

91.301 Organization of Programming Languages, Spring 2008

Syllabus

Contact Information

Prof. Holly Yanco
Office: Olsen 206
Lab: Olsen 304
E-mail: holly@cs.uml.edu (best way to reach me)
Phone: 978-934-3642

Class Meetings

Tuesday and Thursday, 10:00 – 11:15, Olsen 403

Office Hours

Tuesdays 11:30 to 12:30
Wednesdays 10:00 to 12:00
and by appointment.

Course Description

We will study programming languages using Scheme. You'll learn about features of various programming language paradigms including the imperative, functional, logical, and object-oriented approaches. Key concepts include: building abstractions, computational processes, higher-order procedures, compound data, data abstractions, controlling interactions, generic operations, self-describing data, inheritance and message passing, streams and infinite data structures, meta-linguistic abstraction, interpretation of programming languages, machine model, compilation, and embedded languages.

Textbook

Structure and Interpretation of Computer Programs, Second Edition. Harold Abelson and Gerald Jay Sussman with Julie Sussman. MIT Press, 1996

The full text of the book is available on the web at
<http://mitpress.mit.edu/sicp/full-text/book/book.html>

Course Website

<http://www.cs.uml.edu/~holly/91.301>

Software

For the course, we will use Dr. Scheme, which is available as a free download (info about downloading is on the course web site).

Exam Dates

Exam 1: **Tuesday, 11 March 2008**, in class

Exam 2: **Tuesday, 15 April 2008**, in class

Final Exam: To be determined by the Registrar

Grading

Homework	30%
Exam 1	20%
Exam 2	20%
Final Exam	30%

Collaboration Policy

You must do the homework assignments individually. You may discuss the questions with your classmates away from a computer, but you must sit at a computer and program by yourself. To learn, you'll need to actually program in Scheme, not watch another person do it.

Turning in identical (or nearly identical) code violates the collaboration policy. You may not input code into a single file, then turn it in for more than one person. Coding yourself is the only way to learn.

It should go without saying that getting solutions from any website, classmate or old homework assignment is cheating.

Homework Policy

Assignments must be turned in on the date they are due in order to receive full credit. Assignments may be passed in up until the next scheduled class meeting to receive 50% credit. After the next class meeting, no credit will be received for assignments, but you may turn them in to have your code read and commented upon.

Class Schedule

Date	Topic	Reading	Assignment
T 1/29	Course overview Introduction to Scheme	1.1	1 out
Th 1/31	More Scheme intro Substitution model	1.1	
T 2/5	Orders of growth Recursion and iteration	1.2	1 due, 2 out
Th 2/7	Higher-order procedures	1.3	
T 2/12	Compound data Data abstraction	2.1	2 due, 3 out
Th 2/14	Aggregate data: lists	2.2	
T 2/19	Monday schedule: No class		
Th 2/21	Henderson picture language	2.2.4	3 due, 4 out
T 2/26	Aggregate data: trees	2.2	
Th 2/28	Symbolic data	2.3	4 due, 5 out
T 3/4	Multiple representations of data	2.4	
Th 3/6	Generic operators	2.5	5 due
T 3/11	Exam 1, includes material from lectures through 2/28		
Th 3/13	State	3.1	6 out
T 3/18	Spring break		
Th 3/20	Spring break		
T 3/25	Environment model	3.2	
Th 3/27	Object-oriented programming	Handout	6 due, 7 out
T 4/1	Mutable data	3.3	
Th 4/3	More mutation	3.3	7 due, 8 out
T 4/8	Streams	3.5	
Th 4/10	Streams	3.5	8 due
T 4/15	Exam 2, includes material from lectures through 4/3		
Th 4/17	Metacircular evaluator: Overview	4.1	9 out
T 4/22	Metacircular evaluator: Adding expressions and modifying syntax	4.1	
Th 4/24	Metacircular evaluator: Variables and environments	4.1	
T 4/29	Metacircular evaluator: Syntactic analysis	4.1.7	9 due, 10 out
Th 5/1	Lazy evaluation	4.2	
T 5/6	Amb evaluation	4.3	
Th 5/8	Logic programming	4.4	10 due
T 5/13	Memory management and garbage collection	5.3	
TBD	Final Exam		