

## CS 1110. Introduction to Computing using Python

**Credit:** 4 hours

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

**Catalogue description:** Programming and problem-solving using Python. Emphasizes principles of software development, style, and testing. Topics include procedures and functions, iteration, recursion, arrays and vectors, strings, an operational model of procedure and function calls, 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 and GUIs to help develop fluency and understanding.

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

### **Textbook(s) and other materials:**

1. Downey, Allen B., *Think Python: How to Think Like a Computer Scientist, 2<sup>nd</sup> Edition*. O'Reilly, 2015. Obtain free PDF at <http://greenteapress.com/wp/think-python-2e/>
2. Anaconda, a free development environment for Python. Installation details on course website: <http://www.cs.cornell.edu/courses/cs1110/2019sp/materials/>

### **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; stepwise refinement (top-down design) and object-oriented design  
Basic searching and sorting algorithms  
A model of execution  
Programming style considerations

### **Class/laboratory schedule:**

Two 50-minute lectures per week.  
One 50-minute lab 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. Be fluent 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. Understand 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. Have knowledge of basic searching and sorting algorithms, and knowledge of the basics of vector computation. (k)

**Assessment of course outcomes**

Course outcomes will be assessed by examination (midterm tests and a final) and student-submitted homework and programming assignments.

**Person preparing this description and date:**

Anne Bracy, December 17, 2018; 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/cs1110/2019sp/policies/>