COGS 125 / CSE 175
Introduction to Artificial Intelligence

University of California, Merced

Fall, 2021

Instructor

David C. Noelle, Ph.D.
dnoelle@ucmerced.edu
262A Social Sciences & Management Building
(209) 228-4101
Office Hours: 3:00 – 4:00 P.M. on Tuesdays and by appointment

Teaching Assistants

Liza Oh
loh6@ucmerced.edu
Office Hours: by appointment

Ghazal Zand
gzand@ucmerced.edu
Office Hours: by appointment

Class Meetings

Tuesdays & Thursdays, 12:00 – 1:15 P.M., in 116 Classroom & Office Building

Wednesdays, 9:00 – 11:45 A.M., in 135 Social Sciences & Management Building
Wednesdays, 1:30 – 4:15 P.M., in 135 Social Sciences & Management Building
Fridays, 9:00 – 11:45 A.M., in 135 Social Sciences & Management Building
Fridays, 1:30 – 4:15 P.M., in 135 Social Sciences & Management Building

Overview

Artificial intelligence (AI) is the science and engineering of making intelligent machines. In this age of the computer, artificial intelligence typically involves the fabrication and analysis of intelligent computer programs. It is a field firmly rooted in the discipline of computer science while drawing on insights from philosophy, mathematics, psychology, and neuroscience. As a research endeavor, AI has frequently been mocked for its failure to produce a convincing simulacrum of a human intellect, while many of the successes of AI research efforts have become so well accepted that they are now considered the domain of other areas of computer science.

This course briefly introduces elementary artificial intelligence findings and techniques. While it is impossible to fully cover over 60 years of active research in a single term, the curriculum for this class has been designed to cover a moderately broad range of foundational topics, including heuristic search, logical reasoning, planning, reasoning under uncertainty, and machine learning. The general educational goal of this course is to make students familiar with the major problems and methods addressed within these foundational areas, allowing successful students to recognize system development situations in which these techniques might be fruitfully applied.

Learning Outcomes

Students who successfully complete this course will have acquired a sufficient understanding of the basic concepts and methods of artificial intelligence to make use of some elementary artificial intelligence techniques in the design of computer systems. They will also come to possess insights into how these methods may be used to model the information processing mechanisms of human cognition. This class will also provide students with preparation for advanced training in artificial intelligence, such as that provided by basic graduate level coursework in the field.

These *student learning outcomes* are expected to be valuable to students pursuing a variety of undergraduate degree programs at the University of California, Merced. Specifically, with regard to the *Cognitive Science* B.S. and B.A. degree programs, these *student learning outcomes* contribute to a number of *program learning outcomes*, including increasing student knowledge of landmark findings and theories in cognitive science, developing student appreciation for formal and computational approaches in cognitive science, and improving the ability of students to argue for or against the use of computational models to address various scientific questions. With regard to the *Computer Science and Engineering* B.S. degree program, the *student learning outcomes* outlined in the previous paragraph contribute to a number of the computer science *program learning outcomes*, including practice applying knowledge of computing and related mathematics, practice designing, implementing, and evaluating computer-based systems, exposure to current computing techniques, and exposure to examples of trade-offs in software system design. In summary, students enrolled in a variety of undergraduate education programs should find this course useful for meeting their program's objectives.

Resources

Meetings

Class meetings will take place every Tuesday and Thursday of this semester according to the schedule outlined at the end of this document. Meetings will take place from 12:00 P.M. to 1:15 P.M. in Room 116 for the Classroom & Office Building (COB). Class meetings will consist of lectures reviewing the highlights of course readings, introductions to new material, tests of student understanding, and explorations of related topics. These class meetings will also include opportunities for discussion, and students are encouraged to actively participate in these conversations. Indeed, student performance will be evaluated, in part, based on class participation.

Prior to each class meeting, students will be expected to complete readings specified in the schedule at the end of this document. In addition, students may be expected to view a small set of short online videos which provide overviews of the topics scheduled for that day.

Four class laboratory sessions have been scheduled for this course. Two of these are scheduled on Wednesdays, from 9:00 A.M. to 11:45 A.M., and from 1:30 P.M. to 4:15 P.M. in the afternoon. Two are scheduled for Fridays, also from 9:00 A.M. to 11:45 A.M., and from 1:30 P.M. to 4:15 P.M. in the afternoon. All laboratory sessions will be held in the computer laboratory in Room 135 of the Social Sciences & Management (SSM) Building. Enrolled students are required to attend *one* of these four laboratory sessions each week. Laboratory sessions will involve instruction in computer programming, the supervised hands-on execution of various exercises, and time to discuss and work on the course's five computer programming assignments. Student mastery of some course material will be evaluated based on student performance on computer programming assignments, and these laboratory sessions will provide opportunities for students to receive assistance from the teaching team on these assignments, as well as time to write the required programs.

Web Site

Online materials for this class will be disseminated through the `CatCourses` learning management system (LMS). This system may be accessed through its central portal:

`https://catcourses.ucmerced.edu/`

Enrolled students should be provided with access to a section of `CatCourses` specifically reserved for this class. In particular, a section shared between CSE 175 and COGS 125, called "`F21-COGS 125 01/CSE 175 01`", should be available to all enrolled students. The instructor should be promptly informed if such access is not appropriately granted.

This website will be used to announce updates to the class schedule, as well as to distribute class materials. Various exercises, assignments, and examinations will be administered through this site. Students are required to obtain regular access to this resource, and they are strongly advised to consult it frequently (e.g., at least daily).

Readings

The primary source of expository readings for this course will be:

AIMA : Russell, S. & Norvig, P. (2020) *Artificial Intelligence: A Modern Approach, Fourth Edition.* Pearson Education: Upper Saddle River, NJ.

This book is available in the campus bookstore. Textbooks may be ordered from the campus bookstore at:

`https://www.bkstr.com/ucmercedstore/shop/textbooks-and-course-materials`

Please note that this course will be making use of the *4th Edition* of this textbook, which has been recently published. In addition to providing explanatory text, this tome will also be used as a rough guide for the sequencing of topics to be discussed in this class, and some exercises and examination questions may be drawn from its pages. Students are required to have regular access to this textbook throughout the semester. In addition to being useful for this class, this book is an excellent reference to have on hand. Indeed, it has consistently been among the most cited publications in the entire discipline of computer science!

While this book provides rather comprehensive coverage of the topics relevant to this course, it does not always offer particularly deep treatments of those topics. Thus, the instructor will supply students with a few additional readings during the semester. These supplementary readings will typically be made available online, through the course website. Students are also encouraged to augment their reading with other sources. Recommended readings on specific topics are available, upon request, from the instructor.

Programming Assignments

Reading about and discussing artificial intelligence techniques is rarely sufficient to develop a deep understanding of those methods. In hopes of fostering a deeper understanding, students enrolled in this class will produce five small computer programs making use of the elementary AI methods covered.

While artificial intelligence methods have been implemented in virtually every widely adopted computer language, the traditional language for AI programming is Lisp. Lisp provides many constructs which have proved particularly useful in AI programming, and early Lisp software development tools have inspired many of the modern software development environments which are popular today.

Unfortunately, few undergraduate students receive formal training in Lisp programming. Instead, other computer programming languages, such as Java™ and C++, frequently form the foundation of early computer science training. An increasingly popular programming language, used in AI systems, is Python. Indeed, local introductory computer programming courses have recently been changed to focus on Python rather than the Java™ language. This course assumes that students possess a basic understanding of computer programming and have fundamental skills in writing and testing computer programs. This knowledge will be leveraged during the completion of computer programming assignments for the class. It cannot be expected, at this time, that all students will be familiar specifically with programming in Python, however. Thus, laboratory sessions will provide an intensive introduction to programming in Python, in addition to assistance with AI programming assignments. Each student will be expected either to already be competent in Python or to be able to quickly aquire or refresh the necessary programming knowledge with the assistance of laboratory tutorials. All computer programming assignments in this class must be completed in the Python language.

Students with doubts concerning their programming skills should consult with the instructor as early in the term as possible. The programming assignments used in this class form a central component of the course curriculum, and the successful completion of these assignments will require basic knowledge of Python programming.

Computer Laboratory

Students are welcome to conduct work on programming assignments in any Python development environment on any computer available to them. Alternatively, there are remote computer laboratory facilities which are accessible through a web browser. These laboratory computers are equipped with the PyCharm integrated development environment (IDE), running on the Microsoft's *Windows* operating system. These remote machines may be used by students to work on computer programming assignments. Indeed, if assignment solutions are developed in any other environment, they should be checked in the laboratory environment, as this is the environment in which submitted assignments will be evaluated. Whether obtained through the remote laboratory facility or installed on a personal computer, students are required to have access to an appropriate Python development environment.

The remote laboratory computers are managed by OIT, and information about this resource may be found at:

`https://remotelab.ucmerced.edu`

There are a variety of Python tools installed on these computers, including the PyCharm IDE. A free, open source version of PyCharm, called the "Community" version, is available for *Windows*, *macOS*, and *Linux* at:

`https://www.jetbrains.com/pycharm/`

Students should feel free to install this environment on their own computers so that they may work on programming assignments in a familiar environment on their own machines.

Expectations & Evaluations

Background

Students attending this course are expected to have some background and interest in computer science, cognitive science, or a related field. An introductory familiarity with basic probability theory and differential calculus is also expected. Perhaps most importantly, students must have a basic competence in generating computer programs, allowing them to quickly learn and use the Python programming language. Our campus' introductory computer programming course sequence, culminating in CSE 21, is a prerequisite for this class, as this sequence provides basic training in computer programming.

Learning Activities

Studying artificial intelligence is a challenging interdisciplinary endeavor requiring familiarity with notions from algorithm analysis, data structures, formal logic, statistics, psychology, and a number of other disciplines. Thus, students may find this class both rewarding and demanding. Mastery of the course materials will require extensive reading, puzzling through unfamiliar concepts, active participation in class discussions, learning new mathematical formalisms, many hours of hands-on experience constructing small intelligent systems, and a willingness to view computation and cognition in new ways.

Specific readings will be specified for each class meeting, and participants will be expected to have studied those readings *prior* to gathering, so as to promote thoughtful questions and knowledgeable discussion. Students will be expected to contribute constructively to discussions, bringing to bear both insights into the material at hand and relevant knowledge acquired in other contexts.

One of the best ways to learn artificial intelligence techniques is to actively use them in the fabrication of working computer programs. Thus, five programming assignments will be issued with the goal of helping students acquire a richer understanding of relevant AI methods. These programming assignments will involve the production of working Python programs, and they will require substantial time and effort to complete. Student solutions to these assignments will be evaluated by the teaching team, and appropriate feedback will be given.

While the fabrication of intelligent programs is an excellent way to explore AI topics, there is insufficient time in a single academic term to allow for the generation of programs covering the full range of AI methods surveyed in this course. Thus, students will necessarily encounter some concepts and techniques only in readings and lectures. Mastery of these topics will be evaluated in two ways. First, a short online quiz will be administered via `CatCourses` on almost every Friday, testing student comprehension of material from required readings, required video presentations, and class meetings. Importantly, these quizzes will consistently be *comprehensive*, with questions potentially addressing any topic covered in the class up to that point. Second, a comprehensive final examination will be administered at the end of the term.

Participating in a Learning Community

This class is intended to house a supportive learning community of people sharing the common goal of enriching our understanding of artificial intelligence. Building and maintaining such a learning community will require contributions from both members of the teaching team and from each student. In an environment of civility and integrity, all participants should feel free to call attention to learning obstacles and suggest approaches to improve learning. Providing mutual support is particularly important during these days of global crisis. All participants will need to try to remain sensitive to the ways in which this unusual time may be introducing difficulties for other members of the class community. The teaching team will be working to adapt to the needs of students, and it is hoped that those needs will be communicated when they are recognized. Students will be expected to be reasonably generous in their responses to difficulties faced by other students or by those offering instruction. A mutually supportive environment will help us achieve our shared goal of learning about this exciting field.

As expressed in its *Principles of Community*, the University of California, Merced, is committed to providing an inclusive educational environment. All participants in this class are expected to exhibit respect and compassion for their fellows. Diversity is expected and welcome in this setting. Students who experience treatment lacking in civility or challenging their dignity in any way — including assaults upon their identity with regard to gender, sexual orientation, race, ethnicity, class, or religion (or the lack thereof) — are encouraged to communicate this to the instructor, with the goal of improving the class climate for all. Alternatively, support may be sought through the Office of Student Affairs:

`https://studentaffairs.ucmerced.edu/students/campus-climate`

All students possessing an appropriate educational background for this class should feel equally welcome to paricipate in all class activities.

The University of California, Merced is committed to creating learning environments that are accessible to all. If you anticipate or experience physical or academic barriers based on a disability, please feel welcome to contact the instructor to privately discuss options. In addition, please contact Student Accessibility Services (SAS) at (209) 228-6996 or "`access@ucmerced.edu`" in order to explore reasonable accommodations. All accommodations must have prior approval from Student Accessibility Services on the basis of appropriate documentation.

If you anticipate or experience educational barriers due to pregnancy, a temporary medical condition, or injury, please feel welcome to contact the instructor in order to discuss options. You are also encouraged to contact the Dean of Students for support and resources at:

`https://studentaffairs.ucmerced.edu/dean-students`

Students are expected to notify the instructor of any absence due to illness or quarantine as early as possible. Students exhibiting COVID-19 symptoms are not permitted to participate in campus activities, and they are expected to stay home.

Course participants are expected to embrace the course material with earnest effort, to contribute constructively to the learning of other students, and to always behave ethically and with civic concern. Students should come to class meetings prepared to discuss relevant topics. Exercises, quizzess, and assignments are to be completed by their due dates. The ideas and contributions of others should be appropriately cited. (This includes ideas and contributions garnered from readings, online resources, presentations, conversations, and any other source.) Students are expected to bring educational obstacles to the instructor's attention as early as possible, so that such problems may be promptly resolved.

Academic Honesty

Learning can be greatly facilitated by interactions between course participants, and these interactions are encouraged. Students should feel free to discuss lecture topics, readings, and even programming assignment concepts with each other. The actual completion of programming assignments and other exercises (including quizzes and examinations), however, should be conducted on an individual basis. All assignments and exercises submitted for evaluation should reflect the understanding and effort of the individual participant. *Not a single line of computer code should be shared between course participants.* If there is ever any doubt concerning the propriety of a given interaction, it is the student's responsibility to approach the instructor and clarify the situation *prior* to the submission of work results. Also, helpful conversations with fellow students, or any other person (including members of the teaching team), should be explicitly mentioned in submitted assignments. Failure to appropriately cite sources is called *plagiarism*, and it will not be tolerated!

Indeed, any infraction of the campus academic honesty policy must be reported to the campus Office of Student Conduct. In addition, infractions will be reported to the School of Engineering, following the Computer Science & Engineering Department Policy on Academic Honesty. (Note that these reports will occur regardless of the majors of the students involved.) This latter policy mandates particular penalties for infractions:

> If it is determined that a student has cheated, plagiarized, or otherwise violated the AHP [Academic Honesty Policy], the student will receive a 0 (or equivalent grade) for the assignment. ...
>
> The School of Engineering keeps a record of all infractions reported by its faculty. If upon receiving a notification it is determined that the student has one or more prior violations of the AHP, the School will inform the instructor who reported the new violation. The additional violation will immediately lead to a failing grade (F) for the course. The student will be informed in writing and will not be allowed to withdraw from the class. According to CSE Policy, students should note that even the first infraction in a class may lead to a failing grade if after reporting it is determined that the student had been previously sanctioned for one or more infractions in other classes. ...

In summary, any discovered act of academic dishonesty will be addressed harshly, as specified by campus, school, and departmental policies. It is hoped that the supportive nature of the class learning community will remove temptations to act without integrity.

Evaluations

The teaching team will provide feedback on assigned work submitted for evaluation by the corresponding deadline. Those students who are to receive grades for this course will have their work assessed approximately as follows:

Pretest	1%	
Programming Assignment #0	4%	
Programming Assignments #1 – #4	10%	each
15 Weekly Comprehensive Quizzes	2%	each
Laboratory Exercises	5%	total
Final Examination	20%	

Student performance will be evaluated in comparison to that of other students, both past and present.

Programming assignments are to be submitted for evaluation through the "Assignments" mechanism in the class `CatCourses` site. Such submissions must be received by the specified due date and time. Late assignments will not be evaluated, and they will receive *no credit*, even if they are substantially complete and correct. An "early submission deadline" will be provided for each programming assignment, however, and assignments received prior to that deadline will receive *extra credit* for their timeliness. Also, submitted programs that crash or do not run under the laboratory PyCharm environment will not be evaluated and will receive *no credit*.

While many laboratory meetings will provide time to receive assistance on programming assignments, some of these sessions will also introduce additional required exercises. These exercises are designed to assist with the acquisition of relevant skills, with a focus on Python programming skills, control system design skills, and knowledge engineering skills. Note that the collection of laboratory exercises assigned throughout the semester carries about half the weight of a computer programming exercise when determining final course grades.

Information about university course grading policies is available from the Registrar's office:

`https://registrar.ucmerced.edu/policies/grades`

Efforts will be made to provide students with regular feedback on their performance, relevant to the assignment of course grades, in the midst of the semester.

Schedule

In the schedule that appears on the following pages, a check mark ($\surd$) identifies an assigned reading for the date in question. A heart symbol ($\heartsuit$) identifies a reading that may be considered supplementary and optional.

Introduction

August 26 : Course Logistics

- √ Course Syllabus

August 31 : Introduction

- √ *AIMA*, §1.1.

September 2 : History

- √ *AIMA*, §1.2 – 1.5.

September 7 : Mathematics Review *(Video Lecture)*

- • *No Meeting*
- √ *AIMA*, Appendix A.
- √ *AIMA*, Appendix B.

September 9 : Agents

- √ *AIMA*, Chapter 2.
- ♡ Braitenberg, V. (1984) *Vehicles: Experiments in Synthetic Psychology.* MIT Press: Cambridge, MA.

September 14 : Philosophy of AI

- √ *AIMA*, §27.1 – 27.2.

September 16 : AI Ethics

- √ *AIMA*, §27.3.
- √ "March of the machines", *The Economist*, June 25, 2016. (available on `CatCourses`)

Search

September 21 : Uninformed Search

- √ *AIMA*, §3.1 – 3.4.

September 23 : Heuristic Search

- √ *AIMA*, §3.5 – 3.6.

September 28 : Local Search, Optimization, & Evolutionary Algorithms

- √ *AIMA*, §4.1 – 4.2.

September 30 : Search in Adversarial Games

- √ *AIMA*, §5.1 – 5.3, 5.7.

Knowledge Representation & Logic

October 5 : Propositional Logic

√ *AIMA*, §7.1 – 7.4.

October 7 : Reasoning With Propositional Logic

√ *AIMA*, §7.5.

October 12 : Effective Reasoning Algorithms

√ *AIMA*, §7.6 – 7.7.

October 14 : First-Order Logic

√ *AIMA*, §8.1 – 8.2.

October 19 : Knowledge Engineering

√ *AIMA*, §8.3 – 8.4.

October 21 : Unification & Chaining

√ *AIMA*, §9.1 – 9.4.

October 26 : Resolution Theorem Proving

√ *AIMA*, §9.5.

October 28 : Ontological Engineering

√ *AIMA*, Chapter 10.

Uncertainty

November 2 : Probability Theory & Decision Theory

√ *AIMA*, Chapter 12.

November 4 : Bayesian Belief Networks

√ *AIMA*, §13.1 – 13.4.

Learning

November 9 : Learning Decision Trees

- √ *AIMA*, §19.1 – 19.3.

November 11 : Linear Regression

- √ *AIMA*, §19.4, 19.6.

November 16 : Artificial Neural Networks

- √ *AIMA*, §21.1 – 21.2.

November 18 : Deep Learning

- √ *AIMA*, §21.3 – 21.7.

November 23 : Deep Learning Applications

- √ *AIMA*, §19.9.
- √ *AIMA*, §21.8.

November 25 : Thanksgiving Day Holiday

- *No Meeting*

November 30 : Reinforcement Learning

- √ *AIMA*, 22.1 – 22.3.

December 2 : Deep Reinforcement Learning

- √ *AIMA*, 22.4.

Conclusion

December 7 : Course Summary

- √ *AIMA*, Chapter 28.

December 9 : Final Examination Review

- √ Review Notes (available on `CatCourses`)

December 16 : Final Examination

- The final examination is scheduled for 8:00 A.M. to 11:00 A.M. on this day.