

CS480 - Artificial Intelligence

Last Updated - 02/07/02

Course Manager - Dr. Peter Greene, Research Professor of Computer Science

3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description - Styles of programming and software engineering with applications to artificial intelligence and to the creation of good programming environments through the use of key ingredients of these styles. These include techniques of search, data-driven programming, demons, frames, object-oriented programming, production-rule systems, logic programming, and code that constructs code; including language-extension through macros. Prerequisite: CS 331. (3-0-3)

Textbook

- Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall Publishers, 1st Edition, ©1995, ISBN-0131038052

References - other textbooks or materials

- LISP References - textbook WWW page <http://www.cs.berkeley.edu/~russell/aima.html>

Course Goals - Students should be able to:

- Describe the Turing test.
- Explain the concepts of optimal reasoning, human-like reasoning, optimal behavior, human-like behavior.
- Develop "PAGE" descriptions of an agents and determine which agent type is applicable to a problem.
- Solve problems in a functional programming language (LISP)
- Formulate an efficient problem space for a problem expressed in English by expressing that problem space in terms of states, operators, an initial state, and a description of a goal state.
- Describe the problem of combinatorial explosion and its consequences.
- Select an appropriate brute-force search algorithm for a problem, implement it, and characterize its time and space complexities.
- Select an appropriate heuristic search algorithm for a problem and implement it by designing the necessary heuristic evaluation function.
- Describe under what conditions heuristic algorithms guarantee optimal solution.
- Implement minimax search with alpha-beta pruning for some two-player game.
- Formulate a problem specified in English as a constraint-satisfaction problem and implement it using a chronological backtracking algorithm.
- Explain the operation of the resolution technique for theorem proving.
- Apply Bayes theorem to determine conditional probabilities.
- Explain the distinction between monotonic and non-monotonic inference.
- Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.
- Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning.
- Determine which of the three learning styles is appropriate to a particular problem domain.
- Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks. Explain the nearest neighbor algorithm and its place within learning theory.

Prerequisites by Topic

- Programming including recursion
- Discrete mathematics and data structures
- Basic analysis of algorithms

Major Topics Covered in Course

1. Introduction, History of AI, Intelligent agents	3 hours
2. Functional Programming (LISP)	7.5 hours
3. Uninformed search, Informed search, Constraint satisfaction, Game-playing	12 hours
4. Logical agents, Propositional logic, First-order logic, Inference in first-order logic	4.5 hours
5. Uncertainty, Probability, Belief networks, Belief network inference, Optimal decisions under uncertainty, Optimal sequential decisions	10.5 hours
6. Learning, Neural networks, Bayesian learning	6 hours
Midterm Exam	1.5 hours
Final Exam	-
	45 hours

Laboratory projects (specify number of weeks on each)

- 4 functional programming labs (2 weeks per lab, each lab contains multiple programming assignments)
- 2 programming projects (individual, 3 weeks each, design and implementation of AI techniques such as alpha/beta pruning, neural networks, etc.)

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	0		Computer Organization and Architecture	0	
Algorithms		2	Concepts of Programming Languages		1
Software Design	0				

Oral and Written Communications - Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

- none

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- History of AI and philosophical/societal issues, 1 hour, possible test question

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- Intelligent Agents, 1 hours
- Problem Solving, 6 hours
- Knowledge and Reasoning, 2 hours
- Uncertain Knowledge and Reasoning, 5 hours
- Learning, 3 hours

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Basically, the entire course is concerned with problem analysis and solution approaches.

Solution Design - Please describe the design experiences common to all course sections.

- 2 programming projects (individual, 3 weeks each, design and implementation of AI techniques such as alpha/beta pruning, neural networks, etc.)

Other Course Information

- Additional Suggested Course Assignments
 - 4 written homework assignments (problems from book on applications and analysis of AI techniques)
 - 1 midterm exam (75 minutes)
 - 1 final exam (120 minutes)
- Planned Course Enhancements
 - Update catalog description to be more standard AI course (Fall 2002) – Introduces students to the fundamental concepts and techniques of artificial Intelligence (AI). These include techniques of problem solving by search, knowledge and reasoning, uncertain knowledge and reasoning, and learning. Also including functional programming and AI applications. Prerequisite: CS 331. (3-0-3)