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/

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 (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 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** 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 the Anaconda distribution of Python 3.11 (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: