COMPUTER SCIENCE 291
PROGRAMMING LANGUAGE PARADIGMS

I. Introduction

A. Catalog Description

This course introduces the functional, dataflow, and logic programming paradigms, and contrasts them with the imperative paradigm underlying languages like C and Java. Concepts from each new paradigm are emphasized through programming assignments in representative languages. *Prerequisite: CSCI 261.*

B. Objective

Declarative programming languages are an important alternative to languages (such as C, C++, and Java) that use the more familiar imperative programming paradigm. This course introduces the functional, dataflow, and logic programming paradigms in depth through assignments in the programming languages Haskell, Lucid/SISAL, and Prolog. These languages are based on models of computation that are fundamentally different from the von Neumann model underlying imperative programming languages, and exposure to these new paradigms provides valuable perspective on programming and problem solving in general.

C. Prerequisite

CSCI 261

II. Required Topics

At minimum, CSCI 291 should cover the following topics.

A. Context: Imperative vs. Declarative Paradigms

B. Functional Programming (Haskell)

1. Origins and history
2. Lambda Calculus
3. Higher-order functions
4. Lazy, eager, and lenient evaluation
5. Polymorphism
6. Type inference
7. Strict vs. non-strict semantics

C. Dataflow Programming (Lucid, SISAL)

1. Origins and history
2. Streams
3. Filters and transformers

D. Logic Programming (Prolog)

1. Origins and history
2. Facts and rules

3. Unification
4. Search trees and search strategies

E. Parallel complexity (given unbounded processing power)

III. Bibliography

Haskell: The Craft of Functional Programming, Simon Thompson

LUCID, the dataflow programming language, William Wadge and Edward Ashcroft

SISAL Reference Manual, A.P.W. Böhm, R. Oldehoeft, D. Cann, and J. Feo

The Art of Prolog, Leon Sterling and Ehud Shapiro