CSCI 1730 Introduction to Programming Languages

Assignments | Syllabus | Staff | Videos | Announcements

Though this course is over, the on-line materials are designed so you can do the exercises any time you want. In other words, you can run this class (for yourself) right now, in full!

The On-Line Version

New Version of the Textbook

Course Software (programming language)

Please make sure you're subscribed to our Google Group. All announcements posted to the announcements page will also be posted to the group.

Course Management

As an experiment, the Brown and on-line students will participate in one combined course management system. We are going to use Piazza. Please go to the course page and click on the Enroll link at top-right. You will need an access code, which we will provide to enrolled Brown students.

About the Course

cs173 is **completely new** this year. Our focus will be on an approach we call *semantics engineering*. There are two parts to that term, both relevant. We will focus on semantics, i.e., on the meaning of programs in languages. But we will do it with an engineering mindset, through construction and validation, not purely through mathematics (as is conventional in programming language research). You will understand this better as we go along.

There are many means of specifying a semantics, some more operational than others. We choose to specify them using interpreters. These have several disadvantages, and in more advanced programming languages courses you will learn much better techniques for describing a language's definition. They also have many advantages, however, not least that they are executable and useful as implementations in their own right. Thus, we will teach you to write interpreters to capture a variety of important programming language constructs.

In the first month of the course, we will go through the semantics engineering process in the small: you will be given an illustrative little language built by us, whose meaning you must completely decipher. (We use the phrase “decipher” intentionally: most real-world languages are under- or poorly specified, so almost all of them at least slightly resemble hieroglyphics.) In the remaining two months, you will repeat this exercise in the large: namely, for Python. By the end of the semester, then, a completely successful student will have an authoritative knowledge of Python and will have created a full, independent implementation of it. Naturally, we don't expect very many students to be completely successful.

In addition to these projects, there will also be written homeworks covering several topics including scope, garbage collection, and a spread of material on types. We will also have nearly daily quizzes that will force you to master basic mechanical skills and keep on top of the course as it progresses. All these components will contribute to your grade.