



CS1210 A (majors) / Introduction to Programming / 2024 Fall

Compiled: 2025-05-16 15:58

Class meetings: M/W 3:30–4:45 PM, Votey 303

Instructor: Clayton Cafiero

Email: cbcafier@uvm.edu

Office: Innovation E309

Office hours: Drop in M 2:00–3:00 PM, Th 8:30–10:00 AM, F 12:00–1:30 PM, or by appointment

Website: <https://www.uvm.edu/~cbcafier>

GTA: Mohsen Ghasemizade

Email: mghasemi@uvm.edu

Office: Innovation E446

Office hours: M 10:00 AM–12:00 PM, or by appointment

UTA help desk: S–Th 5:00–7:00 PM

Location: Lafayette L311

Supplemental UTA office hours: S 1:00–3:30 PM

UTA: Alex Schaefer

Location: Via MS Teams

Welcome to the course!

Description: Introduction to Programming teaches problem solving, and the problem solving process, using Python. Topics include types, variables, functions, branching, loops, flow of control, simple input/output, and basic data structures. Throughout the course, good programming style will be emphasized. The course assumes no prior programming experience. Satisfies Quantitative Reasoning (QR) and Quantitative and Data Literacy (QD).

Learning objectives: Throughout this course, you will

- learn the fundamentals of computer programming,
- begin writing idiomatic Python,
- develop critical thinking and problem-solving skills,
- think creatively to solve problems,
- express solutions clearly and accurately,
- learn to implement non-trivial programs in a high-level programming language, and
- develop programming maturity (confidence, willingness to experiment).

Textbook: Cafiero, C. (2023) An Introduction to Programming and Computer Science with Python. ISBN: 979-8-9887092-0-6. Textbook is required but is made available to you in multiple formats:

- Hard copy is available through the UVM Bookstore for \$14.00 (USD).
- Free PDF is available via my website or through link in Brightspace.
- Chapter content is posted to Brightspace (this does not include frontmatter, index, and certain other content).

Flipped classroom: This course is taught as a “flipped” class—that is, it incorporates some lecture and extensive “active learning” exercises. The objective is to provide you with more hands-on experience and practical guidance as you learn how to solve problems through computer programming. This means that you are expected to complete direct instruction—typically in the form of readings and videos—*prior* to coming to class. In our first meeting each week, there will be “mini-lectures”, question and answer sessions, and review. Then we will engage in hands-on, active learning exercises, with instructor and TA support.

Each week’s instructional material will include readings and videos posted on Brightspace.

Computer: For this course, you should have a reliable computer on which you can write, run, and debug code. Windows ≥ 10 and macOS ≥ 14.0 are supported. If you have a Linux machine, that’s fine, but you’re on your own for support.

Software: You will be writing programs in Python, and accordingly you should have a reasonably current version of Python installed on your computer (e.g., version $\geq 3.10.0$). Python source code is in plain-text format, and you may write and edit Python code with any plain-text editor (e.g., Sublime or similar). However, it’s helpful to use an integrated development environment (IDE). An IDE provides additional functionality not available with a plain-text editor (syntax highlighting, debugging, etc.). Recommended:

- **Thonny**, Python IDE for beginners: <https://thonny.org/>

Thonny includes a built-in package manager that will make things a little easier for you when we get to using Matplotlib. Installation instructions are available on Brightspace.

Python supplies its own IDE, called IDLE. You may use this if you wish. You may wish to try JetBrains PyCharm or Microsoft Visual Studio Code. These are more feature-rich IDEs, but can be overwhelming to new users. You’re welcome to try either, but as far as support goes, you’re on your own. PyCharm has a free “community” edition for educational purposes. See: <https://www.jetbrains.com/pycharm/download/> or <https://code.visualstudio.com/>. If you’re in an engineering or a scientific discipline you may wish to try Spyder IDE. See: <https://www.spyder-ide.org/>

Prior knowledge: This course assumes no prior knowledge of computer programming. However, we do assume you have working knowledge of the basics of arithmetic and elementary algebra. A high-school course in algebra should suffice. You should also know how to use your computer’s operating system, be able to navigate your computer’s file system, and be able to open, close, rename, and move files and directories (folders).

Important websites:

- Brightspace, for course materials and announcements: <https://brightspace.uvm.edu>.
- Gradescope, for submitting assignments and seeing grading feedback: <https://www.gradescope.com/>.
- Official Python Documentation: <https://docs.python.org/3/>. This is the definitive resource for Python documentation.

Correspondence: Please use email for electronic correspondence (and not MS Teams). As I teach multiple courses please indicate the course in which you are enrolled in the subject line. Please use your UVM email for all correspondence.

Weekly schedule of topics
(tentative and subject to change):

Week	Topic	Readings (ch/s)
1	Introductions; types and literals; variables, operators, and expressions	1–3, 4.1–4.3
2	Functions; modular arithmetic; comments and style	4.4–4.7, 5.1, 6
3	Console I/O; f-strings; more on functions; imports and the math module	5.2–5.7, 7
4	Branching, flow control, and Boolean logic; program structure	8–9
5	Sequences: lists, tuples, and strings; mutability and immutability; (mini) exam	10
6	Loops and iteration	11
7	The random module; games and simulations	12
8	Review and second exam	
9	File I/O; the csv module	13
10	pip and virtual environments; the statistics module; introduction to Matplotlib	14
11	Exception handling; brief intro to dictionaries	15, 16.1–16.2
12	Dictionaries; introduction to graphs	16.3–16.6, 17
13	Review and third exam	
14	Work on final project; additional topics TBD	

Grading:

Weight	Assessment
10.0%	in-class quizzes (iClicker; 10)
5.0%	online quizzes (Brightspace; 5)
10.0%	lab / pair programming (11, drop 1)
25.0%	programming homework (12, drop 2)
4.0%	brief metacognition essays (2)
30.0%	exams (3; 5/10/15%)
16.0%	final project
100.0%	TOTAL

Assignment of letter grades will be on a conventional scale. Any grade appeal (assignment, quiz, lab, exam, *etc.*) must be directed to your grader within one week of the grade being posted. If grading is done on Gradescope (*e.g.*, for homework), there's a *regrade request* feature which should be used for grade appeals.

Final exam: There is no final exam in this course. The Office of the Registrar schedules final exam dates and times regardless. You may ignore whatever the Registrar has scheduled.

Programming homework and labs: You will have many opportunities to practice concepts and language features presented in class, readings, and videos, primarily through programming homeworks and labs.

Programming homework may be done individually or in teams of two (absolutely no more than two). Programming homework is submitted via Gradescope. If you work with someone on homework you must make a note of who you collaborated with in the docstring of each file submitted. All students must type up and submit their own work, regardless of collaboration.

Labs are pair programming exercises. Occasionally, if we have an odd number of students in the room, I will allow one person to work alone, or have one team of three, at my sole discretion. A portion of each lab will be submitted via Gradescope. The remaining portions are to be completed on your own in a team.

It sometimes happens that one programming partner has more experience or grasps certain concepts more quickly than the other. This can lead to an “unbalanced” condition, where one team member does more of the work or accomplishes more objectives than the other. Obviously, this is undesirable, as it does not give the other team member sufficient opportunity to engage with course materials, learn, and grow. Accordingly, I reserve the right to reassign partners or prohibit certain pairings for homework or lab. If this occurs, I will be discreet, and please trust that such decisions are in the interests of all parties involved.

Quizzes: In most weeks there will be a brief quiz administered in class using iClicker. Please install iClicker student mobile app on your smartphone. Sign-on will be through UVM’s institutional login using your UVM NetID and password. Join link for this course: <https://join.iclicker.com/ZQIQ>. More details will be presented in class.

In weeks 3, 5, 7, 10, and 12, there will be a brief, online quiz administered via Brightspace.

Metacognition essays: Over the course of the semester you will write two *very brief* metacognition essays. See additional instructions, guidelines, and prompts posted on Brightspace.

Final projects: There is no final exam in this course. Instead there is a final project. Final projects are to be completed in teams of two. Occasionally, someone with very specific interests requests to do a solo project. Such requests are evaluated on a case-by-case basis.

Final project proposals are due in late October (1% of your grade). Final projects are due on the last Thursday of classes (5 December). Late projects will *not* be accepted. Final projects account for 15% of your grade. See additional instructions, guidelines, and suggestions posted on Brightspace.

TA help desk etiquette: The TA help desk is intended for questions and limited support in programming. Do not expect TAs to write or debug your code for you. Come prepared with specific questions. Do not monopolize TA time. Be aware that TAs are instructed to answer questions, demonstrate programming concepts, and to give suggestions that advance your understanding and help you solve programming problems. They are not there to do it for you! TAs are instructed to report abuse of the help desk system, so please be polite, and respect the aforementioned boundaries. Habitual abuse of the TA Help Desk system may result in a temporary or permanent ban!

Academic integrity: The Department of Computer Science enforces UVM’s Code of Academic Integrity. Any suspected violation of this policy will be referred immediately to UVM’s Center for Student Conduct (<https://www.uvm.edu/sconduct>). Sanctions for a violation may include a grade of XF in the course. Additional violations can result in dismissal from the university. In a word: *Don’t*. All students should read and understand this policy. See: <https://go.uvm.edu/cai>.

Collaboration on quizzes and exams is strictly prohibited. Use of online services as a source of solutions is strictly prohibited. Using generative AI such as ChatGPT or Claude, or websites such as Chegg or Course Hero to complete coursework is a form of academic dishonesty. Work you submit for an individual grade must be your own. Any work not produced by you must be cited. For certain assignments, students may collaborate on homework (typically limited to teams of two). If you collaborate with another student on an assignment, be sure to indicate team members as specified. If you have any questions, ask!

Any attempt to tamper with or defeat any autograder is a form of academic dishonesty. This applies wherever autograders are in use, for example on Brightspace or Gradescope.

All code submitted by students is subject to code similarity review.

Exams, quizzes, homework assignments, answer keys and solutions, presentations or lecture notes, specifications and rubrics are copyright protected works, unless clearly and explicitly indicated otherwise. Any unauthorized copying or distribution of protected works is a violation of federal law and may result in disciplinary action. This includes submission of protected works as prompts to generative AI. Sharing of course materials without the specific, express approval of the instructor may be a violation of the University’s Code of Academic Integrity and an act of academic dishonesty, which could result in disciplinary action. Violations will be handled under UVM’s Intellectual Property Policy and Code of Academic Integrity, as appropriate. See: <https://go.uvm.edu/ipp> and <https://go.uvm.edu/cai>.

Attendance: The UVM attendance policy is available at <https://go.uvm.edu/srr>. There will be no make-ups for in-class active learning exercises if you did not attend class without prior notification and approval. While your attendance is ungraded, we will have frequent in-class assessments, and a good attendance record may be taken into consideration when handling exceptions if they arise.

If you are not able to attend in-person classes please notify the instructor via email as soon as possible. Depending on the nature of your absence, it may be appropriate for you to contact UVM Student Health Services (<https://www.uvm.edu/health/SHS>), CEMS Student Services (<https://www.uvm.edu/cems/student-services>), or the Dean's Office for your college. In many cases, these can provide an official request for flexibility on your behalf. While reasonable accommodations will be granted in the event of documented illness or emergencies, you are responsible for making up any work you have missed.

Class participation: You are expected to be an active participant in class. The more engaged you are, the more you will learn—and the more fun you'll have. This includes being prepared and attentive, responding when called on, participating in group discussion, and asking questions as appropriate. When it comes to asking questions, *please don't be shy!* There's no such thing as a "dumb" or "silly" question. If there's something you don't understand—*ask!* Asking questions helps you understand the material presented in the course. Asking questions is good for your classmates. It's almost certain that if you need clarification on some point, that there's at least one other student in the class with the same question. So help each other out—*ask!* Finally, when you ask a question you help the instructor to do a better job of explaining. If someone explains something, and you still don't quite grasp it, it's not unlikely that the explanation could be improved or clarified.

You're expected to read materials, watch videos, *etc.* as *advance* preparation for class. In class, we will have extensive, hands-on programming exercises. Much of this will be done using the "pair programming" approach—two people work as a team. There are two roles: one person serves as the "driver" (at the keyboard) and the other serves as the "navigator." Every 15–20 minutes you'll switch roles. So be prepared to work in teams.

Late policy / extensions: Each homework assignment has a specific due date / time. You may submit work up to 24 hours after the due date / time, however, late submissions will be penalized 20%. Submissions that are more than 24 hours late will not be accepted unless an extension has been granted. We will consider reasonable requests for extensions when extenuating circumstances arise. (It can't hurt to ask.) However, extensions will not be granted if the request for extension is made within 24 hours of the time an assignment is due, except in the most extraordinary circumstances. So if you wish to request an extension, *do so early!* If an extension is granted, you must submit your work by the agreed-upon extension date.

Student course evaluations: Students are warmly encouraged to complete an evaluation of the course at its conclusion. Evaluations are anonymous and confidential, and the information gained, including constructive criticisms, will be used to improve the course.

Defects / bonus points: As you might expect, we will deduct points on assignments, quizzes, or exams where you've made an error. It's only fair that instructors should be held to a similar standard. Accordingly, bonus points are awarded for reporting and correcting defects in instructor-written course materials.

- 1.0 point: Material defect. This includes any error whatsoever in code or solutions, or any error in writing that changes materially the sense of what is written. This also includes incorrect due dates for posted assessments.
- Up to 0.5 point: Minor defect. This includes typos, misspellings, or minor errors which do not affect materially the readability or sense of what is written. Determination of points for minor defects is at the instructor's sole discretion.

Due to the unfortunate behavior of some students who have favored the practice of "bonus point mining" over proper study, no student may earn more than three bonus points in this course.

Due to the fact that we are still writing and revising autograders, autograder defects are not fair game for defect bonus points (but by all means, please report, and perhaps we will award a discretionary point here and there).

Bonus points for any given defect are only awarded to the first student (across all sections) who correctly identifies the error *and provides a valid correction*. Bonus points are not available for any materials which are clearly marked as drafts. Bonus points are added to your final grade before assigning letter grades—a point or two may make a big difference. Happy hunting.

The secret word is “garland.”

Diversity, equity, and inclusion: UVM is a place where you should be treated with respect and kindness. We welcome individuals of all ages, backgrounds, beliefs, interests, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability, and other visible and non-visible differences. All students are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the community. If you ever feel that you have been unfairly treated or judged by an instructor, a mentor, another student, or another member of the community, please let someone know. Your instructors and advisors in the CEMS Office of Student Services are available to discuss any concerns, or you can report an incident of bias through the bias report program (<https://go.uvm.edu/brp>).

Conduct: Be kind to one another and to yourself. Be respectful of yourself, others, and the institution. Please arrive on time. Please, no food in class. Please, no cell phones in class (except for using the iClicker app when requested). You may use a laptop or tablet, but only for active learning sessions, pair programming, taking notes, or assistive technologies.

For other policies on classroom conduct, please see: <https://go.uvm.edu/srr> and <https://go.uvm.edu/csc>.

Accommodations: In keeping with UVM policy, if you have a documented disability and are interested in utilizing ADA accommodations, you should contact Student Accessibility Services (SAS), the office of Disability Services on campus for students. SAS works with students and faculty in an interactive process to explore reasonable and appropriate accommodations, which are communicated to faculty in an accommodation letter.

Contact SAS: A170 Living/Learning Center; +1 802 656 7753; access@uvm.edu; or visit <https://www.uvm.edu/access>.

If you are entitled to use the Exam Proctoring Center, please book reservations at least four days in advance.

Promoting health and safety: If you are concerned about a UVM community member or are concerned about a specific event, we encourage you to contact the Dean of Students Office (+1 802 656 3380). If you would like to remain anonymous, you can report your concerns online by visiting the Dean of Students website at <https://www.uvm.edu/deanofstudents>.

Wellbeing resources:

- Center for Health and Wellbeing: <https://www.uvm.edu/health/services>
- Counseling and Psychiatry Services (CAPS): +1 802 656 3340
- Food Insecurity Assistance: <https://www.uvm.edu/health/food-insecurity-uvm>

Student advocacy: https://www.uvm.edu/deanofstudents/student_advocacy

Your identity at UVM: Students at UVM can specify the first name and pronoun they want used on campus. For information on how to update your preferred name and personal pronouns as well as keeping your legal name private, and UVM policy on lived name and pronouns, see: <https://go.uvm.edu/lnpr>.

Religious holidays: Students have the right to practice the religion of their choice. In order to receive extensions or excused absences, you should submit via email your documented religious holiday schedule within the first week of the course. Reasonable extensions will be granted where assignment deadlines conflict with religious holidays.

Student athletes: In order to receive extensions or excused absences, you should submit via email appropriate documentation as soon as possible, preferably within the first week of the course. Reasonable extensions will be granted where assignment deadlines conflict with team events or team travel.

Statement on alcohol and other drugs: We want you to get the most you can out of this course. Therefore, you are expected to familiarize yourself and abide by the University's policies with regard to alcohol, cannabis, tobacco, and other drug use. See: <https://go.uvm.edu/actd>. Please do everything you can to optimize your learning and to participate fully in this course.

Class format changes: The University of Vermont reserves the right to make changes in the course offerings, mode of delivery, degree requirements, charges, regulations, and procedures contained herein as educational, financial, and health, safety, and welfare considerations require, or as necessary to be compliant with governmental, accreditation, or public health directives.

Changes to this document: This document is subject to change. Any such change will be communicated via an announcement on Brightspace. The latest version of the syllabus will always be available on Brightspace.

DRAFT