

Fayetteville State University
College of Arts and Sciences
Department of Mathematics and Computer Science
CSC 207.D1 – Symbolic Programming
Fall 2009

I. Locator Information

Instructor: Dr. Michael Almeida

Course # and Name: CSC 207.D1 – Symbolic Programming

Day and Time Class Meets: online

Room: online

Semester Credit Hours: 3

Office Hours: M 2:00–5:00; W 12:00–2:00

TR 11:00 – 12:30 & by appointment

Total Contact Hours for Class: 45

Office Location: 340 SBE

Email address: malmeida@uncfsu.edu

Office Phone: 672-1161

FSU Policy on Electronic Mail: Fayetteville State University provides to each student, free of charge, an electronic mail account (username@uncfsu.edu) that is easily accessible via the Internet. The university has established FSU email as the primary mode of correspondence between university officials and enrolled students. Inquiries and requests from students pertaining to academic records, grades, bills, financial aid, and other matters of a confidential nature must be submitted via FSU email. Inquiries or requests from personal email accounts are not assured a response. The university maintains open-use computer laboratories throughout the campus that can be used to access electronic mail.

Rules and regulations governing the use of FSU email may be found at

<http://www.uncfsu.edu/PDFs/EmailPolicyFinal.pdf>

II. Course Description

This course introduces the basic concepts and methods of symbolic programming. Symbolic programming involves the construction and analysis of complex symbolic expressions which can be used to represent many different types of information, including mathematical formulas, objects and their relations, and natural language sentences. This course also introduces functional programming and logic programming as two widely used paradigms for symbolic computation. Course topics include recursion, list processing, tree processing, backtracking, unification and resolution.

Prerequisites: Grade of “C” or better in CSC 130 and MATH 150

III. Disabled Student Services

In accordance with Section 504 of the 1973 Rehabilitation Act and the Americans with Disabilities Act (ACA) of 1990, if you have a disability or think you have a disability to please contact the Center for Personal Development in the Spaulding Building, Room 155 (1st Floor); 910-672-1203.

IV. Textbooks

In this course the Blackboard notes are primary while the textbooks are only supplementary!

Dybvig, R. Kent (2003) *The Scheme Programming Language, 3rd Edition*. MIT Press.

A free electronic version of this textbook is available online at: www.scheme.com/tspl3/

Bratko, Ivan (2001) *Prolog Programming for Artificial Intelligence, 3rd Edition*. Addison-Wesley.
(recommended, but any other Prolog book will do)

Scheme, a dialect of Lisp, will be used as the functional programming language, and Prolog will be used as the logic programming language. Implementations of these programming languages are available in the CS labs in SBE and online.

PLT Scheme can be downloaded free at: <http://download.plt-scheme.org/drscheme/>

MIT/GNU Scheme can be downloaded free at: www.gnu.org/software/mit-scheme/ (the IDE isn't as good as that of PLT Scheme but it's ok)

SWI-Prolog can be downloaded free at: www.swi-prolog.org/ (be sure to select **.pro** as the extension when installing)

V. Student Learning Outcomes

Upon completion of this course, students will be able to:

1. design and implement programs in the Scheme programming language.
2. design and implement programs in the Prolog programming language.
3. write procedures that construct and manipulate symbolic expressions.
4. write recursive procedures that perform numerical and list operations.
6. write tail recursive procedures.
7. write programs in the functional programming style.
8. implement recursive rules in Prolog.
9. write programs in the logic programming style.

VI. Course Requirements and Evaluation Criteria

a. Final grades are assigned as follows:

A: 90-100

B: 80-89

C: 70-79

D: 60-69

F: < 60

b. Attendance Requirements – Students are expected to keep up with the class schedule.

c. The course grade is based on: seven homework assignments (9% each), a midterm exam (17%) and a final exam (20%).

d. Policy on Late Assignments - Each assignment must be submitted on time. Up to ten (10) percent of your assignment mark will be deducted for each school day the assignment is overdue. Contact me before the deadline if you won't be able to submit something on time.

e. Dishonesty on graded assignments will not be tolerated. Although students may discuss assignments with one another, they must neither give nor receive excessive help. Students learn by doing things themselves. Having access to another student's work on the system is definitely not allowed. Duplicate answers are not acceptable. Each student is responsible for disposing of printouts safely (Do NOT simply throw away printouts in a trash can where they can easily be retrieved by another person.) and for protecting their home directory. All students involved in dishonesty (those giving as well as those receiving unallowable help) will be penalized.

Please note: If these evaluation criteria must be revised because of extraordinary circumstances, the instructor will distribute a written amendment to the syllabus.

VII. Academic Support Resources – none for this course.

VIII. Course Outline and Assignment Schedule*

week #	start date	readings	assignments	events
1	17-Aug	Scheme Ch.1 notes; Notes on running Scheme	download & install Scheme	Classes start 8/20
2	24-Aug	Ch.2 Secs 1-2 notes; Ch.2 Secs 3-4 notes	Exercise #1 assigned	

3	31-Aug	Ch.2 Secs 5-6 notes; Notes on list construction and manipulation	Exercise #2 assigned	
4	7-Sep	Ch.2 Sec 7 notes; Functional programming notes (part 1)		Monday holiday 9/7
5	14-Sep	Intro recursion notes; Recursion notes on numerical functions	Exercise #3 assigned	Fall Convocation 9/15 2:15 – 3:30 pm
6	21-Sep	Symbolic programming notes (part 1); Notes on tracing in Scheme		
7	28-Sep		Midterm Exam assigned	
8	5-Oct	Recursion notes on simple list functions	Midterm Exam due; Exercise #4 assigned	Midterms start 10/8
9	12-Oct	Recursion notes on list constructing functions		Fall Break 10/15-16
10	19-Oct	Prolog Ch.1 notes; Notes on running Prolog	Download & install prolog; Exercise #5 assigned	
11	26-Oct	Ch.2 notes; Notes on tracing in Prolog		
12	2-Nov	Ch.3 Sec.4 (arithmetic) notes	Exercise #6 assigned	
13	9-Nov	Ch.3 Secs.1-2 (lists) notes		Holiday 11/11
14	16-Nov	Ch.5 notes	Exercise #7 assigned	
15	23-Nov	Notes on resolution		Holiday 11/26-27
16	30-Nov		Final Exam assigned	Classes end 12/4
17	7-Dec		Final Exam due	

* This schedule is subject to change for the optimum benefit of the class as a whole. Therefore it is important to stay alert and attend class regularly.

IX. Teaching Strategies

The primary teaching strategies for this course will be lectures notes, supplementary textbook readings and programming exercises.

X. Bibliography

Bramer, Max. *Logic Programming with Prolog*. Springer, 2005.

Clocksin, William F. & Christopher S. Mellish. *Programming in Prolog, 5th Edition*. Springer, 2003.

Friedman, Daniel P. & Felleisen, Matthias. *The Little Schemer, 4th Edition*. MIT Press, 1996.

Harvey, Brian & Wright, Matthew. *Simply Scheme*. MIT Press, 1994.