CS 1110. Introduction to Computing using Python

Credit: 4 hours

Prerequisites: Basic high school mathematics (no calculus) but no programming experience.

Catalogue description: Introduces programming and problem-solving using Python. Emphasizes principles of software development, style, and testing. Topics include an operational model of Python execution, procedures and functions, iteration, recursion, lists, strings, algorithms, exceptions, object-oriented programming. Weekly labs provide guided practice on the computer, with staff present to help.

Required or elective: Common-curriculum course: one of CS 1110–1115 is required.

Textbook(s) and other materials: Anaconda, a free development environment for Python. Installation details on course website: https://www.cs.cornell.edu/courses/cs1110/2022sp/materials/install-python-landing.html

Course objectives:

The primary goal of CS1110 is to give students a basic introduction to object-oriented and procedural programming, using Python.

Topics covered:

Primitive types and class types
Functions (procedures, and fruitful functions), including recursion
Basic statements: assignment, if-statements, loops, blocks, function calls
Objects and classes, including subclasses, inheritance, and overriding
Sequences and dictionaries
Testing and debugging
Program development; top-down design and object-oriented design
Basic searching and sorting algorithms
Model of python execution
Programming style considerations

Class/laboratory schedule:

Two 50-minute lectures per week.
One or two 50-minute labs per week.

Contribution of course to meeting the professional component:

This course contributes to item (a) of the professional component (one year of a combination of college level mathematics and basic sciences appropriate to the discipline) and to item (b) engineering design.

Course outcomes and their relation to ABET program outcomes a-k:

1. Fluency in the use of procedural statements — assignments, conditional statements, loops, function calls — and lists (a, b, c)
2. Mastery in designing, coding, and test small Python programs that meet requirements expressed in English. This includes a basic understanding of top-down design (a, b, c)
3. Competency in using recursion in Python programs (a,c,e)
4. Competency of object-oriented programming as used in Python: classes, subclasses, inheritance, and overriding. Understand the basics of OO design. (a, b, c, k)
5. Knowledge of basic searching and sorting algorithms, and the basics of vector computation. (k)

Assessment of course outcomes

Course outcomes will be assessed by examination (two midterm exams and one final exam) as well as student-submitted homework and programming assignments. See the assessment details on the course website: https://www.cs.cornell.edu/courses/cs1110/2022sp/policies/

Person preparing this description and date:

Lillian Lee and Walker White, 16 June 2022; adapted from original by David Gries (27 December 2010)

Ethical behavior statement:

Here are the collaboration rules specific to CS1110:

https://www.cs.cornell.edu/courses/cs1110/2022sp/policies/cs1110integrity.html

Be sure to understand what constitutes unauthorized giving or receiving of help for labs (any *small-group, synchronous* collaboration OK but not public posting of questions/answers), assignments (limited collaboration), and exams (none permitted).

Of particular note:

- The *minimum* penalty upon a guilty finding for an academic integrity violation for receiving unauthorized help: besides the mandated letter to the student's college, a negative score on the affected work (this is more than just a grade deduction, where one might retain some points). Hence, *a student who submits fraudulent work receives less credit than a student who didn't turn the work in at all* .
- The *minimum* penalty upon a guilty finding for an academic integrity violation for giving unauthorized help: the mandated letter to the student's college. *Please don't put your friends at risk by asking them for unauthorized help* .