Wright State University

CORE Scholar

Computer Science & Engineering Syllabi   College of Engineering & Computer Science

Fall 2009

CS 784: Programming Languages

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

CS 784 Programming Languages

- **Instructor:** T. K. Prasad
- **Phone No.:** (937)-775-5109
- **Email:** t.k.prasad@wright.edu
- **Home Page:** http://www.cs.wright.edu/~tkprasad
- **Quarter:** Fall, 2009
- **Class Hrs:** TTh, 6:05pm-7:20pm, 066 University Hall
- **Office Hrs:** TTh, 5-6pm (395 Joshi) (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.

Reference

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

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. M. Felleisen, R. B. Findler, M. Flatt, and S. Krishnamurthi, How to Design Programs, MIT Press, 2002.
5. Scheme : Language Reference Manual
6. The Teaching About Programming Languages Project
7. Chez Scheme Download Site (http://www.scheme.com)
8. 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 (Chap. from EOPL-2nd ed)
Class 1	Evolution of Programming Languages	Turing Awards
	Why specify? Scheme	

Class 2	Metalanguage	Chap 1.1, 1.2
Class 3	Abstract Data Types: Algebraic Specs	Chap 2
Class 4	(continue)	
Class 5	Programming Paradigms	
Class 6	Abstract Syntax and its Representation	Chap 2
Class 7	Interpreter for a Simple Expression Language	Chap 3
Class 8	User-Defined Functions; Scoping	Chap 1.3, 3
Class 9	**Midterm Exam** (October 8)	
Class 10	Implementing Recursion	Chap 3
Class 11	Closures and Streams	
Class 12	Imperative Programming : Assignment	Chap 3
Class 13	(continue)	
Class 14	Interpreter for an Object-Oriented Language	Chap 5
Class 15	(oopl.ps)	
Class 16	Introduction to Attribute Grammars	Notes
Class 17	(continue)	
Class 18	Introduction to Axiomatic Semantics	
Class 19	(continue)	
Class 20	Wrap-up	
	Final Exam (November 19, 8pm-10pm)	

Exams (Spring 2009)

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

Assignments (Fall 2009)

- Assignment 1 .
- Assignment 2 .
- Assignment 3 . (asg3.ppt)

T. K. Prasad   *(09/10/09 02:24:04 PM)*