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, and GUIs (graphical user interfaces). Weekly labs provide guided practice on the computer, with staff present to help. Assignments use graphics to help develop fluency and understanding.

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: http://www.cs.cornell.edu/courses/cs1110/2020fa/materials/python/

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:

One 50-minute synchronous, online lectures per week.
Up to 90-minutes of asynchronous, online videos posted per week.
Two 50-minute synchronous (online or in-person) 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 sequences. Be able to design, code, and test small Python programs that meet requirements expressed in English. This includes a basic understanding of top-down design. (a, b, c)
2. Understanding of the concepts of object-oriented programming as used in Python: classes, subclasses, inheritance, and overriding. Understand the basics of OO design. (a, b, c, k)
3. 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 (one midterm tests and one semi-final) as well as student-submitted homework and programming assignments. See the assessment details on the course website: http://www.cs.cornell.edu/courses/cs1110/2020fa/assessment/

Person preparing this description and date:

Walker White, 20 August 2020; adapted from original by David Gries (27 December 2010)

Ethical behavior statement:

We ask you not to cheat, in any way, shape, or form. In return, we will try our best to be fair about the amount of work we are giving you, in the grading of that work, and in giving you a course grade. You can always talk to us if you have any gripe or criticism about the course, and we will attempt to respond to it immediately.

Some forms of cheating are pretty straightforward. Others are less so. Every year, we find several academic integrity violations. In many of these cases, the students were not necessarily intending to cheat. CS 1110 has a very detailed policy on academic integrity. You should refer to the current web page for more information on this policy.

http://www.cs.cornell.edu/courses/cs1110/2020fa/assessment/integrity/