

Wright State University

## CORE Scholar

---

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

---

Fall 2004

### CS 784: Programming Languages

Krishnaprasad Thirunarayan

*Wright State University - Main Campus*, [t.k.prasad@wright.edu](mailto:t.k.prasad@wright.edu)

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs\\_syllabi](https://corescholar.libraries.wright.edu/cecs_syllabi)



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

---

#### Repository Citation

Thirunarayan, K. (2004). CS 784: Programming Languages. .

[https://corescholar.libraries.wright.edu/cecs\\_syllabi/147](https://corescholar.libraries.wright.edu/cecs_syllabi/147)

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# CS 784 Programming Languages

---

- **Instructor:** T. K. Prasad
  - **Phone No.:** (937)-775-5109
  - **Email:** tkprasad@cs.wright.edu
  - **Home Page:** <http://www.cs.wright.edu/~tkprasad>
  - **Quarter:** Fall, ~~2003~~ 2004
  - **Class Hrs:** TTh, 4:10pm-5:25pm, 101 Fawcett
  - **Office Hrs:** TTh, 11:30am-12pm and 3:30pm-4pm, 337 RC (or by appointment)
- 

## Course Objectives

To provide a solid foundation for studying advanced topics in Programming Language Specification and Design.

---

## Prerequisites CS 480/680 Comparative Languages

---

## Course Description

This course introduces concepts related to the specification and design of high-level programming languages. It discusses different programming paradigms, algebraic specification and implementation of data types, and develops interpreters for specifying operationally the various programming language features/constructs. It also introduces attribute grammar formalism and axiomatic semantics briefly. The programming assignments will be coded in Scheme.

---

## Course Load

The course load includes homeworks and programming assignments worth 30 points, a midterm exam worth 30 points, and a final exam worth 40 points.

---

## Text

1. Friedman, Wand and Haynes: Essentials of Programming Languages. 2nd Edition. MIT Press, 2001. ISBN 0-262-06217-8
2. R. Kent Dybvig: The Scheme Programming Language, 3rd Edition. The MIT Press, 2003.

## Reference

1. Guttag, J.V., "Abstract Data Types and the Development of Data Structures," CACM, vol. 20, No. 6, June 1977, pp. 396-404.
2. Chapter 1 of Guttag, J. V., et al, Larch: Languages and Tools for Formal Specification, Springer-Verlag, NY, 1993.
3. H. Abelson and G. J. Sussman, Structure and Interpretation of Computer Programs, 2nd Ed., MIT Press, 1996.
4. Scheme : Language Reference Manual
5. The Teaching About Programming Languages Project
6. Chez Scheme Download Site (<http://www.scheme.com>)
7. DrScheme Download Site (<http://www.drscheme.org/>)

## Grading

The letter grades will be assigned using the following scale: A[90-100], B[80-90), C[70-80), D[60-70), and F[0-60). However, I reserve the right to adjust the scale somewhat to utilize the gaps in the distribution.

## Class Schedule and Syllabus

| Topics with links to Lecture Notes                          | Addl. Readings |
|---|----------------|
| <b>Class 1</b> Evolution of Programming Languages           |                |
| <b>Class 2</b> Scheme Metalanguage                          | Chap 1.1, 1.2  |
| <b>Class 3</b> Abstract Data Types: Algebraic Specs         | Chap 2         |
| <b>Class 4</b> (continue)                                   |                |
| <b>Class 5</b> Programming Paradigms                        |                |
| <b>Class 6</b> Abstract Syntax and its Representation       | Chap 2         |
| <b>Class 7</b> Interpreter for a Simple Expression Language | Chap 3         |
| <b>Class 8</b> User-Defined Functions: Scoping              | Chap 1.3, 3    |
| <b>Class 9</b> Midterm Exam (Oct 6)                         |                |
| <b>Class 10</b> Implementing Recursion                      | Chap 3         |
| <b>Class 11</b> Closures and Streams                        |                |
| <b>Class 12</b> Imperative Programming : Assignment         | Chap 3         |
| <b>Class 13</b> (continue)                                  |                |
| <b>Class 14</b> Interpreter for an Object-Oriented Language | Chap 5         |
| <b>Class 15</b> (continue)                                  |                |
| <b>Class 16</b> Introduction to Attribute Grammars          |                |
| <b>Class 17</b> (continue)                                  |                |
| <b>Class 18</b> Introduction to Axiomatic Semantics         |                |
| <b>Class 19</b> (continue)                                  |                |
| <b>Class 20</b> Wrap-up                                     |                |
| <b>Final Exam</b> (Nov 16, 5:45pm-7:45pm)                   |                |

---

## Old Exams

- Midterm (ps) (pdf).
- Final (ps) (pdf).

---

## Assignments (Fall 2003)

- [Assignment 1](#).
- [Assignment 2](#).
- [Assignment 3](#).

---

*T. K. Prasad (09/07/04 11:06:24 AM)*