

Wright State University

CORE Scholar

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

Fall 2010

CS 480/680: Comparative Languages

Krishnaprasad Thirunarayan

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

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi



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

Repository Citation

Thirunarayan, K. (2010). CS 480/680: Comparative Languages. .

https://corescholar.libraries.wright.edu/cecs_syllabi/259

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.

CS 480/680 Comparative 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, 2010
 - **Class Hrs** : MW, 4:10 - 5:25pm, 054 Rike Hall
 - **Office Hrs** : MW, 3:00 - 4:00pm, 395 JC (or by appt.)
-

Course Description

This course will introduce fundamental concepts and paradigms underlying the design of modern programming languages. For concreteness, we study the details of an object-oriented language (e.g. Java), and a functional language (e.g., Scheme). The overall goal is to enable comparison and evaluation of existing languages. The programming assignments will be coded in Java and in Scheme.

Prerequisites

- Data Structures and Algorithms. (Equivalently, CS400/600.)
 - Experience with programming in imperative languages such as C/C++, Pascal, or Ada.
-

Course Material

1. On-line Lecture Notes.
2. [OOP Basics](#)

References

1. K. Arnold, J. Gosling, and D. Holmes: The Java Programming Language. Addison-Wesley Publishing Co., 4th Edition, 2005. ISBN 0-321-34980-6
2. Michael L. Scott, Programming Language Pragmatics. Morgan Kaufmann Publishers, 2nd Edition, 2006. ISBN 0126339511
3. [The Java Tutorial](#)
4. Ravi Sethi, Programming Languages: Concepts and Constructs. Addison-Wesley Publishing Co., 2nd Edition, 1996. ISBN 0-201-59065-4
5. R. Kent Dybvig, [The Scheme Programming Language](#), 3rd Edition. Prentice Hall, 2003.
6. [Scheme : Language Reference Manual](#)
7. [Chez Scheme Download Site \(http://www.scheme.com\)](http://www.scheme.com)
8. [DrScheme Download Site \(http://www.drscheme.org/\)](http://www.drscheme.org/)
9. [Jython Home Page](#)

10. [Dive into Python](#)

11. [Scala](#)

Relevant Websites

- [The Teaching About Programming Languages Project](#)

Course Load

The course load includes programming assignments worth 30 points, a midterm worth 30 points and a final worth 40 points. Normally, CS680 students are assigned additional homework problems and are expected to solve additional/different problems in the tests.

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. Academic dishonesty will be "rewarded" with a grade of "F". "Sharing/reuse" of solutions to assignment problems is strictly prohibited.

Attendance Policy

All registered students are expected to attend all lectures. In case a student is absent from a lecture due to unavoidable circumstances, the student is still responsible for the material covered in the class, as it is typically available from the course web-page well in advance. Furthermore, the student is expected to find out about in-class announcements from their colleagues/instructor.

Class Schedule and Syllabus

	Topic
Class 1	<u>Evolution of Programming Languages</u>
Class 2	<u>Syntax Specification : Grammars</u>
Class 3	<u>Object-Oriented Programming</u>
Class 4	<u>Symbolic Data; List Processing</u>
Class 5	<u>Styles : Functional vs Procedural</u>
Class 6	<u>Recursive Definitions (Scheme-Startup)(Examples)</u>
Class 7	<u>Abstraction : Higher Order Functions</u>
Class 8	Scoping; Closures
Class 9	Midterm (October 11)
Class 10	<u>Java Design Goals</u>

Class 11	<u>Types, Values, Variables</u>
Class 12	Arrays; Classes
Class 13	<u>Inheritance; Polymorphism</u>
Class 14	Interfaces; Packages; <u>Strings</u>
Class 15	<u>Exceptions</u>
Class 16	<u>Threads</u>
Class 17	(continue) (<u>Scripting vs Systems PL</u>)
Class 18	SCHEME INTERPRETER (2/3 classes)
Class 19	<u>Code (scm/txt)</u>
Class 20	Hand Written Slides (<u>83M pdf</u>) (<u>43M pdf</u>)
Class *	Parameter Passing Mechanisms
Class *	Implementing Subprograms
	Final (November 15, 5:45pm-7:45pm)

Assignments (Fall 2010)

- Assignment 1
 - Assignment 2
-

Exams (Summer 2009)

- Midterm
 - Final
-

T. K. Prasad (08/29/10 03:04:54 PM)

