10/95

COMPUTER SCIENCE 391
PRINCIPLES OF PROGRAMMING LANGUAGES

I. Introduction

A. Catalog Description

The study of programming language concepts and implementation for evaluating existing and future programming languages and their constructs. *Prerequisite: CSCI 281.*

B. Objective

This course will introduce the student to the fundamental criteria guiding programming language design. In addition to study-ing these concepts and their implementations, the student will write programs in three or four programming languages, which are uniquely different from those languages used in the computer science curriculum. This course will also aid those students who are interested in the study of compiler design and construction.

C. Prerequisite

CSCI 281 with a grade of C- or better.

II. Required Topics

At minimum, CSCI 391 should cover the following topics.

A. Language Evaluation Criteria

B. Historical Overview of Imperative Languages

C. Language Classification

1. Imperative vs. functional languages
2. Compiled vs. interpretive languages
3. Sequential execution vs. dataflow

D. Formal Language Syntax Description

1. BNF grammars, Syntax graphs, and parse trees
2. Parsing and derivations
3. Ambiguity, precedence, and associativity

E. Data Types and Variables

1. Keywords, identifiers, and aliases
2. Binding
3. Type checking
4. Scoping rules
5. Constant declarations
6. Primitive data types: booleans, integers, reals, characters
7. Enumerated types, arrays, records, pointers, sets, strings
8. Variable declaration and initialization

II. Required Topics (cont.)

F. Expressions and Assignment Statements

1. Operator precedence and evaluation order
2. Operator overloading and coercion
3. Short-circuit evaluation
4. Assignment operator and multiple targets

G. Control Structures

1. Compound statements
2. Selection statements.
3. Repetition statements
4. Guarded commands

H. Subprograms

1. Procedure and function declarations
2. Parameter passing mechanisms and side effects
3. Subprogram overloading and generic subprograms
4. Separate compilation

I. Data Abstraction

1. Abstract data types
2. Encapsulation
3. Information hiding
4. Modules vs. classes

J. Functional Programming (Lisp)

K. Object-Oriented Programming {Smalltalk}

L. String-Oriented Programming "Icon"

M. Logical Programming (Prolog)

III. Bibliography

Sebesta	Concepts of Programming Languages
Wilson/Clark	Comparative Programming Languages