the concepts have been presented to you to be useful as well. I think you will find that each time you revisit the text you will achieve greater clarity and understanding of what is, frankly, complicated material. Finally, note that I have purposefully chosen free texts, all of which are available at no cost to you, because I am sensitive to the high cost of textbooks. But in return, I ask that you recognize that just because a book is free does not mean that it is of no value! When I assign a reading, it's because that reading is valuable (and note that you are responsible for its content on any subsequent exam).

The programming language used in this class is Python, and more specifically, Python 3 (the 3rd edition of the language). We expect that you will want to download and install a version of this language to your own laptop; we'll explain what to download and how to install Python sometime during the first week or

two of class.² Most students have been able to follow these instructions without any issue, and the TAs are happy to help you if you get stuck. Note, however, we cannot provide technical support for any installation on your personal equipment (some people's machines are highly non-standard, or may have issues that foil installation of new software).

For those who want a head start on this process, I recommend the current distribution (Python 3.12) from Anaconda:

<https://www.anaconda.com/download>

After providing your email, you can just click on "download" as the page should recognize your machine's operating system (Windows, Mac³ or Linux). Anaconda pretty much runs on every platform, and so is rapidly becoming the standard Python distribution for almost everyone.

In the unlikely event you have an older machine with limited disk capacity (or you have too many movies on your hard disk!), you may instead elect to use a diet-sized Anaconda distribution called Miniconda, available here:

<https://docs.anaconda.com/miniconda/>

As with Anaconda, please be sure to choose version 3.12 appropriate for your hardware, and not one of the version 2 options. Miniconda will be fine for much of the term, but as homework problems become more complex, you may find you will eventually need to switch to the full Anaconda edition.

Teaching Assistants, Office Hours, and Supplemental Instruction

Speaking of TAs, we have three excellent graduate students assigned to this class. Each TA supervises three lab sections, where they serve as interactive resources as you work through that week's problem set with your lab partner(s). Please make note of your TA's name as well as your lab section number. Note that lab sections are not "open seating:" slots are limited and determined by registration, and your lab assignment will only be available during your assigned section. You *must* attend the lab section in which you are officially enrolled.

In addition to leading lab sections, each TA provides a certain number of office hours (*i.e.*, "help sessions") each week. Unlike the labs, the help sections are "open seating:" they are open to all CS1210 students, and not just those enrolled in that particular TA's own lab sections. If you need help with class concepts or a homework assignment, your first step should be to attend one of the many TA help sessions. All of the TAs are equally able to help you and you should feel comfortable asking any TA for assistance. TA — as well as my own — office hour times and are posted on the ICON website as well as in the table below:

Alberto Segre	9:30 AM - 11:00 AM Friday
14G MLH	(else by appointment)

² The expectation is that you have access to a personal computer, meaning a desktop or laptop that runs Windows, Mac OS, or Linux: most Chromebooks won't suffice to run Anaconda/Python; those that might work require a reasonably sophisticated skill set to install successfully.

³ Mac users, some of you may be surprised to learn that you likely already have Python installed on your machines. Unfortunately, depending on your machine, it may well be Python 2, so you will still have to download and install version 3.12. Python 2 and Python 3 differ in several ways, some easy to avoid some less so. You must be sure you are using the right platform, or what you see in class is not what you will see at home.

Amy Gilhoi	10:00 AM - 12:00 PM Thursdays
314 MLH	12:30 PM - 2:30 PM Fridays
Alex Guo	3:00 PM - 5:00 PM Mondays
314 MLH	3:00 - 5:00 PM Wednesdays
CJ Tobler	6:30 PM - 8:30 PM Mondays
314 MLH	6:30 PM - 8:30 PM Wednesdays

These may be subject to change over the course of the semester: any rescheduling will be noted in lecture and announced on the ICON course homepage.

This semester we are also fortunate to have access to supplementary sessions offered by Nolan, a CS student and supplemental instruction leader: it wasn't so long ago that Nolan was walking in your shoes! I think you will find attending these communal problem solving sessions is an excellent way to gain insight from someone who is close to the student experience and was very successful in this class. It is also an excellent opportunity to meet your peers and classmates. Supplemental instruction sessions will be held three times a week in the Academic Resource Center (lower level of the Iowa Memorial Union) on Mondays 12:30pm-1:20pm, Tuesdays 10:00am-10:50pm, and Thursdays 7:00pm-7:50pm starting on Tuesday, September 3, through Friday, December 13, except for the Thanksgiving holiday (Saturday, November 23 through Sunday, December 1). See ICON for details.

Piazza: An Interactive Resource

The class ICON page also has an interactive discussion board ("Piazza," the Italian word for a public gathering space) where you can post your own questions, read questions asked by your classmates, make suggestions, read instructor comments, and have follow up discussions with the instructors and your classmates. *Piazza is the single most important resource for CS1210 students! Why?*

- (1) First, it is always available. The TAs and I monitor the board throughout the day, and questions are answered as promptly as humanly possible (last Fall, 648 questions generated 721 instructor responses and 510 student and instructor follow up comments, with an average response time of...43 minutes).
- (2) Second, you are free to post your questions anonymously: although I will still know who you are, this ensures that no one will think less of you because of the questions you ask. No need to feel shy about asking the most basic thing. I can assure you that if you have a question, there is someone else out there who is secretly hoping you will ask it. So feel free to make your posts "anonymous."
- (3) Third, you can also post your question "privately" so that only instructors (the TAs and myself) can see it. This is appropriate when you are asking question about a homework problem you are currently working on; as a basic rule, whenever you post your code, make it "private." The TAs and I have the option to make it "public and anonymous" if we think its useful and appropriate, so win-win.
- (4) Finally, Piazza provides you and your classmates a chance to offer your own advice and insights by participating in the discussion. This is the most valuable aspect of Piazza: it really is an open, moderated, forum for students to discuss and interact outside the classroom (obviously, please keep your posts on topic and within the usual norms of civility; inappropriate posts will be deleted).

Grading

My grading philosophy has changed over the years, informed by practical considerations that are exacerbated by technological advances and ever larger enrollments. In short, most of your grade will be based on your performance on the two midterms (20% each) and the final exam (25%). These exams are closed book, closed note, strictly written exams: you will not have access to a computer, so doing well on these depends on knowing your stuff! And because mastering the course material requires lots of practice, we provide you with lots of homework opportunities. To ensure you get consistent and timely feedback, we make use of an *autograder* (basically a program that evaluates your program, sometimes in surprisingly sophisticated ways) to return most of your work within 48 hours. Unfortunately, autograders aren't that good at assigning partial credit or in judging just how "close" your submission is to a correct answer, so much of your grade will in turn be based on your exam scores. I shouldn't have to tell you that how much you practice directly affects how well you are likely to answer questions on an exam!

Your final grade will be computed based on five distinct elements:

Question of the Day (QotD: 5%): There will be 20+ practice questions posted mostly over the first two thirds of the course. Once posted, a QotD is due just before the next lecture, so you will generally have no more than 24-36 hours to do so. QotDs are to be completed individually in your time; solutions must be submitted online, and only the most recent version submitted prior to the deadline will be graded.

Important: late QotDs will not be accepted, nor will extensions be granted under any circumstances: moreover, as QotDs are graded automatically, no partial credit will be given for incorrect solutions. Instead, your final QotD grade is reported as the percentage of correct QotDs after dropping the lowest 25% of your QotD scores to allow for unavoidable absences or illness. So, if there are 28 QotDs and you answer 19 correctly, your QotD score would be $19/(28-7) = 90\%$ after dropping the lowest 7 scores.

Lab assignments (Labs: 14%): There will be 10-12 laboratory assignments given over the course of the term. Each week you will be randomly assigned a partner to work with for each assignment: lab assignments must be completed with your assigned partner during your assigned discussion meeting time and submitted online by the end of the period. In weeks where there are no lab assignments, discussion sections will be devoted to reviewing material and answering questions.

Important: late laboratory assignments will not be accepted, nor will extensions be granted under any circumstances: moreover, as laboratory assignments are graded automatically, no partial credit will be given for incorrect solutions to laboratory assignment questions (note, however, that each lab may contain multiple questions). Instead, your final laboratory grade is computed after dropping your three lowest laboratory assignment scores to allow for unavoidable absences or illness. So, if there are 13 laboratory assignments, we drop your 3 lowest scores and report your laboratory assignment score based on the remaining 10 assignments.

Homework assignments (HWs: 15%): There will be 2 (or possibly 3 smaller) multi-week homework assignments consisting of larger programming projects over the course of the semester. You can expect to have just under 2 weeks for each assignment; these are to be completed individually (no partners allowed); your solutions will be accepted via the ICON dropbox, and only the most recent version submitted prior to the deadline will be graded.

Important: homework assignments will be due Friday at midnight. Late assignments will be accepted until Sunday at midnight, but grades on late assignments will be capped at 75% of the original credit. No extensions beyond this weekend extension will be granted under any circumstances. Because these assignments are graded individually, partial solutions are eligible for partial credit, although they will not be graded as promptly as QotDs or Lab

Assignments.

Midterm exams (Ex: 18% each): There will be two evening midterm exams, each worth 16% of your final grade, which will be held on September 23 (6:30-8:00pm in MH AUD) and October 28 (6:30-8:00pm in MH AUD).

Important: Students with university-approved conflicts or SDS extra-time accommodations should use ICON's Proctored Exams Portal to set up your conflict exam time.

Final exam (Fin: 30%): The final exam will be held on Friday, December 20, from 10am-noon in W10 PBB (the basement of the B-School building). will be worth 28% of your grade. As for the midterm exams, we reserve the right to schedule follow up oral exams as needed to better ascertain how well you comprehend the material.

Important: Students with university-approved conflicts or SDS extra-time accommodations should use ICON's Proctored Exams Portal to set up your conflict exam time.

This course uses letter grades. Because CLAS discourages the use of norms-referenced grading (the "curve"), we will adopt a standard (A:90-100, B:80-89.99, C:70-79.99, D:60-69.99, F:0-59.99) grading scale. Plus/minus grades will be awarded sparingly at the instructor's discretion to recognize those with exceptional performance or who are unnervingly close to the cusp. Note that, because it is sometimes hard to predict class performance on challenging material, the standard CLAS scale may result in unreasonably low letter grades. We will therefore have three "true-ups" over the course of the term: once after each midterm, and a final "true-up" at end of term. A "true-up" is an opportunity for me to lower (never raise) the cutoffs based on the grade distribution so far: thus each "true-up" can only improve your letter grade, never make it worse.

All grades will be posted on ICON as soon as they are available, but please realize that grading 200+ exams and/or programming assignments is a difficult and time-consuming task. We would rather do the job right the first time than do a poor job and rush to compute your scores. On the other hand, we are committed to grading QotDs and laboratory assignments within 48 hours if at all possible. Finally, please note that grade related questions should be directed to your TA, and then, if unresolved, to me directly during office hours.

Workload and Expectations

Students are responsible for all material presented in lecture and discussion section, as well as additional readings or content posted on the ICON web site, where course announcements, corrections and clarifications to the assignments are also posted. One easy way to keep up is to subscribe to the class calendar on ICON using whatever email/calendaring solution you prefer. For example, because I am an Android user, by exporting the class calendar link from ICON and importing it into Google Calendar, I have all class sessions and assignments on my phone's calendar.

Mastery of the course material entails a significant commitment of time and effort on the part of each student. Each semester credit hour, by definition, assumes a minimum of 50 minutes of instructor-delivered content and at least 2 additional hours of independent work outside of the classroom per week. Expect to spend 200 minutes in the classroom (including the laboratory sections) and at least 8 hours outside of the classroom (that's on average, that is, sometimes a little more and sometimes a little less) every week studying for CS1210 alone.

How to Succeed in CS1210

This is a challenging course that covers a lot of material. My hope is that every student will succeed in this class, and will come to enjoy the material as much as I do. Pragmatically speaking, however, I realize

that many students find this course to be quite challenging. Nevertheless, there are a number of things you can do to improve your performance, starting with the class motto at the top of this document.

First, **come to class**. Although past offerings of this course strongly suggest that attendance is correlated with success, some students inevitably feel that class is optional, and that they can simply read the book and learn the material on their own. Experience reveals that this feeling is often more myth than reality, and that students who do not attend lectures and labs fail at a much higher rate than those who attend. And even if they pass, their grades are on average much lower than those of students who attend. That said, if missing a class is unavoidable, I do upload each lecture's video capture to the ICON web site: however, these videos are intended to help you review concepts presented in class, and they are no substitute for attending the lecture itself.

Second, **do the work**. QotD's are designed to make you practice a little bit every day. Laboratory assignments give you an opportunity to work collaboratively and interactively while you develop your skills. Homework problems and exams are designed to let you demonstrate what you have learned. Take full advantage of each of these opportunities. In particular, start working on your homework problems right away: in this class, procrastination is the best predictor of failure! If you wait until the last minute, you are simply unlikely to successfully complete the homework. Read each homework assignment carefully as soon as it is available so that you are clear what the assignment is asking you to do; come to lecture, discussion, help sessions or office hours prepared to ask questions about the assignment if something is unclear.

Third, **ask for help**. There is no substitute for asking questions in class! Moreover, no question is a stupid question; I can guarantee that any question you might have will also be on the mind of at least one other classmate. Asking questions in class is the quickest, most reliable, most efficient, means of getting an answer, so don't be afraid to raise your hand (I don't bite). In a similar fashion, take full advantage of TA and instructor office hours, TA-led discussion and help sessions, and on Piazza, the online class discussion board.

Fourth, this class **takes time**. As a general rule, each student should expect to devote 2 hours of additional work per week for each course credit hour; since this is a 4 credit course, you should expect to spend, on average, about 8 additional hours per week on readings, homeworks, programming projects, etc. Realize that this estimate is an *average* (e.g., some weeks will require more work than others) and excludes any hours you might spend trying to assimilate material from lectures you did not attend or working problems covered in discussion sections that you missed or did not complete. Moreover, experience tells us that several shorter programming sessions are generally more effective than a single long programming session of like duration, so don't expect to start and finish a homework assignment in one sitting (unless it's a QotD or a Lab, which are mostly designed for one sitting).

Finally, **have fun!** Realize you are learning difficult material and trying to apply it in interesting ways. Give yourself a break! Don't obsess over your grades, focus on learning how to get the machine to do your bidding, and enjoy the rush when your stuff works. We're all here to help you get to that point; if you supply the time, patience, and labor, I think you will find this class to be very rewarding.

Communication: UI Email

Students are responsible for all official correspondence sent to their UI email address (*first-last@uiowa.edu*, or *hawkid@iowa.uiowa.edu*). Because we may on occasion opt to send email announcements to all students in the class, make sure to check your UI email regularly. Moreover, for privacy concerns and to ensure the protection of student records, UI faculty and staff can only correspond with UI email addresses (see top of page for my email address and the table of lab sections for the email addresses of the course TAs).

If you are writing me with questions about course content, please consider attending one of the TA help sessions or using Piazza instead. This is because I get literally hundreds of emails a day, so go easy on me: be specific and to the point with your question(s) and make sure your identity is clear (posting from your UI address makes your email more likely to get through but also serves to identify you as a student). Be sure to the subject line of your message includes the label **CS1210**, so that we can identify high priority class-related items in our inboxes. Just to be concrete, here is a "good message" that will get as prompt a response as possible:

From: Mary Student (mary-student@uiowa.edu)
Subject: CS:1210 help with return values

This is Mary from CS:1210 Section A02. In class today, you said that functions can return more than one value, but that seems counterintuitive. Can you give another example?

and here is an example of a message that would likely not get a response (unclear question, no idea who the student actually is, a topic perhaps best addressed during TA office hours, etc.):

From: Da bomb (puppiesarefuzzy@hotmail.com)
Subject: hey dude!

I got zero points on my last lab but my code works perfectly.
What the heck?

Academic Integrity

All students in CLAS courses are expected to abide by the CLAS standards of academic honesty:

<https://clas.uiowa.edu/academics/handbook/standards/academic-honesty>

Undergraduate academic misconduct must be reported by instructors to CLAS according to these procedures. Graduate academic misconduct must be reported to the Graduate College according to Section F of the Graduate College Manual.

In CS1210, our policy is that violations of academic integrity will not be tolerated. The primary rule is that you should never represent work done by someone else (including any code from the Internet or the product of an AI-based programming assistant) as your own, nor should you share your own work with others. We routinely employ software to detect plagiarism and to test any suspicions we might have. If you are unclear about what constitutes academic dishonesty, ask the instructor.

Obviously, there are shades of gray here. We very much do want students to talk to each other about concepts and ideas that relate to the class; this is one of the reasons we pair you with another student in the laboratory sessions, and sharing lab work with your partner is explicitly allowed. But other assignments, such as homework problems and QotDs, are meant to be solved individually, not with help from a partner. And while it is OK to discuss general homework-related questions, concepts, and ideas, it is important to ensure that these discussions do not lead to the actual exchange of solutions or code fragments: in these circumstances, sharing written, printed or electronic copies of your programs or pieces thereof with anyone but the instructor or the TA is simply not OK.

What about AI tools? Some claim that AI-based (more precisely, LLM-based) programming assistants are the future of coding. We will certainly spend some time over the course of the semester discussing the use of LLMs ("large language models") to write (and explain) Python code. Used properly, LLMs can serve as useful tutors, as long as you keep in mind that they can sometimes give wrong or incorrect code/explanations. But like human tutors, they're meant to support learning, not do your homework for

you. Thus the rule for LLMs is that you are free to use them, but only when explicitly permitted, and you should never, ever, represent solutions or partial solutions generated by an LLM as your own work.

Realize that ultimately your grade in CS1210 must reflect your mastery of the material: your ability to construct good, well-formed, efficient computational solutions to problems. You might think Googling answers or asking ChatGPT will help you solve your QotD, but these shortcuts simply rob you of the opportunity to learn by doing your own work, which ultimately contributes to mastering the material. This is why two-thirds of your grade depends on closed-book written exams. And, yes, this means writing code on paper, without help or support from a smart editor or web search. Experience confirms that, under these circumstances, those who have actually mastered the material will perform much better than those who took shortcuts.

That said, the Piazza resource provides a safe space to support all of these interactions. Questions, discussions, and interactions regarding any aspect of CS1210 are almost always OK on Piazza, as long as solutions or partial solutions are not posted or exchanged. It is, for example, legitimate to ask why a print statement works a certain way, as long as your question is about Python usage and not about a particular homework problem. It is also perfectly OK to post a snippet of Python unrelated to the assignment in order to make the thrust of your question clear. Even posting code that is part of a solution you are working on can be OK if it is marked "private" at submission time (only instructors and the original poster can see private content). Note that if a posting is private but could legitimately be public, it will be up to the instructor to decide and make it "public and anonymous." So when in doubt, post privately.

On the Early Introduction of Formal Methods into the CS Curriculum

This offering of CS1210 is a part of a three-year research study to improve student success in the Computer Science curriculum. We are interested in whether the early integration of formal mathematical reasoning in classes like CS1210 can improve student understanding and overall performance.

The study is funded by the National Science Foundation and is co-directed by Computer Science Professor Garrett Morris and myself. Your participation, which is optional, will be limited to filling out three online surveys over the course of the semester. Your responses will be collected anonymously: completing the surveys will not affect your standing or grade in the class. Professor Garrett will make a brief presentation and answer your questions about the study during the second week of classes.

Student Complaints

Students with a complaint about a grade or a related matter should first discuss the situation with the instructor, and, because in this case the instructor is also the Chair of the Department, to the Department's Director of Undergraduate Studies, Professor Liza Kleiman.

Undergraduate students may elect to contact CLAS Undergraduate Programs:

<https://clas.uiowa.edu/students/handbook/student-rights-responsibilities#rights>

for support when the matter is not resolved at the previous level.

Drop Deadline for this Course

You may drop this course before Monday, September 9, 2024; after this deadline you will need collegiate approval. When you drop a course, a "W" will appear on your transcript. The mark of "W" is a neutral mark that does not affect your GPA. Directions for adding or dropping a course and other registration changes can be found on the Registrar's website:

<https://registrar.uiowa.edu/change-registration-myui>

Undergraduate students can find CLAS-specific registration policies in the CLAS academic handbook

here:

<https://clas.uiowa.edu/academics/handbook/registration>

and, more specifically, policies regulating dropping a class here:

<https://clas.uiowa.edu/academics/handbook/registration#dropping-courses>

Additional Course Policies

The Provost's Office maintains a website containing additional University-mandated course policies here:

<https://provost.uiowa.edu/student-course-policies>

including:

- Free Speech and Expression;
- Non-discrimination Statement;
- Accommodations for Students with Disabilities;
- Absences from Class;
- Absences for Religious Holy Days;
- Absences for Military Service Obligations;
- Classroom Expectations;
- Sexual Harassment/Sexual Misconduct and Supportive Measures;
- Conflict Resolution;
- Mental Health;
- Basic Needs and Student Support; and
- Sharing of Class Recordings.

These policies are incorporated here by reference.

Syllabus Modifications

This syllabus is subject to change. Changes will be announced on, and posted to, ICON.