SYLLABUS

CSE 452

FALL 2008

Course information:

Title:	Organization of Programming Languages
Lectures:	Tue & Thu 12:40–2:00 pm in 1300 Engineering
Prerequisites:	(CSE 331 or CSE 335) and (CSE 320 or ECE 331)
Web page:	http://www.cse.msu.edu/~cse452
Required text:	*Concepts in Programming Languages,* John C. Mitchell, Cambridge University Press

Catalog description: Organization of programming languages. Type systems. Alternative execution models. Comparison of language features: functional, imperative, logical and object-oriented.

Instructor information:

Instructor:	Laura Dillon	Office:	3132 Engineering
Phone:	517-353-4387	Office hours:	Tue & Wed 2:00–3:30 pm and by appointment
Email:	ldillon@cse.msu.edu		
Web page:	http://www.cse.msu/~ldillon		
Teaching Assist.:	Borzoo Bonakdapour	Office hours:	Mon & Thu 9:00–10:30 am and by appointment in 3353 Engineering
Email:	borzoo@cse.msu.edu		
Web page:	http://www.cse.msu/~borzoo		

Course objectives:

- To gain an appreciation of issues in the design of programming languages. You will learn how language features in different programming paradigms impact productivity of programmers, reliability and efficiency of programs, costs of software development and maintenance, security of software, etc.
- To enable better informed choice of programming language. You will learn about trade-offs that, depending on characteristics of software to be developed and on available resources and such, affect the suitability of various languages for software development.
- To enable better program design and implementation. Most programming language features are motivated by goals of improving program design. Understanding these design goals and how different language features support them helps you use them more effectively.
- To increase knowledge of useful programming techniques. You will learn how to use programming techniques associated with different programming paradigms to solve recurring problems of software construction and development.
- To facilitate learning of new programming languages. Familiarity with a range of programming paradigms and language constructs provides you a solid basis for quickly grasping the essentials of new ones. You will be able to separate syntactic differences from semantic ones. You will have a rich "library" of programming constructs at your disposal for comparison and understanding.

Reading assignments:

The assigned reading for a lecture should be read prior to coming to lecture. Readings from the course textbook will be augmented with readings from other sources. All assigned readings can be found either on the course website or in the reserved reading collection for CSE 452 in the Engineering Library. (See the Readings page and the Schedule on the course website.)

The textbook contains errors. You will therefore want to consult the listing of known errata from: http://theory.stanford.edu/people/jcm/books/cpl-errata.html.

Lectures:

Lectures will focus on the key and/or more complex material in the readings. They will assume you are already familiar with the more routine material from your readings. If you do not understand something from the readings and it is not clarified in lecture, you should ask me about it in class or in office hours.

Attendance at lectures is mandatory. I expect you to attend and engage in class. I will give one or more short in-class exercises during most lectures. These exercises will be "lightly graded".

Homework assignments:

Homework assignments include problems that involve analysis of programming language features and some programming. You will write programs in ML and in Eiffel, at a minimum. Time permitting, you may also do some programming in Prolog and/or some concurrent programming. Assignments will be posted on the course Schedule.

Exams:

The two in-class exams will cover all material assigned in readings and all material covered in lectures or in homework assignments through the Thursday of the week prior to the exam. The final exam will be comprehensive, covering all material assigned in readings and all material covered in lectures or in homework assignments during the semester.

Grading:

I intend to assign your course grade based on your attendance at/engagement in lectures (as measured by the in-class exercises) and your performance on homework assignments and exams. I plan to weight these factors as follows:

In-class exercises—10%; Homework—40%; Exam 1—15%; Exam 2—15%; Final exam—20%

I plan to determine minimum grades using the scale:

4.0	90% & above
3.0	80% & above
2.0	70% & above
1.0	60% & above

While I expect to follow these grading guidelines, I reserve to the right adjust the guidelines if circumstances warrant doing so. All adjustments, if any, will be announced in a timely manner.

Late and missed assignment/exam policy:

The in-class exercises cannot be made up. However, you may miss up to three lectures-worth of in-class exercises without it affecting your course grade.

Homework assignments are due by midnight on their due dates. You are permitted up to 3 late days over the course of the semester. Once your 3 late days are used up, no late assignments will be accepted.

You will be allowed to make up an exams only in case of illness or other reasons over which you have no control and only provided that you notify me in as timely a manner as possible. In such event, you must be prepared to document your illness or the extenuating circumstances that caused you to miss the exam.

Academic integrity:

Article 2.3.3 of the Academic Freedom Report states: "The student shares with the faculty the responsibility for maintaining the integrity of scholarship, grades, and professional standards." In addition, CSE adheres to the policies on academic honesty specified in General Student Regulation 1.0, Protection of Scholarship and Grades; the all-University Policy on Integrity of Scholarship and Grades; and Ordinance 17.00, Examinations. (See Spartan Life: Student Handbook and Resource Guide and/or the MSU Web site.)

Unless I explicitly state otherwise, I expect all solutions to homework assignments, programming assignments, and exams will be solely your own work. On homework assignments, you may collaborate with or receive help from others on an occasional basis as long as all parties contributing or assisting are given explicit credit for their contributions to the homework. In particular, I hope you will help each other in learning the mechanics of how to compile programs in new languages. I will inform you if I believe you are collaborating too much. If in doubt as to what is appropriate, ask me. Uncredited collaborations will be considered a violation of the MSU rules governing academic honesty.

You are expected to develop original work for this course; therefore, you may not submit course work you completed for another course to satisfy the requirements for this course. Also, you are not authorized to use the www.allmsu.com Web site to complete any course work in this course.

Students who violate MSU rules may receive a penalty grade, including but not limited to a failing grade on the assignment or in the course.

Acknowledgements:

I have cherry picked ideas and materials from similar courses taught by others, especially those taught by J. Mitchell and S. Freund.

Tentative lecture schedule:

This schedule is approximate only. It will be adjusted as the course progresses. The actual schedule will appear on the course website. You should check the online schedule regularly.

Week	Tuesday	Thursday	Assignments
#1: Aug 25–29	Course logistics & computability *Mitchell 1 & 2*	Intro to language implementation *Mitchell 4.1*	
#2: Sep 1–5	Intro to lambda calculus *Chpt 4.2*	Functional & imperative languages *Chpt 4.3*	
#3: Sep 8–12	ML:basics *Mitchell 5*	ML: types and polymorphism *Mitchell 6*	
#4: Sep 15–19	ML: Type inference *Mitchell 6*	Stacks & scope *Mitchell 7*	
#5: Sep 22–26	Stacks & scope *Mitchell 7*	Exceptions *Mitchell 8*	
#6: Sep 29–Oct 3	Continuations *Mitchell 8*	**Exam 1**	
#7: Oct 6–10	Garbage collection *Mitchell 3.4*	Modularity & data abstraction *Mitchell 9*	
#8: Oct 13–17	Object-oriented languages *Mitchell 10*	Eiffel: basics *TBA*	
#9: Oct 20–24	Eiffel: Design by Contract *TBA*	Subtyping, inheritance, multiple inheritance *Mitchell 11.7, 12*	
#10: Oct 27–31	Java interfaces & virtual machines *Mitchell 13*	Concurrency *Mitchell 14 & TBA*	
#11: Nov 3–7	Concurrency *TBA*	**Exam 2**	
#12: Nov 10–14	Concurrency *TBA*	Concurrency *TBA*	
#13: Nov 17–21	Logic programming *Mitchell 15*	Logic programming *Mitchell 15*	
#14: Nov 24–28	Logic programming *Mitchell 15*	**NO CLASS**	
#15: Dec 1–5	slack	Review	
Dec 8, 12:40–2:45 p.m.	**FINAL EXAM**		