

# Syllabus

## Fall Term 2007

(revised 2008 to eliminate individual oral presentation and report, and reflect new University Attendance Policy)

- **Instructor Contact Information**
- **Meetings**
- **Catalog Description**
- **Prerequisites**
- **Objectives**
- **Textbook & Other References**
- **Assignments & Examinations**
- **Grading**
- **Academic Honor Policy**
- **Accommodation for Disabilities**
- **Communication**
- **Changes**

---

## Instructor Contact Information

---

Instructor: **Ted Baker**

- course URL: <http://www.cs.fsu.edu/~baker/pls>
- Office hours: Monday, Wednesday, Friday 9AM - 10AM, and by appointment
- Office location: 207A LOV
- Phone: 644-5452
- e-mail: [baker@cs.fsu.edu](mailto:baker@cs.fsu.edu)

Teaching Assistant: Josh Lawrence

- Office hours: Monday and Tuesday morning, times TBD
- Office location: 207 LOV
- Office Phone: 644-3421
- e-mail: [lawrence@cs.fsu.edu](mailto:lawrence@cs.fsu.edu)

---

## Meetings

---

Meeting	Day(s)	Time of Day		Location
combined lecture	MWF	8:00-8:50	AM	201 MCH
sect. 1	R	12:30	PM	315A MCH
sect. 2	R	2:00	PM	315A MCH
sect. 3	R	11:00	AM	202 MCH

"MCH" is Carothers Hall. For a campus map showing the location, follow [this link](#).

---

## Catalog Description

---

"A survey of programming languages and language features and an introduction to compilers.

Languages to be discussed include Fortran, Pascal, Ada, PL/1, APL, and Lisp."

---

## Prerequisites

---

- COP 4530 (Data Structures, Algorithms, and Generic Programming)
- implied by COP 4530:
  - CDA 3100 (Computer Organization I)
  - MAD 2104 or 3107 (Discrete Mathematics)
  - COP 3330 (Object Oriented Programming with C++)

Students coming into the course should already have a good working knowledge of:

- Fundamental UNIX programming tools, including a command shell, a text editor, the **Make** utility, e-mail, printing, and debugging.
- The full the C++ programming language, as a common foundation and point of departure for the study of other programming languages.
- Experience implementing data structures using array indexing, pointers, and hashing.
- Machine code instructions, memory addressing, and computer hardware concepts, at the level of CDA 3100.
- Basic formal logic and methods of proof.

---

## Objectives

---

The principal goal of this course is to provide you with general knowledge of programming language design principles, features, and implementation techniques, so that you will be better able to choose among programming languages, learn a new programming languages more quickly, and make more effective use of any given programming language. A secondary goal is provide a basis for further studies in programming language theory and/or implementation, should you choose to pursue the subject further.

In addition, this course serves as one of the "capstones" of the computer science curriculum. As such, it will require you to demonstrate that you are able to apply a collection of the general skills you should have learned over the course of the bachelor's degree program. These skills include problem solving, programming, the use of formal mathematical techniques, and written and oral communication.

In order to pass this course, you will need to demonstrate that you have learned to:

- Interpret and write programming language syntax descriptions expressed using BNF (context-free grammars).
- Interpret and complete axiomatic proofs of program properties based on weakest preconditions and strongest postconditions.
- Classify and compare programming constructs in imperative, functional, and logic programming languages.
- Describe how scoping rules are designed to ensure type safety based on binding time and object lifetime.
- Describe and explain how programming constructs are implemented, in terms of memory layout and machine instructions.
- Implement a tiny programming language using recursive descent parsing, symbol table, and translation or interpretive execution.
- Learn to program in one or more new languages, with little in-class instruction, based on general knowledge of programming language principles and analogy to other languages.

---

## Topics

---

The topics covered in this course will include the following:

- Compilers and interpreters
- Syntax specification techniques, including BNF
- Semantic specification techniques, including axiomatic
- Names, scopes, and bindings
- Control flow
- Subroutines and parameter passing
- Exception handling
- Functional programming, including a functional language, Lisp or Scheme
- Logic programming, including a logic programming language, Prolog
- A strongly typed imperative language with nested scopes, Ada

---

## Textbook & Other References

---

Author: Michael Scott

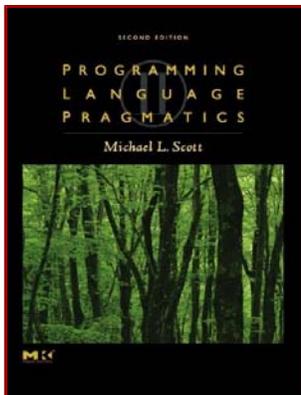
Title: Programming Language Pragmatics (Second Edition, 2006)

Publisher: Morgan Kaufmann

paperback: ISBN 13:078-0-12-633951-2, ISBN 10:0-12-633951-1

Table of Contents:

1. Introduction
2. Programming Language Syntax
3. Names, Scopes, and Bindings
4. Semantic Analysis
5. Target Machine Architecture
6. Control Flow
7. Data Types
8. Subroutines and Control Abstraction
9. Functional Languages
10. Logic Languages
11. Concurrency (\*)
12. Scripting Languages
13. Building a Runnable Program
14. Code Improvement (\*)



Chapters indicated by (\*) will not be covered in this course.

You can follow the link to <http://www.cs.rochester.edu/~scott/pragmatics/> for on-line support materials provided by the author.

Various other reading materials will be provided on the Web as the semester progresses. These include tutorials on specific programming languages, programming examples, and notes on material covered in class. These materials will be accessible from the **Notes** Web page.

---

## Assignments & Examinations

---

**Readings.** You will be assigned readings in the textbook and on the Web. For the textbook readings, you should try to answer the review questions and attempt as many as you can of the exercises at the end of each chapter, to help you find out whether you have understood the material you have read.

### **Class Attendance & Participation.**

Class attendance is required. This includes both lecture and recitation meetings. Attendance will be checked at most class meetings. If you are not present when attendance is taken or you are not in your assigned seat for a lecture meeting, you will be considered absent. Each unexcused recorded absence will result in a reduction of the class participation grade. (See **Grading** below.) The first three (3) absences (or latenesses) will be excused without justification; after that, absences will only be excused for the reasons listed in the University Attendance Policy, which are "documented illness, deaths in the immediate family and other documented crises, call to active military duty or jury duty, religious holy days, and official University activities". Consideration will also be given to "students whose dependent children experience serious illness".

**Examinations.** There will be one midterm examination and a final examination. They are the main check on what you have learned from the course. You cannot pass the course without a passing average score on the examinations. (See **Grading** below.) All examinations will be "closed book". That is, no books or reference materials will be allowed in the examination room. All students will be expected to take the examinations at the scheduled time, shown on the course **Calendar**.

**Examinations from previous offerings of this course** are posted on the Web.

If you know in advance of a reason you will not be able to take a scheduled examination within the scheduled window of time, you are responsible for making arrangements in advance with the instructor. If you make prior arrangements, or have a documented last-minute emergency conflict -- such as a medical emergency or your employer requires you to make an out-of-town trip -- the instructor will attempt to accommodate you. However, under no circumstances will a student be allowed to take an examination before the rest of the class. A make-up examination may have an entirely different format from the examination given in class, and may be an oral examination. If a student misses a midterm examinations and chooses not to take a makeup examination, the final grade for the course will be computed using the final examination in place of the missed midterm examination.

**Homework & Programming Assignments.** There will be several programming and homework assignments. You are expected to work individually on these.

The objective of the programming assignments is to give you practice with the new languages, ideas, and techniques you are reading and hearing about in class, and at the same time improve your programming skills. Details on each assignment will be posted on the **Assignment** Web page as the term progresses.

Homework assignments consist of short-answer questions, essays, or problems. They are intended to push you to read, and to discover whether you really understand some the concepts covered in the readings and classes, before you get to the examinations. Doing them should be a learning process. Do them yourself. If you find and copy an answer that someone else figured out that is not just a violation of the Honor Code. You will have missed out on the learning experience.

It seems that every term some students wait to start an assignment until just before it is due, discover it takes much longer than expected, and then fall hopelessly behind in the course. Please try not to fall into this trap.

To do your programming assignments, you will use a collection of networked Linux systems provided by the CS Department, accessible as "linprog.cs.fsu.edu". You should be already familiar with these systems from the prerequisite courses. If you use another computer to work on an assignment, you are responsible for checking that it will run correctly the "linprog" systems, before you turn it in.

---

## **Grading**

---

You will learn better if you concentrate on learning, and do not worry about the details of grading policy. However, since some students want to know, the following details are provided.

Your final grade should reflect how much you learned from the course. You should be learning right up to the last minute, so the last word on what you have learned is your performance on the final examination. On the other hand, there are several good reasons to collect and score work throughout the term, and for all your work to play a role in determining your final grade. One reason is to motivate you to study every week, and to put in your best effort on all the programming exercises. Another reason is for you to find out about any gaps in your learning before the end of the term, so you can fill them in.

In order for you to learn from your errors without losing motivation, the grading policies incorporate a certain amount of forgiveness. The midterm exam score will be thrown out if the final exam score is higher. All homework assignment scores will be thrown out if the exam average is higher. For programming assignments there will ordinarily be no forgiveness, since you should have adequate opportunity to learn from your errors and revise your program during the development and debugging process. However, the instructor does reserve the discretion to exercise forgiveness, *ad hoc*, for weak grades on homework and programming work done early in the course if there is solid evidence that the student has mastered the material by the end.

**Do not expect to turn in work late.**

Whether to accept any late turn-ins of assignments will be entirely at the discretion of the person (instructor or teaching assistant) who grades the work. If a late assignment is accepted, you should expect that score will be reduced. The amount of the reduction will be entirely at the discretion of the grader, and may depend on how late the assignment is. In any case, no work will be accepted for grading after a solution is discussed or handed out, either in the class or on the Web.

Except for a few special rules listed below, the final course grade will be based on the average of the following data, with the indicated weights:

Graded work	Weight
Midterm exam	20%
Final exam	25%
Programming assignments	40%
Homework assignments	10%
Other class participation	5%

Each of the above will be assigned a score in the range 0 to 100, and averaged to compute your final grade for the term, using the following table:

Percent	Letter	Percent	Letter	Percent	Letter	Percent	Letter
91-100	A [4.00]	81-84	B [3.00]	71-74	C [2.00]	61-63	D [1.00]
89-90	A- [3.75]	79-80	B- [2.75]	69-70	C- [1.75]	59-60	D- [0.75]
85-88	B+ [3.25]	75-78	C+ [2.25]	64-68	D+ [1.25]	0- 58	F [0.00]

**Special Rules and Other Notes:**

1. If either the Programming or the Examination average (considered individually) is below C-, the final grade for the course will not be higher than D+.

### Watch out for this rule!

In the past, some students have complained at the end of the semester when it caused them to fail the course. There is a reason for this rule. A grade of D+ or lower in either of these two areas indicates failure on an essential part of the the course. The knowledge and skills covered by the examinations and the programming assignments are both essential, and do not overlap much. Within the time and equipment limitations of an examination one cannot adequately test programming skills, and there is not time in the semester to assign a program to cover every important concept of the course. It is also true that if a student violates the Academic Honor Policy the grades on homework may not accurately reflect the real knowledge and skills of the individual. For all these reasons, completely unsatisfactory performance in either one of two these areas cannot be outweighed by better performance in the other.

2. Each **unexcused absence** will reduce your class participation grade by 10%.

All graded work will be returned after grading. Students are responsible for picking up the graded copies of their work. Work that is not picked up may be discarded by the instructor one week after the start of the next term. The one exception is the final examination. The University requires that the original copies of all final examinations be retained in the Department for two years.

As soon as the midterm exam is graded, the entire examination, sample answers, and distribution of scores will be posted on the Web. Whether to post a solution to a homework or programming assignment, and when, will be left to the discretion of the grader for that assignment.

---

## Academic Honor Policy

---

The Florida State University Academic Honor Policy outlines the University's expectations for the integrity of students' academic work, the procedures for resolving alleged violations of those expectations, and the rights and responsibilities of students and faculty members throughout the process. Students are responsible for reading the Academic Honor Policy and for living up to their pledge to "...be honest and truthful and ... [to] strive for personal and institutional integrity at Florida State University." (Florida State University Academic Honor Policy, found at <http://dof.fsu.edu/honorpolicy.htm>.)

Take note that this policy not only puts you on your honor not to cheat. It requires that you report any cheating that you observe. In this course, the academic penalty for first violation will be a zero on the assignment or examination where the violation occurred, and the penalty for repeat violations will result in a grade of "F" in the course. Please read the detailed discussion of how the Honor Code applies to programming assignments for this course, at <http://www.cs.fsu.edu/~baker/plagiarism.html>.

---

## Americans with Disabilities Act

---

Students with disabilities needing academic accommodations should:

1. Register with and provide documentation to the Student Disability Resource Center (SDRC).
2. Bring a letter to the Instructor from the SDRC indicating the need for accommodatoin and what type. This should be done during the first week of class.

*This syllabus and other class materials are available in alternative format upon request.*

For more information about services available to FSU students with disabilities, contact the Assistant Dean of Students at:

### Student Disability Resource Center

97 Woodward Ave South  
108 Student Services Building

Florida State University  
Tallahassee, FL 32306-4167

e-mail: [sdrc@admin.fsu.edu](mailto:sdrc@admin.fsu.edu)

voice: (850) 644-9566

TDD: (850) 644-8504

<http://www.disabilitycenter.fsu.edu/>

---

## Communication

---

If you are experiencing difficulty or are concerned about your progress, please contact the instructor right away. Problems are usually easier to solve when they are addressed early.

You are required to verify that your e-mail address is included in the class e-mail list. The Instructor will send everyone a message at the start of the term. If you don't get that first message, make sure you are added to the list.

Check regularly for electronic mail sent to you containing information about this course. You are also encouraged to use e-mail to ask questions and report problems. For this course, we will be using only official University ("@fsu.edu") or CS Department ("@cs.fsu.edu") e-mail accounts.

So that you receive all e-mail promptly, if you ordinarily read your e-mail using another account, you should set up automatic e-mail forwarding from your "@cs.fsu.edu" account and your "@fsu.edu" accounts to that other account.

This syllabus and other information, such as course announcements, assignments, lecture notes, and some useful links to other Web sites, will be posted on the Web site for the course, which is located at <http://www.cs.fsu.edu/~baker/pls>. Access to most of the materials on that site will require a special login name and password, which will be provided to you by the instructor in class or by e-mail.

---

## Changes to the Syllabus

---

This syllabus may be changed during the term, with notice by e-mail and in class.

© 2007-2008 **T. P. Baker**