Fall 2012

CS 3180/5180: 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
https://corescholar.libraries.wright.edu/cecs_syllabi/347

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 corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
School 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, C, C++), a functional language (e.g., Scheme), and get introduced to multiparadigm languages (e.g., Python, Scala). The overall goal is to enable comparison and evaluation of existing languages. The programming assignments will largely be coded in Java and in Scheme, and optionally in Python or Scala.

Prerequisites
- Data Structures and Algorithms (Equivalently, CS3100/5100.)
- Experience with programming in imperative languages such as C/C++, Pascal, or Ada.

Course Material

1. On-line Lecture Notes
2. OOP Basics

References

3. The Java Tutorial
7. Chez Scheme Download Site (http://www.scheme.com)
8. DrScheme Download Site (http://www.drscheme.org)
9. Python 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, graduate 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

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Evolution of Programming Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 2</td>
<td>Syntax Specification : Grammars</td>
</tr>
</tbody>
</table>
Assignments (Fall 2012)

- Assignment 1
- Assignment 2

Exams (Summer 2012)

- Midterm
- Final

T.K. Prasad