CS 730/730W/830 Introduction to Artificial Intelligence
Spring, 2008
The course website is `http://www.cs.unh.edu/~ruml/cs730/`.

Overview

The goal of this class is to help you learn **how to build intelligent software**. We'll cover concepts and algorithms in areas such as agent architecture, combinatorial search and decision making, knowledge representation and reasoning, planning, reasoning under uncertainty, and learning. You should already be a fluent programmer, understand common data structures, and be familiar with basic complexity analysis and big-O notation.

Contact Info

Prof. Wheeler Ruml
`ruml` at `cs.unh.edu`
Kingsbury W233, 2-2683
Office hours: Tuesdays 3:30–4:30pm or by appointment

Jordan Thayer
`jtd7` at unh.edu
Kingsbury N216

Please come to my office hours! If you'd like to talk but can't attend office hours, we can set up an appointment — just call, email, or catch me after class. If you choose to send me email, please include `CS 730` in the title and be aware that, due to the volume of email I receive, I cannot promise a prompt reply.

Required Text

Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, second edition, Prentice Hall, 2003. Available at the UNH Bookstore as well as on-line. If you buy a used copy (`used.addall.com` is a good site to know about), please be sure to select the **second** edition (green cover), not the first (maroon cover).

Evaluation

These breakdowns are tentative. There are currently no plans to hold a final exam.
Non-writing intensive:

- 50% five programming assignments.
- 30% final project, including a proposal, oral presentation, and final write-up.
- 20% two written exams

Writing intensive:

- 50% five programming assignments. Special attention will be given to the clarity and beauty of your code (style, organization).
- 30% final project, including a proposal, oral presentation, and final write-up. Special attention will be given to your write-up, which you may submit multiple times before the deadline for feedback.
- 10% journal, submitted weekly, of your thoughts on the course content. This can include, for example, outlines of potential class projects, ideas for commercial applications, reactions to our formalizations of concepts, and discussions of algorithms.
- 10% two written exams

Please note that assignment deadlines are not flexible and there will be no credit for late work. This allows us to discuss solutions promptly after the deadline. The schedule indicates when assignments are due so that you can plan your work in advance.

Except for the final project, which may be done in a team, all your work in this class should be your own. Any collaboration must be cleared with me in advance and you must cite all sources you use in preparing your work, other than the textbook, lectures, and recitations. As a scientist, I take a dim view of academic

dishonesty and UNH policy provides for actions starting with failing the course and including dismissal from the University.

If you are registered with the student disability office, please let me know right away so that I have time to provide proper accommodation.

Mechanics

Programming assignments:

You may use any programming language you wish in this course. We will give you example inputs for your programs that indicate the sort of problems your solutions will be tested on. For each assignment, 10% of the grade will be calculated from a 'milestone' solution, due about a week before the full solution, that will exhibit only partial functionality. The best performing final solution submitted by an undergraduate and by a graduate student will receive extra credit and recognition in class.

While students in the non-writing intensive version will not be explicitly graded on code beauty, please note that it is in your best interest that we be able to quickly understand your code so that we can try to give you partial credit if your program doesn't behave as nicely as our solutions do.

You must hand in source code and a 'make script' that will generate a working executable from the source. Assignments must compile and run on any reasonable Linux set-up, including the CIS Unix machines. If your code has dependencies beyond very standard stuff like gcc or python, please include copies of those things in your code base and configure your make process to use them.

Source code should be submitted via CISunix scripts (detailed in each assignment) and a brief hardcopy write-up should be handed in at class, containing: any comments you have about the assignment, things we should know before evaluating your solution, suggestions for improving the assignment itself, a transcript of your code solving the example cases, and a listing of your source code (2 pages per page, as with `a2ps -2`).

Written exams:

Held during common exam time, current scheduled for March 13 and May 1. No notes or books allowed.

Final project:

You may work in teams of up to three people on the final project. The project is intended to be an implementation and evaluation of one or more algorithms, but feel free to consult me if you would like to undertake a different kind of project. The topic must relate to the course (for example, it should appear in the textbook somewhere) but otherwise you may propose any topic you wish. Feel free to submit your proposal well before the deadline, but wait until it has been approved before starting serious implementation work.

The project proposal must specify what you intend to do, why it is interesting, who is responsible for each portion of the work, and what references and other sources you will rely on.

The project write-up contributes the vast majority of the project grade. You will need to submit any source code you write, but the write-up is what will be graded. It should clearly state the problem that the project addresses, discuss the methods employed in solving the problem, evaluation their performance and adequacy, and relate them to other existing possibilities. You should also mention possible extension of your project and things you would have done if you had more time. If you are in the writing-intensive version, you may submit your write-up multiple times before the due date for feedback.

Just before exam period, each project team will give a 15 minute oral presentation of their project and the results obtained so far.

Journal (writing intensive only):

Please email me and Jordan an entry of at least 300 words each week before 2:10pm Tuesday. (Alternatively, you may email a pointer to a blog containing your entry.) Journal entries need not make a formal academic argument, but should have correct spelling and grammar, be coherent, and relate to specific topics discussed in the course. I suggest you use your entries to think about potential project ideas. Other possibilities include critiquing the algorithms presented in class, proposing new algorithms, or outlining possible applications of AI. If you are unsure if a particular topic is appropriate for a journal entry, feel free to ask.