

## School of Computer Science

**Course Title:** Principles of Programming Languages

**Date:** November 6, 2003

**Course Number:** COP 4555

**Number of Credits:** 3

|  |   |
|--|---|
| <b>Subject Area:</b><br>Foundations  | <b>Subject Area Coordinator:</b> Jai Navlakha<br><br><b>email:</b> navlakha@cis.fiu.edu |
| <b>Catalog Description:</b> A comparative study of several programming languages and paradigms. Emphasis is given to design, evaluation and implementation. Programs are written in a few of the languages.  |   |
| <b>Typical Textbooks:</b><br>John C. Mitchell, <i>Concepts in Programming Languages</i> . (Cambridge, 2003)<br>Adam Brooks Webber, <i>Modern Programming Languages: A Practical Introduction</i> . (Franklin, Beedle & Associates, 2003)<br>Kenneth C. Loudon, <i>Programming Languages: Principles and Practice, 2<sup>nd</sup> edition</i> . (Brooks/Cole, 2003)<br><br>Jeffrey D. Ullman, <i>Elements of ML Programming</i> . (Prentice Hall, 1998) |   |
| <b>References:</b>   |   |
| <b>Prerequisite Courses:</b> COP 3530 (Data Structures)  |   |
| <b>Corequisite Courses:</b> None.  |   |

Type: Required

Prerequisites Topics:

- Familiarity with programming in Java or C++.
- Familiarity with basic techniques of algorithm analysis.
- Familiarity with recursive algorithms.
- Familiarity with linked data structures such as linked lists and binary trees.
- Basic mathematical maturity.

Course Outcomes:

- O1. Master programming a functional language, such as Standard ML.
- O2. Master programming with recursion.
- O3. Be familiar with the use of context-free grammars to specify programming language syntax and with recursive descent parsing.
- O4. Be familiar with natural semantics for imperative and functional programming languages and their use in building interpreters.
- O5. Be familiar with polymorphic type systems and type inference.
- O6. Be familiar with issues in the design and implementation of programming languages, such as lexical versus dynamic scoping and static versus dynamic type checking.

**School of Computer Science**  
**COP 4555**  
**Principles of Programming Languages**

**Outline**

| <b>Topic</b>  | <b>Number of Lecture Hours</b> | <b>Outcome</b>    |
|---|--------------------------------|-------------------|
| 1. <u>Functional Programming in Standard ML</u><br>1.1. Checklist for programming with recursion<br>1.2. First-class functions, currying<br>1.3. Efficiency: using 'let' to avoid recomputation<br>1.4. Polymorphic types, SML97 value restriction<br>1.5. SML datatype                         | <u>18</u>                      | <u>O1, O2, O5</u> |
| 2. <u>Programming Language Syntax</u><br>2.1. Context-free grammars<br>2.2. Parse trees<br>2.3. Ambiguity<br>2.4. Recursive descent parsing   | <u>3</u>                       | <u>O3</u>         |
| 3. <u>Programming Language Semantics</u><br>3.1. Natural Semantics for the <i>Simple Imperative Language</i> and <i>PCF</i><br>3.2. Interpreters<br>3.3. Expressions, commands, declarations<br>3.4. Variables, L-values, R-values<br>3.5. Environments, stack frames, lifetime, tail recursion | <u>13</u>                      | <u>O1, O4, O6</u> |
| 4. <u>Types</u><br>4.1. Dynamic type checking<br>4.2. Static type checking<br>4.3. Type Inference   | <u>6</u>                       | <u>O5</u>         |

**School of Computer Science**  
**COP 4555**  
**Principles of Programming Languages**

**Course Outcomes Emphasized in Laboratory Projects / Assignments**

| <b>Outcome</b> | <b>Number of Weeks</b> |
|----------------|------------------------|
| O1             | 4                      |
| O2             | 2                      |
| O3             | 2                      |
| O4             | 2                      |
| O5             | 2                      |
| O6             | 1                      |

**Oral and Written Communication:**  
No significant coverage

**Social and Ethical Implications of Computing Topics**  
No significant coverage

**School of Computer Science  
COP 4555  
Principles of Programming Languages**

**Approximate number of class hours devoted to fundamental CS topics**

| Topic  | Core Hours | Advanced Hours |
|--|------------|----------------|
| <b>Algorithms:</b>                             |            |                |
| <b>Software Design:</b>                        |            |                |
| <b>Computer Organization and Architecture:</b> |            |                |
| <b>Data Structures:</b>                        |            |                |
| <b>Concepts of Programming Languages:</b>      |            | 3.0            |

**Theoretical Contents**

| Topic                    | Class time |
|--------------------------|------------|
| Formal semantics         | 6 hours    |
| Polymorphic type systems | 6 hours    |

**Problem Analysis Experiences**

1.

**Solution Design Experiences**

1.
2.

**School of Computer Science  
COP 4555  
Principles of Programming Languages**

**The Coverage of Knowledge Units within Computer Science Body of Knowledge<sup>1</sup>**

| <b>Knowledge Unit</b>                     | <b>Topic</b> | <b>Lecture Hours</b> |
|---|--------------|----------------------|
| PL3. Introduction to language translation | 2, 3         | 6                    |
| PL4. Declarations and types               | 1, 4         | 6                    |
| PL5. Abstraction mechanisms               | 1, 3         | 6                    |
| PL6. Functional programming               | 1            | 6                    |
| PL9. Type systems                         | 1, 4         | 6                    |
| PL10. Programming language semantics      | 3            | 5                    |
| PF4. Recursion                            | 1            | 5                    |
|   |              |                      |

---

<sup>1</sup>See <http://www.computer.org/education/cc2001/final/chapter05.htm> for a description of Computer Science Knowledge units