

University of Illinois at Chicago
Spring 2011

CS 476 / MCS 415 — Programming Language Design Course Syllabus

Room: Grant Hall 205

Time: Tu Th 9:30 - 10:45

URL: via Blackboard, under Engineering: CS - Computer Science: Spring 2011 CS476

Staff

Instructor: Barbara Di Eugenio

Office: 916 SEO

Phone: 312 996-7566

E-mail: bdiugen@uic.edu

Office Hours: TBA

Teaching Assistant: Lin Chen

Office: 940 SEO

TA's email: linchen04@gmail.com

TA's Office Hrs: Tue 2:15-4:15

Course Objectives

The aim of this course is to provide students with the tools necessary for the critical evaluation of existing and future programming languages; and to expose them to programming language paradigms not taught elsewhere in the curriculum. We will do so by examining the process design and features of three of the four main types of programming languages: imperative (C, Fortran, C++, Java); functional (Common Lisp, Scheme, ML, Haskell); logic (Prolog). Students will be exposed to one main programming paradigm which is not usually taught in other classes: functional (mainly via Common Lisp); we will also discuss one general scripting language, Python. If time allows, we will also touch on Ruby (general scripting), and / or Haskell (functional), and / or Erlang (functional).

Note: object-oriented programming is not a focus of this class.

Textbooks

Required: Michael Scott, *Programming Language Pragmatics*, 3rd Edition, Morgan Kaufmann, 2009 (ISBN: 978-0-12-374514-9)

Note for students registered under MCS415: In the bookstore, the book may be available only under CS476, not MCS415.

Highly Recommended book on Lisp: Paul Graham. *ANSI Common Lisp*. Prentice Hall, 1996.

An on-line tutorial on Common Lisp is *Lisp Primer*, <http://mypage.iu.edu/~colallen/lp/>, by Collin Allen and Maneesh Dhagat. Also, <http://www.lisp.org/alu/home>

On-line references on Python can be found at <http://www.python.org/>.

Prerequisites

CS 340 (for CS students), MCS 360 (for MCS students)

Notes

- I use email and the web page message board a lot to communicate with the whole class. Please check your email frequently, especially around deadlines (homeworks and exams). You should use the web page message board to post any questions of general interest, such as clarification on the homework etc. I won't answer general questions by email but on the board.
- The web page will contain all materials relevant to the class, syllabus, assignments, lecture notes, additional papers, etc. You can also see you own grades.

Tentative Schedule

This schedule is subject to change. One or two Ruby tutorials will be interspersed as appropriate through the semester

Dates	Topic	Readings
Week 1	Introduction	Ch. 1
Week 2	Syntax, Semantics	Ch. 2, 4
Week 3	Names, Scope	Ch. 3
Week 4	Data Types, Expressions	Ch. 7
Week 5-7	Functional Programming & Common Lisp	Ch. 10, Lisp materials
Week 8	Logic Programming & Prolog	Ch. 11
Week 9-10	Scripting Languages & Python	Ch.13, Python materials
Week 11	Control Structures	Ch. 6
Week 12-13	Subprograms	Ch. 8
Week 14	Concurrency, Exception Handling	Ch. 12
Week 15	Catch up, Review, etc ...	

Important Dates: Midterms

Date	Exam
2/10 (Th)	Midterm 1
3/17 (Th)	Midterm 2
Finals week (5/2-6)	Final

Grading Criteria

The class will be graded out of 1000 points, distributed as follows:

- **3 or 4 Homework Assignments** (350 points total): Each homework will be worth between 50 and 150 points, depending on difficulty and number of assignments.
- **3 Exams** (650 points total): 2 midterms (first worth 150 points, second worth 200 points), 1 final (300 points).

Important Note: To pass the class you must get at least 60% of the total available points for the three exams.

Letter grades will be decided **only at the end**. However, the following guidelines will be adhered to:

Overall Score (undergraduate)	Overall Score (graduate)	Letter grade
88%	92%	A
78%	82%	B
68%	72%	C

Policies on homeworks and exams

General Policies

1. Late homeworks will not be accepted in any case, unless there is a **documented** personal emergency. Arrangements must be made with the instructor as soon as possible after the emergency arises, preferably before the homework due date.
Advice: If for whatever reason you don't manage to finish an assignment, hand in what you have. Partial credit may be given at the grader's discretion.
2. Statute of Limitations: **Two weeks!** No grading questions or complaints — **no matter how justified** — will be listened to two weeks after the item in question has been returned.

Homeworks

There will be 3 or 4 homeworks. Homeworks will be a mixture of pen and pencil and programming assignments. Homeworks will be handed either in class, or via the facility available under the web page. More details will be available later.

Exams

1. The two midterms will be given during class time; consequently, **no make-ups** will be given.
2. Exams will be closed-book.
3. The final is cumulative, with more emphasis on the last part of the class.

Policy on Academic Integrity

Academic dishonesty will not be tolerated. Please see the CS department policy below on the topic; this policy specifies penalties for violations.

What is academic dishonesty? To hand in any work which is not 100% the student's creation, unless you are explicitly allowed to do so. Thus:

1. **Exams.** All work on all exams must be individually performed.
2. **Homeworks:** no student may give any other student any portion of their solutions or code, through any means. Students are not allowed to help each other debug the code, or to show each other any portions of code or homework.

Important Note: every semester somebody is caught red-handed and as a consequence fails the class. Isn't it better to get a B or a C than an F?

CS department policy on academic dishonesty

The CS Department will not tolerate cheating by its students. The MINIMUM penalty for any student found cheating will be to receive an F for the course and to have the event recorded in a department and/or College record. The maximum penalty will be expulsion from the University.

We intend to devote more effort than in the past to detecting and punishing cheating. Cheating includes all the following, though this is not a complete list:

- Copying or any other form of getting or giving assistance from another student during any test, quiz, exam, midterm, etc.
- Plagiarism—turning in writing that is copied from some other source.
- Obtaining solutions to homework by posting to the Internet for assistance, purchasing assistance, obtaining copies of solutions manuals for instructors, and obtaining copies of previous year's homework solutions.
- Computer programs: Any time you look at another student's code, it is cheating. (Exception: If you are EXPLICITLY told that you may do so by the instructor, for instance, in working on group projects.)

For computer programs, if for some reason we cannot determine who copied from whom, we may, at our discretion, give failing grades to both students.

It is the responsibility of all engineering and computer science professionals to safeguard their company's "trade secrets." An employee who allows trade secrets to be obtained by competitors will almost certainly be fired. So, YOU are responsible for making sure that your Unix directories have permissions set so that only you can read your files, for being sure to log out at the end of working in the computer lab, etc.