CS 300 – Artificial Intelligence
Syllabus – Fall 2009

Instructor: David Furcy

Office: Halsey 220

Email: furcyd@uwosh.edu

Phone: 424-1182

Office Hours: MWF 11:00-12:30, Th 8:30-10:00

Feel free to drop in at any other time when my door is open. Alternatively, to ensure that I have enough time to answer your questions, I strongly encourage you to make an appointment.

Class Meetings: 9:10-10:10 on MWF in Halsey 208

Prerequisites: CS 262 with a grade of C or better.

Required Textbook: None

References:

- Lecture slides: Organize them, take notes on and about them. Handouts that are not liberally saturated with your own explanatory notes will likely prove useless when you need them most.
- Materials, resources, and other information posted on the course web site at: http://www.uwosh.edu/faculty_staff/furcyd/cs300
- Not required, but a good encyclopedic reference if you do more with AI after this course: *Artificial Intelligence: A Modern Approach*, Second Edition, by Stuart Russell, Peter Norvig

Tests:

- Exam #1: Thursday, October 15th, 5:00-7:00PM
- Exam #2: Thursday, November 12th, 5:00-7:00PM
- Exam #3: Thursday, December 17th, 5:00-7:00PM

Exams are scheduled at night to circumvent the one-hour time limit imposed by in-class exams. Please block these time slots at the beginning of the semester. The procedure and criteria of eligibility for making up a missed exam are described below.

If you have special needs (or ABSOLUTELY cannot make it to one of the above night exams), please come and talk to me as soon as possible so I can accommodate your needs.

Topic Areas and Corresponding Learning Outcomes:

1. **What is artificial intelligence?** You will be expected to ...
   (a) describe the different characterizations of AI
   (b) identify how those characterizations apply to various stages in the historic development of AI as a discipline
2. **Stimulus-response agents.** You will be expected to ...
   (a) distinguish how the PAGE model (Percepts/Actions/Goals/Environments) is used to characterize different types of stimulus-response agents
   (b) identify what each component of the PAGE model would be in specific examples of AI agents
   (c) specify goals for an SR agent as a quantifiable measure in terms of the agent's environment
   (d) determine rational actions for an SR agent based on maximizing goals given a set of percepts
   (e) evaluate different sets of condition-action rules in terms of their ability to reach an agent's goals
   (f) design an actual software agent to act rationally in the context of an object-oriented framework for solving a simple AI problem
3. **Planning agents and state spaces.** You will be expected to ...
   (a) formulate certain AI problems as state-space search
4. **Uninformed search.** You will be expected to ...
   (a) identify various uninformed search algorithms such as depth-first, breadth-first, least-cost, and iterative deepening
   (b) carry out in manual fashion these algorithms on small search spaces
   (c) compare and contrast the strengths of each of the algorithms vis-a-vis the others
   (d) given a problem, to select the search algorithm that has the best chance of succeeding
   (e) explain why that algorithm has the best chance of succeeding

5. **Informed searches.** You will be expected to ...
   (a) identify various informed search algorithms such as A-star, beam search, gradient descent, simulated annealing
   (b) carry out in manual fashion these algorithms on small search spaces
   (c) design new heuristic functions appropriate for a particular search space
   (d) compare the strengths and weaknesses of different solutions, both in terms of different algorithms and different heuristics
6. **Constraint satisfaction problems.** You will be expected to ...
   (a) formulate a constraint satisfaction problem as a particular type of state-space search
   (b) apply consistency algorithms such as AC-3 in solving constraint satisfaction problems
7. **Adversarial search.** You will be expected to ...
   (a) carry out in manual fashion the minimax algorithm for small adversarial search spaces
   (b) discover how the adversarial search space can be reduced using techniques like alpha-beta pruning
   (c) design heuristic evaluators for a given adversarial search space
   (d) test those heuristic evaluators in a competitive setting
   (e) generalize the basic minimax algorithm to games with more than two players
   (f) generalize the basic minimax algorithm to games where chance is involved in addition to pure strategy
8. **Agents that reason logically.**
   (a) **Introduction to logic and propositional calculus** You will be expected to ...
      i. formulate an agent's knowledge base as a set of sentences in a formal language
      ii. interpret logical entailment in the context of models
      iii. apply traditional propositional calculus to make inferences from a knowledge base
      iv. appraise the weaknesses of propositional calculus in expressing a rich variety of knowledge
   (b) **Predicate calculus.** You will be expected to ...
      i. explore how the predicate calculus allows a much richer degree of knowledge expressibility than does the propositional calculus
      ii. apply forward chaining in building a knowledge base from a set of original premises
      iii. apply backward chaining in answering queries from a logic knowledge base
      iv. assess the inefficiencies of traditional predicate calculus proof methods when used in the context of state-space search
   (c) **Proofs by resolution refutation.** You will be expected to ...
      i. discover the relationship between the methods of proof in traditional propositional/predicate calculus and those of resolution refutation
      ii. apply resolution refutation in solving reasonable problems
      iii. distinguish various heuristics used in efficiently guiding a resolution proof
   (d) **Logic Programming.** You will be expected to ...
      i. transform problems into the knowledge base syntax of the Prolog language
      ii. analyze the search strategy used by Prolog in searching its knowledge base
      iii. plan the structure of a knowledge base to optimize the search carried out by Prolog
      iv. apply Prolog in designing a small expert system
9. **S-R agents that learn – feed-forward neural nets.** S-R agents that learn – feed-forward neural nets
   (a) extrapolate from general brain architecture to connectionist models of AI
   (b) recognize the distinguishing characteristics of a perceptron
   (c) carry out in manual fashion the perceptron learning algorithm for perceptrons with a limited number of inputs and outputs
   (d) explain the relationship between training a perceptron and fitting a line or surface to data points in two- and three-dimensional space
   (e) identify pattern recognition problems that are linearly inseparable and hence not solvable by a perceptron
   (f) solve the linear inseparability problem by introducing a hidden layer of neurons
   (g) explain the role of an activation function in hidden layer networks
   (h) explain how the back propagation training algorithm works in hidden layer networks
   (i) distinguish the back propagation algorithm for training neural nets from the meta-algorithm used by AI practitioners to run neural nets through their actual training
   (j) apply the meta-algorithm for training a neural net in the context of real AI problems
   (k) apply heuristics in determining the appropriate number of hidden nodes for a neural network
   (l) measure how effectively a neural net has been trained
   (m) judge whether or not a neural net overfits its training data

10. **State machines and recurrent neural nets.** You will be expected to ...
    (a) recognize limitations of feed forward neural nets
    (b) discover how those limitations are overcome by recurrent neural nets such as a Hopfield net
    (c) manually carry out the parallel relaxation algorithm for a small Hopfield net
11. **Machine evolution and genetic programming.** You will be expected to ...
    (a) extrapolate from the vernacular of biological evolution to that of genetic programming
    (b) formulate certain AI problems as solution-space search that is appropriate for genetic programming
    (c) carry out in manual fashion the basic genetic algorithm for simple examples
    (d) integrate the various key factors in designing a GA solution to a problem
12. **Semantic networks.** You will be expected to ...
    (a) design a semantic network representation of everyday knowledge
    (b) apply the semantic network representation of that knowledge in answering queries
    (c) appraise the strengths and weaknesses of semantic networks in representing knowledge
    (d) abstract from semantic networks to their application in the emerging "Semantic Web"
13. **Decision trees.** You will be expected to ...
    (a) represent inductive concept learning via decision trees
    (b) carry out in manual fashion the construction of a decision tree via the ID-3 algorithm
    (c) discern between the alternative ways that the ID-3 algorithm allows for choosing the "best" attribute of a node
    (d) apply the information gain method in choosing the "best" attribute of a node
14. **Reasoning with uncertain information.** You will be expected to ...
    (a) construct a belief network (Bayesian network) to model a problem with probabilistic information
    (b) represent queries in uncertain circumstances in terms of their conditional probabilities
    (c) apply probabilistic reasoning to a belief network to answer queries formulated in terms of conditional probabilities
15. **Seeing the "big picture" of AI.** You will be expected to ...
    (a) describe the broad variety of problem-solving techniques that have come to be recognized as "AI"
    (b) discern which of those techniques is the appropriate one to apply in solving a particular problem
    (c) in situations where more than one technique could be applied, select the best technique
    (d) in situations where more than one technique could be applied, test a variety of techniques and be able to assess their success (or lack thereof) in solving the problem
    (e) design and implement actual AI software agents in the context of larger object-oriented frameworks

Course Grading Policy:

Your final grade for this course will depend on programming assignments, 3 exams, and review problems. Each assignment and exam will be graded on a scale from 0 to 100. All assignments will carry the same weight when computing your overall assignment grade. At the end of most classes, I will distribute one or more review problems to be completed and submitted before the next class. Each review problem will be graded on a two-point scale, based on your written answer and oral discussion of it in the following class. These points will be added up at the end of the semester to yield an overall grade for this component of the course. Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	**Weight**
Assignments	45%
Exam #1	15%
Exam #2	15%
Exam #3	15%
Review Problems	10%

Finally, your letter grade for the course will be computed using the following mapping:

Numerical Score	Course Grade	Numerical Score	Course Grade
≥ 92	A	≥ 72	C
≥ 90	A-	≥ 70	C-
≥ 88	B+	≥ 68	D+
≥ 82	B	≥ 62	D
≥ 80	B-	≥ 60	D-
≥ 78	C+	< 60	F

I will be glad to discuss any questions you may have about grades. However, make sure to bring them up **right away**, upon return of each graded submission. Last minute requests, especially after the final exam, will not be entertained.

Attendance and Participation:

You are expected to not only attend every class meeting but also to come prepared for and participate actively in it. Necessary preparation requires you to have studied and assimilated the material covered in previous sessions, to have met with me outside of class to discuss any questions you may have, to have done the assigned reading (when applicable), and to have completed the programming assignments on time.

It is hard to imagine how a student could do well in this course while missing classes, attending them unprepared, or not participating.

On the positive side, I have high expectations for my students and will always support and encourage you. I strongly encourage you to ask any question or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment. Send me email or give me a call to make an appointment. While I will meet with you as soon as my schedule permits, do not expect me to be widely available before an assignment is due.

Late Submissions:

I will describe the submission procedure for your assignments when the time comes. However, let me point out right away that each assignment will come with a due date and a cutoff time after which any submission is considered late, with no exception. The penalty for late submissions is computed as follows:

Turned in	**Penalty**
On due date, but after cutoff time	-10%
The day after the due date	-20%
Two days after the due date	-40%
Three days after the due date	-60%
Four days after the due date	-80%
Five or more days after the due date	-100%

Note that submissions that are more than 4 days late will receive zero points. Late submissions can easily be avoided by starting to work on the assignment right away and asking me questions early if you get stuck.

The penalty for late submissions can be waived in only one scenario, namely if you give me a signed note from the attending physician or a written justification for the extension from the Dean of Students Office. If you miss a scheduled exam, you may be able to take a make-up exam provided you give me a valid written justification (see above) ahead of time if possible. **Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester.**

Pair Programming:

Programming assignments may be done individually or in teams of two. In contrast, writing assignments will typically have to be completed individually. **Caution:** If you are considering pair programming, be sure that you are doing it to maximize what you learn, understand, and achieve in the course, not to reduce your work load. To get some idea of whether you might work effectively as part of a "programming pair," check out the article "All I Really Need to Know About Pair Programming I Learned in Kindergarten" at `http://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF`. Much more information can be found on the "Pair Programming" home page at `http://c2.com/cgi/wiki?PairProgramming`. If you work on a team, only one electronic copy and one hard copy should be submitted. The name of both you and your partner should appear on all submitted materials.

Collaboration versus Cheating:

All submissions must be the work of either one or two students (when allowed), namely the one(s) whose name appears on the submission. You must submit your own work. You may not look at or borrow any piece of code of any length from anyone else, unless you can live with a zero and the other potential academic sanctions of cheating. Check out the UWO Student Discipline Code (UWS 14) at http://www.uwosh.edu/dean/conduct.htm for details.

Review Problems:

It is well known that, as time goes by, students remember less and less about what they heard about in lectures, unless the material is actively processed (not just heard) in order to build strong brain connections between the new material and existing knowledge. To foster deep learning, I require students to study the material right after each lecture and to submit the solution to relevant problems by the beginning of the next class. For this strategy to work, students must work on these problems RIGHT AWAY. I will be more than happy to help you with these problems if you come to my office hours within three days of the corresponding lecture. After those three days (not counting weekends), *because you have made the choice to not learn effectively*, you are on your own with these review problems.

Final Note:

I expect every committed and hardworking student to do well in this course. I am looking forward to a fun and rewarding semester together. Given my high standards, I could not be more satisfied than if everyone earned an A in this course.