UNIVERSITY OF CALIFORNIA
Department of Electrical Engineering
and Computer Sciences
Computer Science Division

CS 263 **Spring 1996**
K. A. Yelick

CS 263: Design of Programming Languages
Handout 1: General Information

Instructor: Kathy Yelick
E-mail: yelick@cs.berkeley.edu
Phone: x2-8900
Office: 505 Evans Hall

1 Purpose of this Course

This course is an introduction to modern programming languages. We will evaluating languages based on pragmatic considerations, such as ease of expressibility and implementation, and theoretical considerations using operational, denotational, and axiomatic semantics. We will cover basic language concepts, including parameter passing mechanisms, higher order functions, abstract data types, exception handling, type inference, object-oriented programming, and concurrency.

2 Prerequisites

The stated prerequisite for this course is CS164 or an equivalent undergraduate compiler course. In addition, you should have a significant amount of programming experience. Although there is no formal prerequisite in mathematics or computer science theory, a certain degree of mathematical maturity will be expected. This course is primarily intended for graduate students in Computer Science. If you are an undergraduate or a graduate student from another department you must talk to me for permission to take the course.

3 Course Work

There is no textbook for the course. Instead, we will be reading several of the classic papers as well as recent research papers in programming languages. The most up-to-date information about the course and readings will be on the course home page at:

`http://www.cs.berkeley.edu/ yelick/263`

In addition to reading lecture notes and papers, problem sets will be given out approximately once every two weeks. These will involve programming problems, exercises in formal semantics, and essay questions. There will be two exams, one during the first half and one during the second half of the semester. There also be final term paper or project. I will hand out suggestions for

projects later in the semester and encourage you to choose projects that overlap with your own research interests.

The problem sets will not be heavily weighed in your final grade (approximately 10%), but the exam questions will be closely related to the kind of questions asked on problem sets. The two exams will constitute roughly 50% of your grade and the final project/paper will be another 40%.

4 Collaboration

You are encouraged to talk to other students in the class about questions on the problem sets, although write-ups must be done individually and in your own words. You should also use the good scholarly practice of acknowledging your collaborators on your problem sets. The final project may be done alone or in groups or two or three students.

5 Topics

The following is a list of the major topics to be covered in lectures, roughly in the order in which they will be discussed.

- **Foundations of Programming Language Research**

  The Lambda Calculus

  Operational Semantics

  Axiomatic Semantics

  Denotational Semantics

- **Programming Paradigms and Language Concepts**

  Functional and Declarative Languages

  Object-Oriented Languages

  Exception handling

  Parallel Languages

- **Static Semantics**

  Types Checking and Type Inference

  Abstract Data Types

- **Applications of Formal Semantics**

  Abstract Interpretation

  Partial Evaluation

  Program Verification