CS 263. Design of Programming Languages

Martín Abadi
Digital Equipment Corporation
Systems Research Center

Spring 1997

Time: MF 10:30-12:00
Location: Soda Hall room 405

Course Description

This is a graduate-level course on the design and analysis of programming languages. The course covers some basic notions that are relevant to a wide range of languages, like parameter-passing mechanisms, higher-order functions, type soundness, and formal semantics. The main emphasis of the course is on object-oriented languages and their features; in particular, we will consider objects, dynamic dispatch, object types, subtyping, classes, and inheritance.

The course is not a tour of programming languages, although we will discuss several languages. The course is not a step-by-step tutorial on how to design your own language, either; however, it should help you in this respect.

Prerequisites

The prerequisites for this course are programming and mathematical experience. The ideal programming experience is practical exposure to several different languages (like C, ML, and Java). The ideal mathematical experience is knowledge of mathematical logic and the ability to construct rigorous proofs (in particular by structural induction). A bit less may do.

If you are an undergraduate or a graduate student from another department, please talk to me.

Course Texts

There are two texts for the course:

- *The Formal Semantics of Programming Languages*, by Glynn Winskel.
- *A Theory of Objects*, by Martín Abadi and Luca Cardelli (compliments of Digital).

There will be some additional reading:

- I will hand out some papers.
- You will need some source to learn a little bit of Java. For this, you can go to Sun's web pages (`http://www.javasoft.com/doc/`), or you can pick a text (such as *Java in a Nutshell*, by David Flanagan).

Course Work

Course work will include reading, some exercises, and a term project.

Most of the work for the exercises will be fairly mathematical, but the exercises will also include some light programming and writing.

The term project can be a programming project, a survey of recent work on some relevant topic, or a small research paper. A short project proposal is due on April 18. The project is due on May 12, and will be presented to the class near the end of the semester.

You may talk to other students in the class about the exercises, provided your write-up gives them proper credit for their contributions. You may use any texts or other material in the exercises and in the project, as long as you cite your sources. You may do the term project in small groups, but in my opinion it is easier to do alone.

There is no final exam.

Grading

I will give you feedback on the exercises, but no formal grade. Your overall performance on the exercises will count for 10% of your grade for the course. The remaining 90% of the grade will be determined by the term project.

Course Topics

The following is a list of the major topics to be covered in the course:

- Introduction to semantics (operational, denotational, axiomatic)
- Types and typechecking, and other static checking
- A simple imperative language—IMP
- A language with higher-order functions—the lambda calculus
- Survey of object-oriented languages and features
- A calculus of objects
- Object types, subtyping, inheritance
- Polymorphism and data abstraction

Contact Information

Martín Abadi
`ma@pa.dec.com`
Telephone: (415) 853-2196 at Digital, 3-4009 in Berkeley.
Office hours M 1:30-3:00 Soda Hall room 615, and by appointment.
On the web, see:

- `http://www.research.digital.com/SRC/personal/Martin_Abadi/home.html`
- `http://www.research.digital.com/SRC/personal/Martin_Abadi/UCB_CS263/home.html`