General Information

Instructor: Matthew Fluet
E-mail: mtf@cs.rit.edu
Office hours: MW 4:00pm – 5:00pm and TR 11:00am – noon (or by apt.); 70-3555

Lectures:
- Section 01 MTWR 2:00pm – 2:50pm; 70-1620
- Section 70 MW 6:00pm – 7:50pm; 70-1620

Home page: www.cs.rit.edu/~mtf/teaching/20091/plc
mycourses.rit.edu

Course Description

A study of the syntax and semantics of a diverse set of high-level programming languages. The languages chosen are compared and contrasted in order to demonstrate general principles of programming language design. The course emphasizes the concepts underpinning modern languages rather than the mastery of particular language details. Programming projects will be required.

Prerequisites

- 4003-334 (Computer Science 4) and 1016-265 (Discrete Math I)
- or permission of instructor

Course Goals

The main goal of this course is to widen a student’s perspective and understanding of programming languages by learning the power of alternative paradigms, thus becoming more critical and self-aware as programmers and software designers.

This goal is achieved in two ways. First, the course exposes students to programming paradigms different from the one that they experienced in the foundational courses. Second, it gives them insight into the theory that underlies many of the mechanisms that, up to now, they have taken for granted. It presents the implementation choices available to language designers and compiler writers, and discusses the range of consequences arising from these choices.
Text Books
Required:
Title: Programming Language Pragmatics (Third Edition)
Author: Michael L. Scott
Publisher: Morgan Kaufmann Publishers
ISBN: 978-0-12-374514-9
Home page: http://www.cs.rochester.edu/u/scott/pragmatics/

Topics

Concepts of Programming Languages (20 hours)

- **Background**
  - why study programming languages
  - history of programming languages
  - language classification
  - language design issues
  - translation including compilers and interpreters
  - basics of language specification: syntax and semantics.

- **Syntax**
  - scanning and parsing
  - Backus-Naur Form (BNF)
  - extended BNF
  - concrete parse tree and abstract syntax
  - scanning and parsing techniques using tools (lex&yacc, . . .)

- **Semantics**
  - informal: attributes; binding (static and dynamic); scope
  - formal: attribute grammars; operational semantics; denotational semantics

- **Data Types**
  - primitive
  - structured
  - abstract
  - type systems

- **Control Structures**
  - sequencing (straight, conditional, iterative, . . .)
  - subroutines (calls, recursive calls, parameters, . . .)
  - other control structures (exceptions, iterators, coroutines, continuations, . . .)

- **Memory management**
  - activation records
  - garbage collection

Language Studies and Comparisons (16 hours)

- exercises in a functional language (Scheme, Lisp, Haskell, ML, . . .)
- exercises in a logic language (Prolog, . . .)
- exercises in a scripting and/or string-processing language (JavaScript, Lua, Perl, Python, . . .)
- exercises in parsing

Note: The order in which topics are discussed in lectures will likely differ from that given above. Furthermore, not all topics will receive equal (or, possibly, any) time. Exams will only cover topics explicitly discussed in lecture or in an assigned reading.

Assignments, Exams, and Grades

Students are expected to attend and participate in class. There will be occasional reading assignments to support in-class dicussions; contributions to these discussions will carry extra weight with the participation grade.

There will a number of programming labs (≥ 4), all of equal weight (regardless of points assigned). Late submission of programming labs will not be accepted, unless special circumstances have been discussed with the instructor in advance of the due date and/or other arrangements have been made.

There will be a mid-term exam (in class, 50min), an end-term exam (in class, 50min), and a final exam; see below for dates.

Students enrolled in 4003-709 are required to prepare (and deliver) a 30 minute oral presentation on a topic in programming languages. Students enrolled in 4003-450 are invited to give a presentation for extra-credit.

Grades will be assigned based on the following grading scheme:

<table>
<thead>
<tr>
<th></th>
<th>4003-450</th>
<th>4003-709</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance &amp; participation:</td>
<td>10.0%</td>
<td>9.0%</td>
</tr>
<tr>
<td>Programming labs:</td>
<td>40.0%</td>
<td>36.0%</td>
</tr>
<tr>
<td>Mid-term exam:</td>
<td>12.5%</td>
<td>11.5%</td>
</tr>
<tr>
<td>End-term exam:</td>
<td>12.5%</td>
<td>11.5%</td>
</tr>
<tr>
<td>Final exam:</td>
<td>25.0%</td>
<td>23.0%</td>
</tr>
<tr>
<td>Oral presentation:</td>
<td>+10.0%</td>
<td>9.0%</td>
</tr>
</tbody>
</table>

Important Dates

- **October 7 (Wed.):** Mid-term exam (in class, 50min)
- **November 9 (Mon.):** End-term exam (in class, 50min)
- **November 16 (Mon.):** Final exam (6:00pm – 8:00pm; 70-1620; Section 70)
- **November 18 (Wed.):** Final exam (8:00am – 10:00am; ??-????; Section 01)

Academic Integrity

As with all courses, the RIT Honor Code and the RIT Academic Honesty Policy apply. See the Department of Computer Science’s statement on academic integrity for more details.

In this course, all submitted work must be your own work (i.e., written up or programmed by you alone) and must include acknowledgments of any collaborators or sources (other than course text books or handouts) used to produce your submission.

You are encouraged to discuss course material with other students. Discussion of programming labs is also allowed, but sharing solutions or code is not allowed.
Disclaimer

I reserve the right to make any changes to the syllabus as I deem necessary throughout the course. Minor changes, such as programming lab due dates, will be announced orally during class and posted on the course mailing list and home page. Major changes, such as grading percentages, will additionally be provided in writing.