

CSE282/382: Introduction to Artificial Intelligence
Spring 2008 MW 2-3:15 ITE 125

Professor: Robert McCartney
office: ITE 255, x5232
robert@enr.uconn.edu
office hours: By appointment

Assistant: Greg Johnson
office: ITE 214, x0005
gjohnson@enr.uconn.edu
Office hours: To be determined

Purpose of course: ○ Survey the basic research areas in AI
 ○ Write small versions of classic AI programs
 ○ Apply AI techniques to a substantial project

the text The required text for this course is *Artificial Intelligence: A Modern Approach (2nd Ed.)* by Stuart Russell and Peter Norvig. It is the currently the “standard” AI textbook. It provides a broad introduction to AI, using the notion of agents and their capabilities as a unifying theme. It is also a very large book; we will only cover a subset of the material, but the book can provide you with a reference on the areas we do not cover.

programming In the first part of the course we will be programming in LISP (specifically the CommonLisp dialect). There are various free versions available. Later in the course you may want to program in another language, which will be allowed for the project.

the project The final project, which is substantial, will be a program that learns and plays Checkers. The background for this will be some classic work by Arthur Samuel in the 1950’s. Projects will be done in teams, and we will provide a game server that will allow your program to compete against others. There will be a competition between the programs developed in the class at the end of the semester.

projects and exams The course will include 3-4 programming assignments, the final project, a midterm examination, and a final examination.

grading policy Numerical scores will be given on all assignments. These scores will be used to compute a final grade for the course. Roughly a third of the grade will be based on the programming assignments, a third on the exams, and a third on the project.

expectations Only a few:

1. Honesty is expected; no copying of assignments or code will be tolerated. Specifically, any code that you use that you obtain from any source (written or electronic) should be so identified, and you should clear it with the instructor first.
2. Attendance at lecture is required; much of the lecture material will not be from the book *per se*. If you feel that coming to all of the lectures will be too difficult, drop the course.
3. Any material from lecture, readings, and programs is fair game for exams. If you do not understand any of the undiscussed material, it is up to you to ask questions. Much of the reading will *not* be covered in lecture.

4. The programming required in this course may be fairly difficult at times, and the amount of programming experience varies greatly across the people in the class. If you find yourself having problems, let us know ASAP, not when program or project is due.

Tentative schedule:

Week of	Lecture	Reading
January 21	Course Information & Intro	Chapter 1
January 28	Introduction, Lisp	Chapters 1,2
February 4	Lisp Programming	
February 11	Lisp Programming, Search	Chapter 3
February 18	Search	Chapter 4
February 25	Constraint Satisfaction	Chapter 5
March 3	Adversarial Search	Chapter 6
March 10	No Classes – Spring break	
March 17	Midterm Exam	Review
March 24	Logic and Knowledge Rep (plus Checkers project details)	Chapter 7 Samuel paper
March 31	First-order logic	Chapters 8, part of 9
April 7	Learning	Chapter 18
April 14	Learning	Chapter 19
April 21	“Culture” lectures	
April 28	Project presentations	
May 5	Final Examination/tournament (May 9)	