Course Syllabus

Cornell Bowers C·IS
Computer Science

Course: **CS 1112 Introduction to Computing: An Engineering and Science Perspective** (Fall 2023)

Instructor: **K.-Y. Daisy Fan** (kdf4)

Course Website: **https://www.cs.cornell.edu/courses/cs1112/2023fa/ (https://www.cs.cornell.edu/courses/cs1112/2023fa/)**

Credit Hours and Credit-Hour Options: **4 credits, Graded.** S/U Optional.

Pre-requisite: MATH 1110, MATH 1910, or equivalent. Assumes student is comfortable with mathematics (at level of one semester of calculus) but has **no** prior programming experience. Co-enrollment in MATH 1110 or MATH 1910 is acceptable assuming that student is comfortable with the concepts of series summation and derivatives.

Alternative: **CS1110 (http://www.cs.cornell.edu/courses/cs1110/)** (Introduction to Computing: A Design and Development Perspective)

Forbidden Overlap: due to an overlap in content, students will not receive credit for both CS 1110 and CS 1112.

Course Description

Programming and problem solving using Python. Emphasizes the systematic development of algorithms and programs. Topics include iteration, functions, arrays, strings, recursion, object-oriented programming, algorithms, and data handling and visualization. Assignments are designed to build an appreciation for complexity, dimension, randomness, simulation, and the role of approximation in engineering and science. Weekly discussion section provides guided practice on the computer, with staff present to help. NO programming experience is necessary; some knowledge of Calculus is required.

Expected Outcomes

By the end of the course, a student will be able to:

- apply procedural statements--assignments, conditional statements, loops, function calls--and arrays in Python programs.
- design, code, and test small Python programs that meet requirements expressed in English. This includes a basic understanding of top-down design.
- implement Python classes and subclasses given an object-oriented design.
- implement basic sorting and searching algorithms.
- process data and produce scientific graphics using Python

Times & Places

- **Lecture**: **TuTh 2:55 - 4:10pm** Olin Hall 155 (28 sessions total)
- **Discussion**: Student enrolls in one section and attends weekly (14 sessions total). *Bring a laptop to Sections 203, 205, and 206,* which take place in regular classrooms without computers. Sections 201, 202, and 204 take place in a computer lab; therefore you can, but don't have to, bring a laptop.
  - **Section 201 W 10:10AM - 11:25AM** Upson Hall 225 (computer lab)
  - **Section 202 W 11:40AM - 12:55PM** Upson Hall 225 (computer lab)
  - **Section 203 W 1:25PM - 2:40PM** Hollister Hall 306 (classroom--bring a laptop)
  - **Section 204 W 1:25PM - 2:40PM** Upson Hall 225 (computer lab)
  - **Section 205 W 2:55PM - 4:10PM** Snee Hall 1150 (classroom--bring a laptop)
  - **Section 206 W 2:55PM - 4:10PM** Hollister Hall 306 (classroom--bring a laptop)

Office and Consulting Hours

The instructor and teaching assistants hold weekly office hours, and undergraduate consultants hold weekly consulting hours. See the **Staff and Office Hours page** for the time and location beginning on Aug 28.

Communication

Course announcements and materials will be posted on Canvas. Assignment submission and feedback will be managed by **CMS (https://cmsx.cs.cornell.edu/)** and Gradescope. If you have a question about course material, post it to **Ed Discussion** (online forum); public posts are preferred so that others can benefit from the discussion (posts can be anonymous to other students). If you need to request special accommodation or discuss something one-on-one with the instructor, please use the instructor's office hours or email.

Material

- Our textbook is **Think Python 2e (https://greenteapress.com/wp/think-python-2e/)** by Allen B. Downey. The e-book is free and can be read in HTML or downloaded in PDF. You can also buy a print copy if you like. This is an excellent and terse book--you should read in detail (not just skim over) the sections that we refer to and it won't take long! Be aware that it presents the course material in a different order than we do.
- We will use Python 3.10 (programming language), Jupyter Notebook (programming environment), and an additional integrated development environment (IDE). All of them are free!
- You need an iClicker remote (the physical device, not the mobile app) to answer in-class questions.

Assessment

Basis of Grade Determination: *exercises* (weekly), *projects* (programming projects), *exams* (two prelims and a final), and *in-class questions*.

- **Exercises** are short programming tasks that give you practice on the topics of the week. They are assigned weekly and you get help and additional instructions on them during your discussion section. Exercises are "graded" mostly on effort. You are encouraged to collaborate with discussion section classmates but you will submit your work as an individual.
- **Projects** are medium-size programming assignments through which you synthesize the knowledge gained in recent weeks. For most projects you may work individually or with one partner (you may have a different partner on different assignments).
  - Your lowest project score will be dropped automatically at the end of the semester *provided that you scored at least 50% on that project*. This helps accommodate an unusual, difficult situation without having to justify the circumstances to the instructor, so long as the submission still reflects a reasonable amount of effort. Submissions will be accepted one day late with penalty.
- **Exams**: You must write the exams at their scheduled times unless University-allowed accommodations have been granted and discussed with the instructor at least two weeks before the exam in question.
- In-class questions are *for you* to check your understanding of concepts discussed in class; they are usually done using your iClicker remote. A small weight is assigned in order to encourage active reflection in class but not to check attendance--you can miss up to half the questions without affecting your score.

Your course score is computed using the following weights:

In-class questions	1%
Exercises	5%
Projects	27%
Prelim 1	20%
Prelim 2	20%
Final Exam	27%

Grading Scale: Letter grades or S/U grades (depending on the student's enrolled grade option) will be assigned in accordance with the **[Cornell University grading system (https://courses.cornell.edu/content.php?catoid=31&navoid=7933#Grading_System)](https://courses.cornell.edu/content.php?catoid=31&navoid=7933#Grading_System)** . Historically, a course total over 93 is in the A range (A-, A, A+), a course total over 80 is at least in the B range, and a course total over 65 is at least in the C range. With the S/U grade option, you need a "C-" or better (determined as stated above) in order to receive an "S".

Schedule and Topics (subject to change)

- ○ Weeks 1-3: Fundamentals and procedural programming: variables & assignment, conditionals, loops
- ○ Weeks 4 & 5: Developing algorithms, user-defined functions, and memory
- ○ Weeks 6 & 7: Simple data structure: list and string
- ○ Weeks 8-10: Nested lists, NumPy arrays, and applications
- ○ Weeks 10-12: Object oriented programming: custom type through class definition, list of objects, introduction to inheritance
- ○ Weeks 13 & 14: Recursion, algorithms for sorting and searching, including divide-and-conquer

Alongside the above topics, throughout the course we will practice the skills of **problem analysis and decomposition** as well as **testing and debugging**.

Prelim 1: October 17, 2023 7:30 - 9:00pm
Prelim 2: November 14, 2023 7:30 - 9:00pm
Final Exam: to be scheduled by the University

Course Management

Community of Learning

We aim to create an inclusive learning environment where diversity and individual differences are respected, appreciated, and recognized as a source of strength. It is expected that students and staff in this class will respect differences and demonstrate diligence in understanding how other peoples' perspectives, behaviors, and worldviews may be different from their own. By participating in this course, all students and staff commit to contribute positively to our community of learning:

- Recognize that everyone will start from different bases of knowledge. Be respectful and constructive when pointing out mistakes.
- Listen to one another and, especially during group work, actively encourage everyone to contribute.
- Help build a lively and active learning environment. Ask *and answer* questions on our Ed Discussion forum, always remembering to be respectful and constructive.

Academic Integrity

Integrity is a cornerstone of our learning community; it is about respecting yourself and respecting others. You respect yourself by submitting work completed through your own effort; you respect others by acknowledging contribution from others when such external contribution is allowed (e.g., on group projects). When your individual effort is required (such as on an exam), you may neither seek nor accept help from others. Refer to the University **[Code of Academic Integrity (http://theuniversityfaculty.cornell.edu/academic-integrity/)](http://theuniversityfaculty.cornell.edu/academic-integrity/)** for further information. Ignorance of the Code is not an acceptable excuse.

During lecture, you may only respond to polling questions using your own iClicker remote; it is a violation of academic integrity to respond on behalf of other students not in attendance.

On any programming project, it is a violation of academic integrity to:

1. Look at or be in possession of the code of another group in this semester or a previous one with a similar assignment.
2. Show or give your code to another student not in your group.
3. Post code on any communication platform (including Q&A sites and public posts on Ed Discussion) that other students can see.
4. Use any artificial intelligence (AI) powered coding tool to generate code for any part of your project. See further explanation below.

You may discuss assignments with others at a high level, but the discussion should not extend to writing actual code, picking variable names, agreeing on specifications or comments, etc. If someone else contributes a key idea affecting your program design,

you must credit them in a code comment, clearly specifying the scope of their contribution.

If you do an assignment with a partner, you must work together as much as possible. It is a violation of academic integrity to submit as a group if both members cannot claim *joint authorship of all portions of the submission*.

In an introductory programming course, we believe you must code yourself to internalize the material. The act of construction reinforces concepts, checks your knowledge, and gives you opportunities for learning by failure that ultimately accelerate your learning. Generative AI tools such as ChatGPT, GitHub Copilot, and advanced auto-completion plug-ins deny you the opportunity to learn by construction. Consequently, we forbid the use of AI-powered coding tools in this course. To be explicit, use of GitHub Copilot, ChatGPT, or any other AI-powered coding tool is an academic integrity violation.

If we suspect that the Code of Academic Integrity is not being upheld, we may upload student submissions to 3rd-party services that detect plagiarism; enrollment in this course implies consent for your submissions to be used in this manner.

Accommodation

Students with Disabilities: Your access in this course is important to us. In order to have adequate time to arrange your approved accommodation, you must request your accommodation letter from **Student Disability Services (SDS) (http://sds.cornell.edu/) no later than the add/drop deadline for the semester**. If you become registered with SDS later in the semester, you must request your accommodation letter as soon as possible. Once SDS approves your accommodation letter, it will be emailed to both you and me.

Students with exam accommodations, this course is participating in the SDS Alternative Testing Program. You will receive your exam logistics from the ATP Testing Coordinator **sds-testing@cornell.edu (mailto:sds-testing@cornell.edu)** closer to the exam date (no later than 48 hours prior). Please do not contact the course staff with questions about exam logistics, *as they will not be able to answer them*. Direct logistics questions to **sds-testing@cornell.edu (mailto:sds-testing@cornell.edu)**. Please be aware that the ATP evening prelim exams will begin at 6:30pm, one hour earlier than the main exam.

Students with other academic accommodations (not testing-related), please follow up with the course instructor to discuss the necessary logistics of your accommodation(s).

Physical and Mental Health: Your health and wellbeing are important! If your physical or mental health prevents you from completing required work, email the course instructor as soon as possible to make an alternative arrangement for the missed work. (Remember: your lowest project score will be dropped so long as you made a good attempt given your condition). Additionally, there is a continuum of campus resources available to support your mental health: **https://mentalhealth.cornell.edu/get-support/support-students (https://mentalhealth.cornell.edu/get-support/support-students)** .

Copyright Notice

All materials distributed in this course are copyrighted and may not be distributed further. They are intended for your sole use and may not be posted on any public or private website, or by any other sharing method (e.g. fraternity/sorority exam files).