The University of Georgia

Department of Computer Science

Course Information Sheet

CSCI 4540

Symbolic Programming

Brief Course Description

Programming in LISP and PROLOG, with emphasis on artificial intelligence techniques. Other languages used for artificial intelligence work will be presented more briefly.

Extended Course Description / Comments

This course is cross-listed with ARTI 4540.

Pre-Requisites and/or Co-Requisites

CSCI 1302 (pre-requisite), or POD
Software Development in Java

CSCI(PHIL) 4550 (Co-requisite)
Introduction to Artificial Intelligence

Approved Textbooks
(If more than one, course text used during a semester is at the discretion of the instructor)

1) *PROLOG Programming in Depth*, (1996 edition) by Covington, Nute, and Vellino; Coverage: Chapters 1-9, (and parts of 10-12, time permitting). 0-13-138645-X

Specific Learning Outcomes (Performance Indicators)

1-Outcome: *Language structure; program development and execution.* By the end of the semester, students should be able to use a standard development environment to write and execute Prolog and Lisp programs. They should be able to distinguish correct from incorrect syntax. They should be able to use the console to successfully interact with programs.

Sample homework and programming assignments:

1) Create a small knowledge base encoding a family tree.
2) Given the knowledge base, query it to determine a given person's great-grandmother.
3) Extend the family knowledge base with additional relationships and process several correct queries.
4) Given the definition of predicate or function *mystery*, describe its behavior.

2-Outcome: *Recursion*. By the end of the semester, students should be able to program non-recursive, recursive, and tail-recursive versions of routines. Students should be able to follow debug traces of recursive programs.

Sample homework and programming assignments:

5) Write non-recursive, recursive, and tail-recursive versions of a routine to compute the nth Fibonacci number.
6) Trace the *member/2* predicate using the query *member(b, [a, b, c])*.

3-Outcome: *Search tree pruning*. By the end of the semester, students should be able to follow tree traversals, anticipate backtrack points, and analyze multiple search tree paths using backtracking. Students should understand the effects of cuts in programs and be able to effectively apply them to limit search within a search tree.

Sample homework and programming assignments:

7) Implement a Prolog program to enumerate solutions to a Kakuro puzzle involving dozens or hundreds of variables.
8) Identify the size of the puzzle's search space.
9) Develop strategies to manage the exponential growth of the search tree for the Kakuro puzzle.
10) Identify how different strategies affect the memory and time required to obtain a solution.
11) Compare and contrast the number of logical inferences involved in programs with and without cuts.
12) Modify your Kakuro puzzle solver using cuts to improve its efficiency.

4-Outcome: *Metaprogramming.* By the end of the semester, students should be able to write programs that manipulate themselves—that is, students should be able to write programs that learn, evolve, reason, etc.

Sample homework and programming assignments:

13) Implement a forward or backward-chaining inference engine, together with associated routines, suitable for use in expert systems.
14) Use "assert" and "retract" predicates to implement a dynamic programming algorithm.
15) Modify the "located_in" program to have the capability to "learn" new geographical facts while executing user queries.

Relationship Between Course Outcomes and Learning Outcomes

		Program Outcomes										
		a	b	c	d	e	f	g	h	i	j	k
Learning Outcomes	1	•	•	•						•	•	•
	2	•	•	•						•	•	•
	3	•	•	•	•		•			•	•	•
	4	•	•	•						•	•	•
	5											
	6											
	7											
	8											
	9											
	10											
	11											

Program Outcomes

(These are ABET-specified and should not be changed)

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
d. An ability to function effectively on teams to accomplish a common goal.
e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
f. An ability to communicate effectively with a range of audiences.
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
h. Recognition of the need for and an ability to engage in continuing professional development.
i. An ability to use current techniques, skills, and tools necessary for computing practice.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k. An ability to apply design and development principles in the construction of software systems of varying complexity.

Major Topics Covered
(Approximate Course Hours)

3 credit hours = 37.5 contact hours
4 credit hours = 50 contact hours

Note: Exams count as a major topic covered

Propositional logic (3)
Predicate logic (3)
Programming in logic (2)
Prolog rule base dynamics (3)
List data structure (3)
Recursion is our friend (3)
Procedural algorithms (3)
Data input and output (1)
Languages and metalanguages (2)
Searching and sorting (2)
AI techniques for solving problems (1)
Expert systems (2)
Expert system shells (2)
LISP programming (2)
Functions (2)
Recursion and Iteration (2)
Looping and other special constructs (2)

Assessment Plan for this Course

Each time this course is offered, the class is initially informed of the Course Outcomes listed in this document, and they are included in the syllabus. At the end of the semester, an anonymous survey is administered to the class where each student is asked to rate how well the outcome was achieved. The choices provided use a 5-point Likert scale containing the following options: Strongly agree, Agree, Neither agree or disagree, disagree, and strongly disagree. The results of the anonymous survey are tabulated and returned to the instructor.

The course instructor takes the results of the survey, combined with sample student responses to homework and final exam questions corresponding to course outcomes, and reports these results to the ABET committee. If necessary, the instructor also writes a recommendation to the ABET committee for better achieving the course outcomes the next time the course is offered.

How Data is Used to Assess Program Outcomes

Each course Learning Outcome, listed above, directly supports one or more of the Program Outcomes, as is listed in "Relationships between Learning Outcomes and Program Outcomes". For ARTI/CSCI-4540, Program Outcomes (a)-(d), (f), and (i)-(k) are supported.

Course Master

Dr. Don Potter (updated 6/28/2013)

Course History