COURSE 4200/ COMP 5430: ARTIFICIAL INTELLIGENCE

Instructor	Dr. Jonathan Mwaura
Office	Dandeneau Hall 314
Email	jonathan_mwaura@uml.edu
Class Time	Tue/Thu: 1700: 1815
Class Location	Southwick Hall 407
Teaching Assistant	TBD

Catalog description

Topics include: search techniques and their properties, including A*; game-playing, including adversarial and stochastic search; probabilistic reasoning, including Markov Decision Processes and Hidden Markov Models; and reinforcement learning, including value iteration and q-learning.

Topics are developed theoretically and with programming assignments. The course includes a student-directed final project and paper.

Co-Requisites: COMP 3010 Organization of Programming Languages and MATH 3860 Probability and Statistics.

What is AI?

There are few areas in computer science as broad as artificial intelligence. AI encompasses algorithm design, agents and robots, natural language understanding, expert systems, music and sound, as well as philosophical questions about the nature of consciousness!

One organizing principle across the field is the formalization of real-world problems based on utility functions, and then the use of various techniques to maximize expected utility. Based on this unifying theme, we will study:

- Problem solving with Search (including uninformed, informed techniques)
- Adversarial and stochastic (probabilistic) search (as used in multi-player games)
- Probabilistic inference, including Markov Decision Processes (MDPs) and Hidden Markov Models (HMMs)
- Learning: Supervised, Unsupervised, Reinforcement learning, including value iteration and q-learning.

Ideas will be developed theoretically and with practical programming challenges using the U.C. Berkeley "Pac-Man projects," in which you program a progressively complex series of challenges inspired by the original Pac-Man arcade game.

These problem sets "teach foundational AI concepts, such as informed state-space search, probabilistic inference, and reinforcement learning. These concepts underlie real-world application areas such as natural language processing, computer vision, and robotics" (see http://ai.berkeley.edu/project_overview.html).

Coding will be done in Python, which you must teach yourself during the class if you don't know it yet.

A big part of AI is not just knowing a collection of approaches, but understanding which approach is right for which problem. To encourage the development of these skills, there will be a significant semester project where you will apply ideas from the course to a problem of your own choosing.

TEXT BOOKS/ Reference Materials

(1) **Artificial Intelligence: A MODERN APPROACH (AIMA3), by Russell and Norvig (3rd Edition)**

The book is comprehensive, is well-written in the most exemplary manner, and has detailed bibliographies for each chapter. It is by far the most popular book used to teach AI, and it is also the best.

Other recommended texts

(2) **Artificial Intelligence: Foundations of Computational Agents by David K. Poole and Alan K. Mackworth (2nd Edition)**

You can buy a copy off Amazon or download freely onhttp://artint.info/2e/html/ArtInt2e.html

(3) **Computational Intelligence: An Introduction by Andries P. Engelbrecht (2nd Edition)**

You can buy a copy off Amazon. The book is well detailed and has a focus on the field of Computational intelligence rather than general artificial intelligence techniques.

Tentative Course schedule

This week by week schedule is provisional in nature. It will be adjusted based on the pace of the course

Date	**Topic**	**Readings**
Jan 22nd	AI: Introduction	AIMA3-chap 1
Jan 24th	Intelligent Agents	AIMA3-Chap 2

Problem Solving with Search		
Jan 29th	Search 1: Uninformed Search	AIMA3-Chap 3.1-3.6
Feb 01st	Search 2: Informed Search	
Feb 05th	Search 3: Local Search & Optimization	AIMA 3 – Chap 4
Feb 7th	Adversarial Search: Game playing	AIMA 3 – Chap 5
Feb 14th	**(In-Class Quiz with TA)**	
Feb 19th	**No class: Monday Schedule***	
Feb 21st	Constraint Satisfaction Problems	AIMA3 – Chap 6
Knowledge, Reasoning and Planning		
Feb 26th	Propositional Logic: Satisfiability & Planning	AIMA3 – Chaps 7/10
Feb 28th	Mid-Term #1 (Search, Adversarial Search, CSP, Logic)	
Mar 05th	Mid Term Review	
Reasoning and Acting under Uncertainty		
March 7th	Probability	AIMA3 – Chap 13
March 19th	Markov Decision Processes 1	AIMA 3- Chap 17.1 -17.4
March 21st	Markov Decision Processes 2	
March 26th	Bayesian Networks: Representation	AIMA3 – Chap 14
March 28th	Bayesian Networks: Independence	
April 2nd	Bayesian Networks: Inference	

April 4th	Markov Models	AIMA3 Chap 15.1 -15.3
April 9th	Hidden Markov Models	
April 11th	Mid-Term #2 (Naves Bayes, MDPS, HMMs, Bayes Net)	
April 16th	Mid-Term # Review	
Learning		
April 18th	Learning 1: Reinforcement Learning	AIMA Chap 21.1 -21.3
April 23rd	Learning 1: Supervised learning	AIMA3 Chap 18.1 – 18.2
April 25th	Learning 2: Artificial Neural Networks	AIMA Chap 18.7
Course conclusion		
April 30th	Final Project Presentations #1	
May 2nd	Final Project Presentations #2	
May_	Final Exam / Final Project paper Due.	

Grading

There are four categories of work that will be assessed. These are:

- **Programming assignments 25%.** The assignments are the primary way for developing an understanding of how to put AI theory into practice. Assignments are due at 11:30pm the day that they are due. Assignments will lose 20% of their value per day that they are late. One minute late counts as one day late.
- **Take Home Quizzes --- 10 points**
- **Two mid-semester exams 10% each x 2 = 20%.** Exams will be held in class. Make-ups are allowed only in the case of documented emergencies.
- **Final exam 20%.** Date shall be announced later.
- **Term (team) project 25%.** A significant part of the class will be an independent implementation project, which you will specify and carry out, primarily over the last month

of the semester. We'll start conceptual work on the project earlier than that. I will expect the project to represent a significant work effort.

You will apply the ideas developed in the class in an original software implementation. You may thus connect the ideas of the class with your own interests—music, robotics, art, databases, the web, networking, gaming, etc. The learning goal of the project is to have you find some real-world relevance of the ideas in the class.

This is required to be a team project. Supports will be put in place to make sure that all members of each team are fairly graded on their respective contributions. Teams must consist of two or three persons.

Undergraduate Project Sequence

For undergraduates, COMP.4200 Artificial Intelligence can be grouped with COMP.4210 Machine Learning *or* COMP.4500 Robotics I to form a course pair.

Graduate Group

For graduate students, COMP.5430 is in Group III, “b Human-Computer Interaction, Visualization, Robotics and AI.”

Collaboration and Academic Integrity Policy

You are welcome to discuss ideas in the class with your peers. However, pair programming or other side-by-side work that involves sharing of code is not allowed. By turning in an assignment, you attest that *you have written* the new code that it includes. Please be familiar with the university's academic integrity policies: for undergraduates for graduate students

Plagiarism

Copying others' code or writing is not allowed.

Your coding assignments will be verified as being your own work using Stanford's Measure of Software Similarity (MOSS), as well as inspection by the TAs/graders and course professor.

If your code is found to substantively include the work of others, you will receive a grade of zero for that entire assignment. If there is a second offense, you will fail the course.

For the written assignments, you will be given instruction on how to properly cite others' work. Your written assignments will be verified using TurnItIn.

- Citing others' work is a sign of respect for others' contributions, and diligence and confidence on your part.
- Copying others' words into your own writing—even including rephrasing others' work without citation—is considered plagiarism.

Plagiarism is a significant academic offense. Plagiarism on the final paper will result in receiving **a grade of F for the course.**

No Posting of Solution Code Policy

UMass Lowell's policy on academic integrity states that *assisting students in their own acts of academic dishonesty is itself an a violation of academic integrity*. See Academic Misconduct Subject to Disciplinary Action, 1(f).

Consistent with this, you are not allowed to post solution code to problem sets assigned in this class in public places (e.g. Github). This includes your own solutions as well as solutions that may be provided by the instructors.

Please note that this is typical policy at premier computer science departments. E.g.:

- Princeton COS 126. *"Your work must never be shown or communicated to anyone who is taking COS 126 now or who might take COS 126 in the future. ... You must never place your work in any public location (including websites, leaving printouts in a classroom, etc.). ... The rules ... continue to apply even after this semester is over."*
- Harvard CS50. *"Not reasonable: Providing or making available solutions to problem sets to individuals who might take this course in the future."*
- MIT 6.01. *"**Students should never share their solutions (or staff solutions)** with other students, including through public code repositories such as Github."* (emphasis in the original)

Thus: **Do not publish your solutions to problem sets.** Doing so will be considered an act of academic dishonesty and you will receive a grade of F for the course.

You may save your work to private repositories that cannot be retrieved by others. I encourage you to sign up for Github's Student Developer Pack, which allows you to create private repositories (among other benefits). It's free to students.

Attendance Policy

Students are responsible for all material covered in class. Attendance will not be taken.

Exams will be announced at least one week before they are administered. In-person participation of final project presentations is required.

Make-up opportunities will be made ***only*** in the case of emergencies, not scheduled conflicts (e.g., work).