CS 258. Winter 1997. Handout 1
January 7, 1997

Introduction to Programming Language Theory

TTh 9:30-10:45, Gates 100

Instructor: John Mitchell (`jcm@cs.stanford.edu`) Gates 476, 723-8634

Secretary: Phyllis Winkler (`winkler@cs.stanford.edu`) Gates 495, 723-4377

Sign Up: Please send email to `jcm@cs.stanford.edu` with "CS 258 sign-up" in the Subject header as soon as possible.

Electronic information: Homework assignments will be available on-line, in a directory reachable from `http://theory.stanford.edu/people/jcm/home.html`.

Text: The course will use *Foundations for Programming Languages,* by J. Mitchell, MIT Press, 1996.

Prerequisites: The prerequisites are familiarity with programming concepts and basic notions of mathematical logic (formulas, proof systems, semantics). In addition, the course requires the proverbial "appropriate level of mathematical sophistication" associated with rigorous proof. Translated into course numbers, the requirements are approximately (i) CS 157 or Phil 160A, (ii) some course in programming, such as CS 106, and (iii) familiarity with computability theory, at approximately the level of CS 109B or CS 154.

Course Work: There will be a weekly homework, a midterm exam, and a final. These will count 50%, 20% and 30% of the final grade, respectively. Tentatively, homework will be handed out every Thursday and due the following Thursday. Late homeworks will be accepted only if prior permission is given and papers are submitted before the solutions are handed out.

Exams: The midterm exam will be a one-week take-home exam (in place of homework for that week). The official final exam period for the course is 12:15–3:15 PM on Friday, March 21. The take-home final will be due at the end of the official exam period.

Handout Hangout: Handouts and graded homeworks that are not picked up during class will be available from the file cabinet at the entrance to the fourth floor B-wing of Gates Hall.

Abbreviated Course Outline

1. A typed functional language with higher-order functions. Syntactic issues and discussion of alternative evaluation mechanisms (operational semantics).

2. Universal algebra and algebraic data types. Basic properties of equational logic; specification and execution models.

3. Denotational semantics of functional languages with recursion over algebraic data types.

4. Operational and denotational semantics of simple imperative programs, by translation into lambda calculus. Floyd-Hoare logic. Soundness and relative completeness.

5. Brief discussion of more advanced topics at the end of the course.

For PhD students, this course is intended to provide background for further work in this area.
For MS students, this course will count toward specializations in software theory and theoretical computer science.

Calendar

```
   January 1997
 S  M Tu  W Th  F  S         S  M Tu  W Th  F  S         S  M Tu  W Th  F  S

          1  2  3  4                            1                            1
 5  6  7  8  9 10 11         2  3  4  5  6  7  8         2  3  4  5  6  7  8
12 13 14 15 16 17 18         9 10 11 12 13 14 15         9 10  * 12  * 14 15
19 20 21 22 23 24 25        16 17 18 19 20 21 22        16 17 18 19 20 ** 22
26 27 28 29 30 31           23 24 25 26 27 28           23 24 25 26 27 28 29
                                                        30 31

Holidays and Exams:

Jan 20 - M.L. King         Feb 17 - Pres. Day          *  = Dead Week
                                                       ** = Scheduled Final Exam
```