Syllabus  Artificial Intelligence  
CS 465  Spring 2009  
CRN: 29206  
Time  Monday, Wednesday, Friday 11AM-11:50AM  
Location  BARRET 035 (computer lab in basement of library)  
Instructor  Eric Breck (http://faculty.rhodes.edu/brecke)  
Office  419 Ohlendorf (901-843-3725)  
Office Hours  Office Hours will be posted on my website; please send e-mail to make an appointment if you cannot make office hours.  
Email  brecke@rhodes.edu  
To ensure a quick response, the subject line of your emails should read cs465: [subject of question]  
Do not assume I will read and reply to your messages instantaneously.  

Moodle  
All assignment submission will be via moodle. Do not e-mail me assignments. Course announcements and any updates to this syllabus will be posted via moodle.  

Book  
The course textbook is Artificial Intelligence: A Modern Approach, Second Edition, by Russell & Norvig. Supplemental material may be distributed in class  

Prerequisites  
The course assumes successful completion of CS 141, CS 142, and CS241 (a grade of C- or better).  

Course Description  
This course provides an introduction to the major subareas and current research directions in artificial intelligence. The course will cover the following topics (not necessarily in this order):  

- Search  
- Game-playing  
- Logical reasoning  
- Probabilistic reasoning  
- Machine learning  
- Natural language processing  

Depending on time and/or interest, we may also discuss additional topics such as robotics, vision, or planning.  

Programming Assignments  
Programming assignments will generally be distributed on moodle immediately following the due date of the prior assignment.  

In general you are permitted to use any major programming language you choose for assignments in this course. If you use anything other than the versions of Python or C++ installed on the lab machines, you must (a) get approval from me before beginning to code, and (b) give me detailed instructions on how to compile/execute your code. I will generally approve any language, I just don’t want you to use a language that makes a particular assignment utterly trivial. Specific assignments may require use of a particular language.
The first line of each program source code file submitted for credit must be a comment that states your name and the assignment number. Unless otherwise stated in the assignment, submit only source files (e.g. .cpp, .h, .py) not other files generated by Visual Studio or the IDE of your choice (e.g. .exe, .obj, .pdb). Assignments should be submitted via Moodle (before 11:55 pm on the date due). Projects received after 11:55 pm on the date due are considered late. You are allowed to use the course textbook and the course notes for these programs. The use of any other material is forbidden.

**Final project**

You will complete a final course project (in groups of 2-3). Your project can be related in any way to AI - you might write a program that plays checkers, build a robot, or create a question answering system.

**Timeline**

- Project pre-proposal due Jan 23. Your initial proposal should list the group members, describe the problem, issue, or task that you are interested in studying, clearly describe what the final system will actually do in terms of input/output behavior, and provide a rough work schedule. This pre-proposal should be no longer than two pages in length (one page is fine too).
- Finalized project proposal due Jan 30.
- Status report one due February 27
- Status report two due March 30
- Project demos will occur in late April

**Exams**

There will be two midterms and one final exam.

- **Midterm 1**: Monday, 16 February, in class
- **Midterm 2**: Friday, 27 March, in class
- **Final Exam**: Monday, May 4, 1PM
- Make-up exams will only be given in extreme circumstances.

**Grade Breakdown**

- 40% Programming Assignments (there will be 6-8 assignments).
- 20% Final project
- 16% Midterms
- 24% Final
Course policies

The instructor reserves the right to alter this syllabus as necessary.

Attendance

Attendance is expected at each class lecture period. You are responsible for learning any material presented during a class that you miss. Lecture notes and textbook chapter references may be offered, but are not intended as a substitute for attending lecture – additional material may be presented beyond what is covered in these references.

Backups

Each student is responsible for keeping a back-up copy on disk of all files turned in for an assignment. Failure to do so could result in loss of credit for an assignment.

Classroom behavior

Please respect me and your classmates and do not browse the web, check e-mail, send or read text messages, or have private conversations during class.

Collaboration

You are expected to work individually on assignments. However, you are allowed and encouraged to discuss high-level details of the assignments. If group work is allowed, it will be mentioned explicitly in the assignment. Do not share your own code or copy others’ code. Tools such as MOSS make it easy for me to check if you have copied code and just changed the whitespace and variable names; don’t do it.

Communication with me

Like any human, my memory is imperfect. Therefore, if you make a verbal, in-person request or suggestion during class or elsewhere, you should follow up with an e-mail containing the same information. This includes but is not limited to: scheduling meetings, asking for regrades or other grade changes, indicating errors on the moodle site or in assignments, and anything else that requires me to remember to do something later.

Grading programs

Programs will generally be graded using the following guidelines.

- A (100 pts): The program is carefully designed, efficiently implemented, well documented, and produces clearly formatted, correct output.
- A- (94 pts): This is an ‘A’ program with one or two of the minor problems described for grade ‘B’.
- B (88 pts): A ‘B’ program could easily have been an ‘A’ program, but it may have minor/careless problems such as poor, inadequate, or incomplete documentation; several literal values where symbolic constants would have been appropriate; wrong file names (these will be specified per program assignment); sloppy source code format; minor efficiency problems; etc. (This is not an exhaustive list.) You would be wise to consider ‘B’ the default grade for a working program — this might encourage you to review and polish your first working draft of an assignment to produce a more professional quality final version of your program.
- C (75 pts): A ‘C’ program has more serious problems: incorrect output for important special cases (the "empty" case, the "maxed-out" case, etc.), failure to carefully follow design and implementation requirements spelled out in the assignment description, very poor or inefficient design or implementation, near complete absence of documentation, etc.
• D (60 pts): A ‘D’ program compiles, links, and runs, but it produces clearly incorrect output for typical cases. Or, it may deviate greatly from the design or implementation requirements stated in the assignment description.

• F (35 pts): Typically, an ‘F’ program produces no correct output, or it may not even compile. It may “look like a program” when printed as a hard copy, but there remains much work to be done for it to be a correct, working program.

Late assignments
Late assignments will be accepted, with a penalty of one letter grade per day. If a genuine emergency situation arises, please see me and we will work something out.

Numeric to letter grade conversion
Grades during the course will generally be given as numbers. The final course letter grade will determined from the aggregate numerical grade using the following scale:

- A : [93%, 100%]
- A- : [90%, 93%)
- B+ : [87%, 90%)
- B : [83%, 87%)
- B- : [80%, 83%)
- C+ : [77%, 80%)
- C : [73%, 77%)
- C - : [70%, 73%)
- D : [65%, 70%)
- D- : [60%, 65%)
- F : [ 0%, 60%)

For borderline cases, I may take into account participation, and/or attendance, and improvement during the semester.

Scholastic Behavior and Academic Integrity
Plagiarism, cheating, and similar anti-intellectual behavior are serious violations of academic ethics and will be correspondingly penalized. If you are concerned about a possible violation of this kind, please talk with me. I understand the pressure that students may experience while at Rhodes, and I will try to help as best as I can.

All projects and tests must be the student’s own work, unless otherwise instructed by your instructor. Copying all or part of a project assignment, or downloading code or text from the Internet and submitting it as your own, or having someone else write code or text for your assignment, or having someone else debug your assignment, or allowing someone else to copy from you, or writing or coding or debugging someone else’s assignment — these are all included in the definition of reportable Honor Code violations for this course. If you have any doubts about whether or not a work practice on a project assignment is acceptable, please clear it with the instructor before proceeding.

Special Accommodation
If you are in need of special accommodations, please register with the Office of Student Disability Services (http://www.rhodes.edu/disability) as soon as possible so that all necessary arrangements can be made. Accommodations cannot be made unless you register; they also cannot be made retroactively.