Spring 2007

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
https://corescholar.libraries.wright.edu/cecs_syllabi/403

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.


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 5 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 Text and Material

1. On-line Lecture Notes.

References

2. The Java Tutorial
6. Chez Scheme Download Site (http://www.scheme.com)
7. DrScheme Download Site (http://www.drscheme.org/)
8. Jython Home Page
9. Dive into Python

Relevant Websites

- Sun's Java Page
  - Java 5.0 Core APIs
- The Teaching About Programming Languages Project

Download Sites

1. JDK 1.5 (http://java.sun.com/j2se/1.5.0/download.jsp)
2. JBuilder 2005 Enterprise 30-Trial and Foundation is the same download
   (http://www.borland.com/products/downloads/download_jbuilder.html)
3. NetBeans 4.0 (http://www.netbeans.org/community/releases/40/index.html)
5. TextPad Editor (www.textpad.com)
6. WinZip (www.winzip.com)
7. Apache Tomcat (www.apache.org)

Java IDE Tutorials by Y. Daniel Liang

1. Compiling and Running Java from the Command Window
2. Compiling and Running Java from TextPad
3. JBuilder Tutorial
4. NetBeans Tutorial
5. Eclipse Tutorial

Course Load

The course load includes a mix of homeworks and 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

http://www.cs.wright.edu/~tkprasad/courses/cs480/cs480.html
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>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Class 1</strong> Evolution of Programming Languages</td>
</tr>
<tr>
<td><strong>Class 2</strong> Syntax Specification : Grammars</td>
</tr>
<tr>
<td><strong>Class 3</strong> Object-Oriented Programming</td>
</tr>
<tr>
<td><strong>Class 4</strong> Java Design Goals</td>
</tr>
<tr>
<td><strong>Class 5</strong> Types, Values, Variables</td>
</tr>
<tr>
<td><strong>Class 6</strong> Arrays; Classes</td>
</tr>
<tr>
<td><strong>Class 7</strong> Inheritance; Polymorphism</td>
</tr>
<tr>
<td><strong>Class 8</strong> Interfaces; Packages; Strings</td>
</tr>
<tr>
<td><strong>Class 9</strong> Midterm (April 24)</td>
</tr>
<tr>
<td><strong>Class 10</strong> Exceptions</td>
</tr>
<tr>
<td><strong>Class 11</strong> Threads</td>
</tr>
<tr>
<td><strong>Class 12</strong> (continue) (Scripting vs Systems PL)</td>
</tr>
<tr>
<td><strong>Class 13</strong> Symbolic Data; List Processing</td>
</tr>
<tr>
<td><strong>Class 14</strong> Styles : Functional vs Procedural</td>
</tr>
<tr>
<td><strong>Class 15</strong> Recursive Definitions (Scheme-Startup).(Examples)</td>
</tr>
<tr>
<td><strong>Class 16</strong> Abstraction : Higher Order Functions</td>
</tr>
<tr>
<td><strong>Class 17</strong> Scoping; Closures</td>
</tr>
<tr>
<td><strong>Class 18</strong> Scheme Interpreter</td>
</tr>
<tr>
<td><strong>Class 19</strong> (continue)</td>
</tr>
<tr>
<td><strong>Class 20</strong> (continue)</td>
</tr>
<tr>
<td>**Class *** Parameter Passing Mechanisms</td>
</tr>
<tr>
<td>**Class *** Implementing Subprograms</td>
</tr>
<tr>
<td><strong>Finals (June 7?)</strong></td>
</tr>
</tbody>
</table>

**Assignments (Spring 2007)**
• Assignment 1
• Assignment 2

Exams (Spring 2006)

• Midterm
• Final