CS1420 Course Syllabus

Course Prerequisites: Math at least equivalent to college algebra. Pass CS 1420 Placement Exam. Note: Placement Exam is waived for Fall 2024. Some prior programming experience is expected.

Semester: Spring
Year: 2025
Course Prefix: CS
Course and Section #: CS1420-001
Course Title: Accelerated Introduction to Programming
Credits: 3

Course Description

Introduces techniques, tools and skills necessary to effectively program computers. Demonstrates algorithmic thinking using procedural and object-oriented concepts. Presents problems of increasing size and complexity requiring standard libraries and other appropriate language constructs.

This course in not a GE course. It is a one-semester alternative to taking CS 1400 and CS 1410.

Course Attributes

This course has the following attributes:
☐ General Education Requirements
☐ Global/Intercultural Graduation Requirements
☐ Writing Enriched Graduation Requirements
☒ Discipline Core Requirements in Program
☐ Elective Core Requirements in Program
☐ Open Elective
Other: *Click here to enter text.*

Instructor Information

Instructor Name: Eshita Zaman

Student Learning Outcomes

Upon successful completion of this course, students will be able to:

- Design procedural and object-oriented solutions to programming problems.
- Implement solutions to programming problems that combine procedural and object-oriented features.
- Write readable, maintainable code with a consistent programming style.
- Test programs to assure quality of design and implementation.

Course Materials and Texts

Required materials, fees and technology:

Textbook

Learn AI-Assisted Python Programming with GitHUb Copilot and ChatGPT, 1st Edition. by Leo Porter and Daniel Zingaro
ISBN-13 : 978-1633437784
Publisher: Manning

Other materials, fees and technology

- Sign up for a Github user account (free)

Optional materials, fees and technology:

- Codespaces online IDE or VSCode local install with Copilot and other extensions
- Install the latest version of Python 3.12 on your own machine (recommended) [free]
- Install IPython shell (free)

Course Requirements

Course Assignments, Assessments, and Grading Policy

Grading Scale:

The following grading standards will be used in this class:

Grade	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	E
Percent	93+	90+	87+	83+	80+	77+	73+	70+	67+	63+	60+	0+

Assignment Categories

Activity	Percent
Projects 1-5	32%

Activity	Percent
Desert Shop (12 parts)	40%
Module Questions (14)	8%
Midterm Project	10%
Project Design + Final Project	10%
SRI (Extra Credit)	1%

Late Work Statement:

Partial credit for points can be deceptive on software assignments, because it is possible to get a "good grade" for software that doesn't work or where a student never learns all the stages needed for completion but passes. If you find yourself in a situation where you cannot complete a project on time, it may be better to submit something complete a little late than to routinely submit partially-complete projects always on time.

0-1 business day late: 10% penalty
more than 1 business day late: 25% penalty

Last day to submit anything other than the final project: TBD

Your instructor may allow you to correct problems and resubmit assignments, especially for Dessert Shop. Any late penalties assessed on the date of original submission will still apply. Suppose Dessert Shop part 5 is submitted 1 day late with problems. Those problems are fixed in part 6 and part 5 is resubmitted. Part 5 is regraded with the 1-day penalty applied. If part 5 is resubmitted after part 6 is due, the instructor may apply the more than 1 day late penalty to the resubmission.

Assignment and Assessment Descriptions

Assignments:

Projects

Do not copy someone else's code or files or allow someone else to copy yours. You are welcome, and encouraged, to discuss assignments with others, get help from other students, and so on, study together. Don't share code, or work on one shared code base unless specifically given permission to do so.

If I find copied code, all participants' scores will be penalized, which could result in a score of zero for an assignment. That means that if you let another student read your code you might not get any credit on the assignment. Repeated incidents of copied code will result in more severe penalties, possibly including failing the course and being reported to the Office of Student Conduct.

Plagiarism and copy issues become problematic in the age of Generative AI. If everyone uses ChatGPT and the same or nearly identical prompts, the AI will generate similar code. This is one reason that you are asked to note use of AI in your code when you use it. I will simply say, this statement is meant to set expectations for learning, and no one is expected to be penalized for this without reason.

Whatever an AI generates, don't turn in code you could not, would not, or should not have written. That will be penalized heavily. The instructor is the judge of that.

It's very important to follow the assignment specifications when given. If your program doesn't produce any relevant output, you won't receive any credit for it. Your program should not depend on platform-specific features such as Windows line endings or path separators. Instead, you should use Python platform-independent mechanisms so that your program can run on any platform that has an appropriate Python interpreter.

Please check your files before turning them in. Github Classroom submission triggers may do some checking upon submission too.

Projects are graded as follows

Correctness is determined by the rubric and the judgement of the instructor. Style score is determined by **ruff check** score. The project score is computed as 34×(correctness)+14×style. For example, suppose a submission passes 2/3 correctness checks, gets .87 out of 1.0 on style. The score would be .78 or 78%. If the submission is corrected to pass 3/3 checks, 1.0/1.0 style, with no late penalties the score would be 1.00 or 100%.

Module Questions

Module Questions are a set of mostly reflective questions to be answered at the end of each module. They may come from many sources, not just the reading--but they will be related to the module and the project, not out-of-the-blue. The accuracy and quality are scored out of 10 points for each module. Partial scoring is possible. Late penalties, if any apply to that latest submission.

Exams

Project 6 is the midterm exam.
The final project is the final exam.

Assessments:

The midterm project and the final project are the only summative assessments in the course.

Projects are both formative and summative: Formative in that they may be resubmitted and regraded based on feedback received. Dessert Shop may be considered both formative and summative because the parts build on each other.

Module Questions are formative in that a student may answer the questions, do some additional study if needed, and resubmit new answers for a new score.

In most respects, the midterm and final are like the other projects. Two things are different however--you are not allowed to discuss them with anyone else except to ask questions of your instructor or IA. You may not get help on them from other people.

Required or Recommended Reading Assignments
This course has required reading and exercises each week.

General Description of the Subject Matter of Each Lecture or Discussion

Description of how course works:

Canvas

- Used to record grades
- Has Modules with Readings, Project Phases, Module Questions
- Announcements will be posted using Canvas Announcement
- Students may contact instructor using Canvas email or Outlook or Teams

Teams

- Class Teams channel is preferred place for posting questions to the class generally or asking for help on stuff
- Use the Chat feature or email or video call for private messages to the instructor or IA

For this three (3) **credit-hour** course students should expect to spend up to **9+ hours a week** completing course activities. In face to face sections of this course, each hour of attending class counts toward 1 of the 9 hours for academic engagement. What about students who choose not to attend class? This is really about setting workload expectations, not about attendance policy. If a student is spending significantly more than 6 hours per week (face to

face mode) on assignments, stop and get help from your instructor. Don't suffer, don't complain that the workload is unfair--get help. Please.

The main thrust of this course is to teach you problem-solving skills and how to write programs with Python as the vehicle. Some people call this "how to think like a Computer Scientist" using Python. This class will be very much like taking CS 1400 and 1410 on the block, but with the added edge that we also teach you how to use an AI to help you learn as an integral part of the class. The pace is FAST, so be prepared to work. In a face to face class, you are expected to attend class whether your instructor takes attendance or not. If you miss class, it is up to you to acquire the information you would have known if you had attended. If you know you will routinely be unable to attend in person, contact your instructor or consider switching to an online section.
This class is framed around **project-based learning**. There are no shortcuts to learning by doing, and in this class, you learn by doing. See below for a statement on how you are expected to use AI in this class. There projects due almost every week, and questions that must be answered at the end of each module. There are no "small homework" type exercises that you might be used to, because an AI can generate both exercises and answers if you need them for practice on your own. It will also grade your answers if you ask it to. Furthermore, an AI can generate good answers to every project you will do in this class--so the point really is for you to learn to program aided by an AI, not for it to replace your learning.

Tips For Success

1. Attend class (face to face or online)
2. Make use of any extra, notes, code and video hints you are given.
3. Ask your instructor or IA for help, and learn to use Gen AI as indicated to help you learn faster and get done quicker.
4. The course schedule lists due dates that are unlikely to get moved. People learn at different rates, and sometimes need a little longer to "get it". Give yourself extra time by starting sooner than you think you need to.
5. It's better to complete a phase 100% and late, than to hand in a partial solutions on time.
6. When you submit code, we grade most things on correctness AND style. Take the time to get the style right because it affects your grade.

Automatic room recordings should be available though Canvas.

For this three (3) **credit-hour** course students should expect to spend up to **9+ hours a week** completing course activities. For this face to face course, 3 of those hours are in-class time each week.

Third Party Usage:

See the sections **Other Materials and Technology** and **Optional materials, fees and Technology** above. Links to privacy policies for required and some optional tools are listed here.

- GitHub privacy statementLinks to an external site.
- ruff
    - a Python code linter and formatter tool

    - a drop-in replacement for the **Black** code formatter and linter tools like pylint, flake8 and pep8.
    - depends on Github Classroom security when run on submissions
    - is local to a machine when installed and run on the student's machine (no PII involved)
      Links to an external site.
- VSCode with Codespaces (a Docker Image running remotely)
    - VSCode privacyLinks to an external site.
    - Docker privacy

Required Course Syllabus Statements

Generative AI

Artificial Intelligence (AI) Expectations and Requirements

ChatGPT (and similar Tools) in This Course: Use ChatGPT as a learning assistant, not as a crutch. If you use it, cite it at the top of your code. **You** are responsible to make sure that any code or content does what it is supposed to do and says what you want it to say. Don't accept anything it generates at face value without checking it critically. These days potential employers will expect you to know how to use tools like ChatGPT to generate code, so it is a skill we need to teach you. If it helps you learn some things faster, GREAT because we can spend class time on more interesting topics. Just remember: If you REALLY want to be good, work for it.

Does your instructor REALLY expect you to use GEN AI in this class? REALLY? Yes! Suggestions for using it responsibly:

- If you don't have a clue, use AI to get a clue.
- If you don't understand a concept, ask AI for an explanation with examples.
- If some code isn't working, ask AI for help on that snippet, including broken APIs.
- If you want help on improving your code, ask AI how you might improve some function or section of your code.
- Tell AI to guide you toward a solution rather than giving you a solution immediately.
- If you use AI, remember to note you've used it in your module docstrings and in your project submission document.
- If you feel like you need smaller exercises or practice with some concept before working on some part of your project, use AI to generate exercises for you.
- Whatever the AI generates, don't turn in code you could not, would not, or should not have written. That will be penalized heavily. The instructor is the judge of that.

Using Remote Testing Software

☒ This course does not use remote testing software.

☐ This course uses remote testing software. Remote test-takers may choose their remote testing locations. Please note, however, that the testing software used for this may conduct a brief scan of remote test-takers' immediate surroundings, may require use of a webcam while taking an exam, may require the microphone be on while taking an exam, or may require other practices to confirm academic honesty. Test-takers therefore shall have no expectation of privacy in their test-taking location during, or immediately preceding, remote testing. If a student strongly objects to using test-taking software, the student should contact the instructor at the beginning of the semester to determine whether alternative testing arrangements are feasible. Alternatives are not guaranteed.

Required University Syllabus Statements

Accommodations/Students with Disabilities
Students needing accommodations due to a permanent or temporary disability, pregnancy or pregnancy-related conditions may contact UVU Accessibility Services at accessibilityservices@uvu.edu or 801-863-8747.

Accessibility Services is located on the Orem Campus in BA 110.

Deaf/Hard of Hearing students requesting ASL interpreters or transcribers can contact Accessibility Services to set up accommodations. Deaf/Hard of Hearing services can be contacted at DHHservices@uvu.edu

DHH is located on the Orem Campus in BA 112.

Academic Integrity
At Utah Valley University, faculty and students operate in an atmosphere of mutual trust. Maintaining an atmosphere of academic integrity allows for free exchange of ideas and enables all members of the community to achieve their highest potential. Our goal is to foster an intellectual atmosphere that produces scholars of integrity and imaginative thought. In all academic work, the ideas and contributions of others must be appropriately acknowledged and UVU students are expected to produce their own original academic work.

Faculty and students share the responsibility of ensuring the honesty and fairness of the intellectual environment at UVU. Students have a responsibility to promote academic integrity at the university by not participating in or facilitating others' participation in any act of academic dishonesty. As members of the academic community, students must become familiar with their rights and responsibilities. In each course, they are responsible for knowing the requirements and restrictions regarding research and writing, assessments, collaborative work, the use of study aids, the appropriateness of assistance, and other issues. Likewise, instructors are responsible to clearly state expectations and model best practices.

Further information on what constitutes academic dishonesty is detailed in UVU Policy 541: *Student Code of Conduct*.

Equity and Title IX
Utah Valley University does not discriminate on the basis of race, color, religion, national origin, sex, sexual orientation, gender identity, gender expression, age (40 and over), disability, veteran status, pregnancy, childbirth, or pregnancy-related conditions, citizenship, genetic information, or other basis protected by applicable law, including Title IX and 34 C.F.R. Part 106, in employment, treatment,

admission, access to educational programs and activities, or other University benefits or services. Inquiries about nondiscrimination at UVU may be directed to the U.S. Department of Education's Office for Civil Rights or UVU's Title IX Coordinator at 801-863-7999 – TitleIX@uvu.edu – 800 W University Pkwy, Orem, 84058, Suite BA 203.

Religious Accommodation

UVU values and acknowledges the array of worldviews, faiths, and religions represented in our student body, and as such provides supportive accommodations for students. Religious belief or conscience broadly includes religious, non-religious, theistic, or non-theistic moral or ethical beliefs as well as participation in religious holidays, observances, or activities. Accommodations may include scheduling or due-date modifications or make-up assignments for missed class work.

To seek a religious accommodation, a student must provide written notice to the instructor and the Director of Accessibility Services at accessibilityservices@uvu.edu. If the accommodation relates to a scheduling conflict, the notice should include the date, time, and brief description of the difficulty posed by the conflict. Such requests should be made as soon as the student is aware of the prospective scheduling conflict.

While religious expression is welcome throughout campus, UVU also has a specially dedicated space for meditation, prayer, reflection, or other forms of religious expression.