

91.301 Organization of Programming Languages, Fall 2002 Syllabus

Contact Information

Prof. Holly Yanco
Office: Olsen 220A (for next couple of weeks, then Olsen 206)
Lab: Olsen 304
E-mail: holly@cs.uml.edu
Phone: 978-934-3642

Class Meetings

Tuesday and Thursday, 10:00-11:15, Olsen 415

Office Hours

Office hours will be held in the robotics lab (Olsen 304). I will be in the lab during the following hours.

Tuesdays 2:30 to 4:00
Wednesdays 1:00 to 3:00
Thursdays 3:00 to 4:00

You may also make an appointment with me if you can not make it to the scheduled office hours.

Course Description

We will study programming languages using Scheme. We describe salient semantic 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

Course Website

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

Software

We will be using MIT Scheme for programming in this course.¹ You can download MIT Scheme for free for Unix/Linux and Windows platforms at <http://www.swiss.ai.mit.edu/projects/scheme/>
The software is also available on all machines in the CS labs. (If you find a computer without it, please let me know.)

Exam Dates

Quiz 1: Thursday, 3 October 2002, in class
Quiz 2: Thursday, 7 November 2002, in class
Final Exam: To be determined by the Registrar

Grading

Homework	30%
Quiz 1	20%
Quiz 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.

¹ Although Dr. Scheme has been used in prior semesters, it will not be used this semester because it does not contain a full implementation of the Scheme language.