CS 345 Programming Languages

Meeting time:	MWF 10:00-11:00
Location:	RAS 213
Unique number:	51620
Class Web Page:	http://www.cs.utexas.edu/users/lin/cs345/
Final exam:	Saturday December 15, 2:00pm

	Professor	Teaching Assistant	Admin. Assistant
Name:	Calvin Lin	Erik Reeber	Gem Naivar
Email:	lin@cs.utexas.edu	reeber@cs.utexas.edu	gem@cs.utexas.edu
Office:	ACES 3.424	TBA	ACES 3.422
Office Hours:	TTh 5:00-6:00	MW 11:00-12:30	

Course Objectives. In this course you will learn fundamental concepts and paradigms in programming languages, and you will learn how to evaluate languages across various dimensions. These goals will be accomplished by studying examples from a number of languages, including Miranda, Java and C++. There is a modest programming component to this course, with assignments using either Java or C++.

Text. R. Sethi. *Programming Languages: Concepts and Constructs.* 2^{nd} Edition. Addison-Wesley, 1996.
Additional handouts will be provided to help in learning Miranda and Java.

Supplementary Texts. The following may be useful if you'd like an alternate presentation of the material.
K.C. Louden. *Programming Languages: Principles and Practice.* PWS-Kent Publishing Co., 1993.
David Flanagan, *Java in a Nutshell*, O'Reilly, 1995.

Prerequisites. In addition to the department's official requirements, this course assumes a working knowledge of C++ or Java.

Schedule. We'll cover the following topics, roughly in the order given. (S.n means Chapter n of Sethi.)

1. **Introduction.** Course overview, administrative matters. Motivation: complexity, modularity, abstraction. Programming paradigms. Criteria for evaluating languages. (S.1)
2. **Syntax.** Expression notations. Lexical syntax. Context free grammars. (S.2)
3. BNF syntax notation. Concrete syntax. Other syntax notations. (S.2)
4. Parsing. Ambiguity. Precedence and associativity. (S.2)
5. **Imperative Programming.** Assignments, state, and control flow. (S.3.1)
6. Control abstraction. Syntactic and semantic differences.(S.3.4, 3.7)
7. Introduction to **Java.**
8. **Procedural abstraction.** Parameter passing. Scope rules. (S.5.1-4)

9. Macro expansion, call-by-name. (S.5.3)
10. Implementation of procedures. Activation records. (S.5.5,7)
11. Nonlocal variables. (S.5.7)
12. **Data abstraction.** Data types, type expressions, type checking. (S.6.1-4)
13. Type inference. Type equivalence. Polymorphism and overloading. (S.6.1-4)
14. **Functional programming.** Expressions. Overview of **Miranda.**
15. Evaluating expressions. Evaluation order. Lazy evaluation.
16. Currying. Polymorphism.
17. Infinite data structures. Higher order functions. Inefficiencies.
18. **Object Oriented programming.** Data types, operators, classes, inheritance. (S.7)
19. Absence of pointers. Garbage collection. Dynamic binding. (S.6)
20. C++ vs. Java. Language support for data abstraction. (S.6)
21. Evaluation of Java. Interfaces and multiple inheritance.
22. The big picture: historical trends.

Office hours. You may make an appointment to see me, or you can stop by whenever my office door is open. My door is open most of the time, so **please do not knock if the door is closed.**

Newsgroup. The course has a Unix newsgroup named `utexas.class.cs345.lin`, which is a good place to communicate with classmates regarding homework and programming assignments.

Web page. All announcements and handouts will be placed on the class web page:
`http://www.cs.utexas.edu/user/lin/cs345.`

Computer Accounts. Temporary class accounts on Computer Sciences Department workstations (Sun workstations) are available for students who do not have their own permanent Computer Sciences Department accounts. To sign up for a class account, take your fee bill to the Undergraduate Office (Taylor 2.126).

Attendance. Lectures will often contain material not found in the textbook, and lecture notes will not be provided, so attendance is strongly encouraged.

Paperwork. Submit your assignments to **Gem Naivar** (ACES 3.422), and contact her to obtain any papers that were not picked up in lecture.

Grading

prog. assignments and pop quizzes	15%	approximately 2 assignments
midterm 1	25%	Monday October 1 (subject to change)
midterm 2	25%	Friday November 2 (subject to change)
final	35%	Saturday December 15, 2:00pm-5:00pm

Assignments are due by 5:00pm of the due date. Except by prior arrangement with the instructor, assignments are not accepted after the deadline. System problems, printer failures, and the like are routine occurrences and are not considered cause for extending homework deadlines. To avoid such problems, start early on your assignments.

Requests for corrections to quiz, programming assignment or exam grades must be submitted in writing to the TA within 3 days of the assignment of the grade. Delay in picking up assignments does not extend this deadline. Note that the entire assignment may be regraded, so the score may go either up or down.

Programming assignments: Programs will be graded on clarity as well as correctness, so documentation is encouraged.

Exams: You may bring *one* page of notes (one sheet of 8.5"×11" paper, written on both sides) to both the midterm and the final.

Illnesses and Absences. If extraordinary illness prevents you from handing in an assignment or taking a test, you should contact the instructor immediately to have any chance of receiving special consideration.

Collaboration vs. Cheating. Collaboration is encouraged, but cheating is not. Cheating will lead to failure of the course. It's easy to copy partial or complete solutions to programming assignments, but it's also easy to catch this. Realize that it is your responsibility to not allow others to copy your programs. Allowing others access to your account, either intentionally or unintentionally, is a form of cheating.

Assignment 0: Your first assignment is to send an email message to the professor and the TA. You may say anything you like but you must include your name, and the Subject line should be: "CS345 Assignment 0". Due date: September 5, 2001. These messages will be used to create a class mailing list.

How to Succeed in this Course. There is no single book that contains all of the material in this course. Attend lectures, stay awake in lectures, take notes, and make sure you understand what's said in lectures.

When all else fails, think!

How to Fail in this Course. Ignore the above advice. Assume you can learn everything from the textbook. Start homework assignments late and don't even start reading the programming assignment until three days before they're due.

Whining. Legitimate grading errors will be corrected, but whining will get you nowhere and only serves to irritate the professor.