CSCI 71010: Programming Languages and Their Implementation

Subash Shankar Fall 2012

Announcements:

- Apr 10: I created a page for the qualifiers (updated on May 29).
- Dec 19: Solutions to HW11 updated (I had some errors in the original posting)
- Now that you have finished the project, you might find this paper interesting, for the obvious reasons.
- Dec 14: Test 2 is on Thursday Dec. 20 at 2-4 in the usual room. I've placed last year's test on blackboard so that you know the format. The content of this year's test is different of course (also, we covered some different material towards the end).
- Dec 14: I think I have updated bb and here with everything I promised to. Please let me know if there is something I forgot.
- Dec 13: Section 12.6 of the 2nd edition is on the reading list, but this material is not in the 3rd edition. I think you don't need it if you followed what we did in class.
- Dec 05: We will stick to the official CUNY schedule and have class on Dec 13 and a 2-hour test on Dec 20.
- Dec 03: I posted an important correction to the project. Please click again.
- Nov 28: Now that you know a few more languages, this history of languages may be interesting.

Archived Announcements

Topics:

Topics to be covered (schedule-permitting) include:

- Programming Paradigms
    - *Imperative (Procedural)
    - Functional (Applicative)
    - Logic
    - Reactive / Synchronous
    - Concurrent
    - *Object-Oriented
- Semantics of Languages
    - Operational Semantics
    - Axiomatic Semantics (Hoare Logics)
    - Denotational Semantics
    - Applications in Software Engineering
- Implementation of Languages
    - Compiler development stages
    - Data structures for compilers
    - Virtual Machines (JVM, WAM)
- Other Topics in Programming Languages
    - Type Systems
    - Software Engineering Applications

Topics preceded with a * indicate prerequisite topics that we will cover very briefly. If you feel that you need a refresher, please leave me email and I can suggest reading material to catch up.

Administrative Details:

Time: Thursdays, 2-4 PM Room: C415A
Recitation: before class (some days, room TBD)
Office Hours:
At Hunter (HN-1000F): Mon 330-530 PM (sharp)
At Graduate Center (4420): immediately after class and/or after CS Colloquium (ask in class)
Phone: (212) 650 3392 (please use email though)
Email: subash<dot>shankar<at>hunter<dot>cuny<dot>edu
(any other email address may not work consistently)
TA (Grader): Allan Zelener (allanzelener<at>gmail<dot>com)

Prerequisites:
This course doesn't require anything beyond what an undergraduate program in computer science does. However, if you feel you need a refresher, the prerequisite topics that are of particular importance are:

- Fluency in some imperative language (e.g., C, Java).
- Data structures (trees, graphs)
- Basic discrete mathematics (propositional and predicate logic, functions and relations)

Textbooks:

1. Kenneth C. Louden and Kenneth A. Lambert, Programming Languages - Principles and Practice, 3rd Edition, ISBN 978-1-111-52941-3, 2012. We will follow the text only for about 40% of the course, and the part we follow has not changed from the 2nd edition of the book (except for numbering). However, please buy the book before the semester starts, as the first few weeks are from the text, and past experience shows that those who delay end up doing very poorly in the course.
2. Krzysztof R.Apt and Ernst-Ruediger Olderog, Verification of Sequential and Concurrent Programs. This is an electronic resource available through the GC library.
3. Haskell textbooks (free online sources):
    1. Primary Textbook: Paul Hudak, John Peterson, and Joseph Fasel, A Gentle Introduction to Haskell
    2. A quick way to get introduced to the HUGS system: Hal Daume III, Yet Another Haskell Tutorial (first 3 chapters)
    3. A quick introduction to functional programming using Haskell: Mertz, Beginning Haskell
    4. If you want extra material, there are other tutorials and a quick overview. Be careful - our main goal is to learn functional programming using Haskell as a tool, while some references concentrate too much on Haskell itself.
4. Prolog textbooks (free online sources):
    1. Primary Textbook: James Lu and Jerud J. Mead Prolog - A Tutorial Introduction
    2. Lots of nice sample programs are available here (if you need examples)
    3. Reintjes, Prolog for Software Engineering, Guidelines on software development using Prolog (optional, if you're interested)

Software:
You will need to install Haskell and Prolog on whichever system you use. I strongly encourage you to use the Hugs and SWI implementations. However, if you really want to use some other implementation, you may at your own risk since your code will be tested/graded on these implementations. For Haskell, some like GHC. In Prolog's case, there are implementations that have little to do with Prolog other than their name, and you should not use those (ask if unsure).

Grading:
Course grades will be based on:

- Homework assignments, including programs. Some programs will be large (i.e., projects), and these will be weighted substantially heavier.
- 2 Midterms: The first one will cover syntax and semantics, and will be in early October. The second one will cover everything else, and be during finals week.

Schedule:

All reading material refers to Louden & Lambert unless stated otherwise. "Lecture" refers to material for which I have no good and reasonably short sources (take good notes!). A&O refers to Apt & Olderog. HPF refers to Hudak, Peterson, and Fasel. LM refers to Lu and Mead, Ren refers to Reintjes, and SWI refers to the SWI Prolog Reference Manual.

Week	Date	Topic	Reading	Lecture Addenda	Homework
1	Aug 30	Introduction	Chap. 1-2 (ed2: 1-3)	Slides on bb	
2	Sep 06	Syntax	Chap. 6 (ed2: 4)		HW 1
3	Sep 13	Formal Semantics - Intro	Chap. 12.1 (ed2:13.1)	Slides on bb	
3	Sep 13	Operational Semantics	Chap. 12.2(ed2:13.2)	Example from class	HW 2
3,4	Sep 13 Sep 20	Denotational Semantics	Chap. 12.3(ed2:13.3)		HW 3
4,5	Sep 20 Sep 27	Axiomatic Semantics	A&O 3.1-3.4 (parts only)	A&O Comments	HW 4
5	Sep 27	Functional Programming	3.1 (ed2: 11.1)	Slides on bb	
6	Oct 04	Introduction to Haskell	Mertz(all) HPF Chap. 1-3		HW 5 (soln. on bb)
7	Oct 11	Haskell Types & Type Classes	HPF Chap. 5-6		HW 6 (soln. on bb)
7	Oct 11	Delayed/Lazy Evaluation (in Haskell)	3.4 (ed2: 11.5) Lecture		
8 9	Oct 18 Oct 25	Lambda Calculus Church Rosser Theorem	3.6 (ed2: 11.8) Lecture		HW 7
8	Oct 18	Midterm - 1 hour (syntax + semantics)			
9	Oct 25	Morals of Functional Programming	Lecture	Slides on bb	
9	Oct 25	Logic Programming	4.1 (ed2: 12.1)	Slides on bb	
10	Nov 08	Intro to Prolog	4 (ed2: 12 LM Chap. 1-4		HW8 (soln. on bb)
11	Nov 15	Prolog Techniques			HW9
11	Nov 15	Prolog: non-declarative and extra-logical parts		Debugging Using SWI	HW 10
12	Nov 29	Negation and Control in Prolog	Lecture		

12	Nov 29	Logic Programming Theory	Lecture		
13	Dec 06	Types	8 (ed2: 6)	Slides on bb	HW 11 (soln. on bb)
13 14	Dec 06 Dec 13	Alternate Logic Programming Languages	Lecture + 11.1,11.8 (ed2:9.1,9.8,12.6)		HW 12
		Software Engineering Applications			
		Overview of Compilers			
		Concurrent Languages			
		Reactive Languages			
	Dec 20	Final Exam (2-4 PM)			

The above schedule is very approximate, and will be changed as we progress through the semester.