# CS-0263: Artificial Intelligence



## COURSE INFORMATION

**Instructor(s):**

**Lee Spector**
lasCCS@hampshire.edu
Office Extension: x5352
Office Hours:
Regular office hours: Tuesdays 10:00-11:30, Wednesdays 1:00-2:30, and Thursdays 10:00-11:30. Other times can be set up by arrangement (in person or via email). Sign up for regular office hours, advising day meetings, and occasionally other signup times on Moodle here.

**TA(s):**

**Edward Pantridge**
erp12@hampshire.edu

**Term:**

2015F

**Meeting Info:**

| | |
|---|---|
| **Tuesday** | 02:00 PM - 03:20 PM Adele Simmons Hall (ASH) 126 |
| **Thursday** | 02:00 PM - 03:20 PM Adele Simmons Hall (ASH) 126 |

**Description:**

Artificial Intelligence is a branch of computer science concerned with the development of computer systems that "think." In this course we will explore the core ideas of artificial intelligence through readings, presentations, discussions, and hands-on programming activities. A range of practical artificial intelligence techniques will be covered, and students will complete programming projects to demonstrate engagement with the themes of the course. Prerequisite: One programming course (in any language).

**Course Objectives:**

- To gain familiarity with a range of concepts and computational techniques that have been developed by AI researchers over history of the field.
- To apply several of these concepts and techniques to hands-on research and development activities involving "intelligent agents" in virtual worlds.
- To develop skills in "functional"-style programming.
- To conduct independent programming-based project work.
- To develop project presentation skills.
- To work collaboratively with classmates.

**Evaluation Criteria:**

You will be evaluated on the basis of attendance, participation, a portfolio of code and text submitted in five increments with scheduled deadlines, and in-class “demo” presentations on assignment due dates. You should demonstrate through your participation, your code, and your text that you have read and thought about the course readings. Demos will be strictly limited to 3 minutes, with 1 minute set-up time.

You should be certain before each demo session that you can get all of the necessary files in place and begin your demo within 1 minute of the start of your demo time. You should also be certain that you can complete your demo within 3 minutes after setup. Your portfolio should demonstrate facility with the code environment used in the class and engagement with several of the class topics at the implementation level.

## Hampshire Students (who get narrative evaluations)

Any missed, late, or inadequate assignments or demos will be noted in your evaluation. If you fail to submit 2 or more assignments, or to present 2 or more of the demos, then you should not expect to receive an evaluation.

## Five College Students (who get grades)

Each assignment will be graded on a scale from 0-100. Each demo will also be graded on a scale from 0-100. Attendance and participation will also be graded on a scale from 0-100. For your final grade I will calculate:

score = (0.4 * assignment average) + (0.4 * demo average) + (0.2 * attendance and participation)

I will then assign grades as follows:

| score | grade |
|---|---|
| >=97 | A+ |
| >=93 | A |
| >=90 | A- |
| >=87 | B+ |
| >=83 | B |
| >=80 | B- |
| >=77 | C+ |
| >=73 | C |
| >=70 | C- |
| >=67 | D+ |
| >=63 | D |
| >=60 | D- |
| <60 | F |

There will be no curve. Plusses and minuses will be given only if your home institution allows them. Note that missed demos will be scored as zero and will have a dramatic negative impact on your grade. A clearly inadequate demo may also be scored as a zero or nearly zero.

**Additional Info:**

---

**Division I Distribution Credit**

Successful completion of this course satisfies the Divison I distribution requirement in Mind, Brain, and Information. This course provides opportunities for satisfaction of Division I cumulative skills requirements in Quantitative Skills and Independent Work.

---

**Texts**

We will read parts of:

- *The Quest for Artificial Intelligence: A History of Ideas and Achievements*, by Nils J. Nilsson. Cambridge University Press., 2010. Available free online.
- *Artificial Intelligence: Foundations of Computational Agents*, by David Poole and Alan Mackworth, Cambridge University Press, 2010. Available free online.
- The ClojureBridge curriculum.

Additional readings are linked to the schedule below, and others may be distributed in class.

---

**Software**

The Clojure programming language, accessed primarily via:

- Nightcode (ClojureBridge Getting Started with Clojure using Nightcode)

---

**Additional Resources**

AITopics (Hub for additional information about AI)
Clojinc (Lee's introduction to Clojure)
Clojure from the Ground Up 1, 2, 3, 4, 5, 6, 7, 8 (a sequence of blog posts introducing Clojure)
Clojure Videos
Decomplecting Clojure (a blog post describing Clojure principles)
Leiningen (command-line tool for handling and running Clojure projects)
Counterclockwise (a Clojure IDE based on Eclipse)
Gorilla-repl (a Clojure worksheet environment)
Introduction to Clojure (tutorial on creativeapplications.net)
Try-clj (a browser-based Clojure repl)
Clojure cheatsheet (the official one from clojure.org)
Grimoire (Clojure documentation)
Clojuredocs (Clojure documentation)
Clojure Atlas (Clojure documentation)
Clojure Style Guide
4clojure (an interactive site for learning Clojure via problems)
Clojure CodeWars (an interactive site for learning Clojure via problems)
Planet Clojure (a Clojure meta blog)
Clojure reddit (a site for Clojure discussions)
Clojure google group (a site/list for Clojure discussions)
G+ Clojure community
clj-ml (machine learning library for Clojure built on top of Weka and friends)
lspector/gp (Clojure code for a simple genetic programming system, for demonstration purposes)
lspector/clojush (The Push programming language and the PushGP genetic programming system implemented in Clojure.)

---

**Facilities**

Students may use their own computers and/or the Macs in ASH 126, which will be available at various posted hours. Students should **not** expect files left on the Macs in ASH 126 to persist; **the discs on those machines may be erased without notice at any time**. Students may find it convenient to use a thumb drive to transport files to and from class.

---

**Difficulty/Level**

This course is intended to serve students with a wide range of backgrounds, including students with only one previous programming course (in any language) and students with significant computer science and programming experience. Students with little previous experience should resist being intimidated by the more difficult readings, etc., and bear in mind that I take background into account in writing evaluations. If a reading or a class discussion is over your head, try to extract the gist of it (which may be all that you need at this point in your learning) and talk to me if you want to understand more. Students with extensive previous experience should note that the class is structured to provide ample opportunities for more advanced work; feel free to talk to me about ideas for projects, etc.

**Demonic Coding**

Many class sessions will be dedicated in part or entirely to "demonic coding." Each student must have access to his/her current work files every day -- on a laptop computer, or a thumb drive, or a networked server, etc. -- and always be ready to participate as a coder in a demonic coding session.

**Policies in Regards to Illness, Epidemic, or Pandemic**

If you have a fever, please stay home, take good care of yourself, and contact me by email or phone. When you are able to work at home you should be able to participate in classes and to submit work electronically. If your illness makes it impossible for you to meet the course deadlines then contact me and we will negotiate an accommodation.

**Adaptations and Accommodations**

If you need course adaptations or accommodations because of a disability, or if you have a medical condition that may impact your performance or participation in this course, then please let me know. If you have approved accommodations then please go to Accessibility Services in CASA/Lemelson Center to pick up Letters of Accommodation to facilitate a proactive discussion about reasonable accommodations for this course. If you have documented disabilities but have not already already contacted Accessibility Services are encouraged to so. Accessibility Services can be contacted via email: Accessibility@hampshire.edu, via phone: 413-559-5498, or in person: Lemelson Center (CASA entrance).

**Plagiarism Policy**

Official policy text:

> All Hampshire College students and faculty, whether at Hampshire or at other institutions, are bound by the ethics of academic integrity. The entire description and college policy can be found in Non Satis Non Scire at handbook.hampshire.edu under Academic Policies/Ethics of Scholarship. Plagiarism is the representation of someone else's work as one's own. Both deliberate and inadvertent misrepresentations of another's work as your own are considered plagiarism and are serious breaches of academic honesty and integrity. All sources used or consulted in the process of writing papers, examinations, preparing oral presentations, course assignments, artistic productions, and so on, must be cited. Sources include material from books, journals or any other printed source, the work of other students, faculty, or staff, information from the Internet, software programs and other electronic material, designs and ideas.
>
> All cases of suspected plagiarism or academic dishonesty will be referred to the Dean of Advising who will review documentation and meet with student and faculty member. Individual faculty, in consultation with the Dean of Advising, will decide the most appropriate consequence in the context of the class. This can range from revising and resubmitting an assignment to failing the course. Beyond the consequence in the course, CASA considers first offenses as opportunities for education and official warning. Multiple or egregious offenses will have more serious consequences. Suspected instances of other breaches of the ethics of academic integrity, such as the falsification of data, will be treated with the same seriousness as plagiarism and will follow the same process.

In this course we will often be sharing and borrowing code. This is an important aspect of the course and an important aspect of modern programming practice. This does not mean, however, that it is acceptable to submit code that is not your own without acknowledging sources. Sources should be clearly and explicitly provided in everything that you produce.

**Schedule**

The following is only an approximate schedule and it is subject to change. Adjustments will be announced in class. Assigned readings should be read *prior* to the indicated classes.

| Tuesday 2:00-3:20 PM | Thursday 2:00-3:20 PM |
|---|---|
| | **September 10**<br>**Introduction to CS263** |

| | |
|---|---|
| | In class: introductions, syllabus, overview |
| **September 15**<br>**Installfest**<br>Before class:<br>• Read:<br>◦ Why would someone learn Clojure?<br>◦ Getting Started with Clojure using Nightcode<br>◦ Nilsson Preface, 1 (Dreams and Dreamers)<br>◦ Spector, Comments on *Blade Runner*<br>In class:<br>• Install Nightcode<br>• Experiment with the Nightcode REPL and Clojinc | **September 17**<br>**Introduction to Clojure**<br>Before class:<br>• Read:<br>◦ ClojureBridge curriculum (work your way through the curriculum slides, but ignore references to Lighttable and InstaREPL and only try to run the examples in Nightcode if you are ambitious)<br>• Watch **one of**:<br>◦ Rich Hickey on Clojure for Java Programmers (Part 1, Part 2)<br>◦ Rich Hickey on Clojure for Lisp Programmers (Part 1, Part 2)<br>In class:<br>• Clojinc |
| **September 22**<br>**Demonic Coding**<br>Before class:<br>• Read:<br>◦ Nilsson 2 (Clues)<br>◦ Beating the Averages<br>In class:<br>• Demonic Coding | **September 24**<br>**Portfolios/Demos**<br>Due: Assignment #1:<br>• Submit a portfolio containing:<br>◦ Original Clojure code (for any purpose).<br>◦ 1-2 page reading reaction paper; Prompt: What interests you most about AI and why?<br>◦ A README file describing each element in your portfolio.<br>In class:<br>• Demos |
| **September 29**<br>**Behavior-Based Robots**<br>Before class:<br>• Read:<br>◦ Nilsson 23.2.3 (Autonomous Vehicles), 25.4 (About Behavior)<br>◦ Pucks README<br>• Optional:<br>◦ Watch the film *Fast, Cheap, and Out of Control*<br>In class:<br>• Pucks ai.world1<br>• Demonic Coding | **October 1**<br>**Hierarchical Control**<br>Before class:<br>• Read:<br>◦ Poole & Mackworth 2 (Agent Architectures and Hierarchical Control)<br>• Watch:<br>◦ Videos of worlds 1-8<br>In class:<br>• Pucks ai.world2<br>• Demonic Coding |
| **October 6**<br>**Demonic Coding**<br>In class:<br>• Demonic Coding | **October 8**<br>**Portfolios/Demos**<br>Due: Assignment #2:<br>• Submit a portfolio containing:<br>◦ All previously submitted code and new, original Clojure code for controlling Pucks in ai.world1 and ai.world2.<br>◦ 1-2 page reading reaction paper; Prompt: What are behavior-based and hierarchical control architectures good for? What are their limits? |

| | |
|---|---|
| | ◦ A README file describing each element in your portfolio.<br><br>In class:<br>• Demos |
| **October 13**<br>***OCTOBER BREAK - NO CLASS*** | **October 15**<br>**Search**<br><br>Before class:<br>• Read:<br>◦ Poole & Mackworth 3 (States and Searching)<br>◦ Nilsson 12.1.1 (A∗: A New Heuristic Search Method)<br>• Watch:<br>◦ Videos of worlds 1-8<br><br>In class:<br>• Search |
| **October 20**<br>**Demonic Coding**<br><br>In class:<br>• Demonic Coding | **October 22**<br>**Planning**<br><br>Before class:<br>• Read:<br>◦ Nilsson 5 (Early Heuristic Programs)<br>◦ Poole & Mackworth 8 (Planning with Certainty)<br><br>In class:<br>• Planning |
| **October 27**<br>**Demonic Coding**<br><br>In class:<br>• Demonic Coding | **October 29**<br>**Portfolios/Demos**<br><br>Due: Assignment #3:<br>• Submit a portfolio containing:<br>◦ All previously submitted code and new, original Clojure code for controlling Pucks in more complex worlds (ai.world3-ai.world8) **and/or** new, original Clojure code applying AI search algorithms to problems of your choosing.<br>◦ 1-2 page reading reaction paper; Prompt: How do AI search and planning methods compare to human problem-solving methods? What do you expect their limits to be?<br>◦ A README file describing each element in your portfolio.<br><br>In class:<br>• Demos |
| **November 3**<br>**Logic**<br><br>Before class:<br>• Read:<br>◦ Nilsson 11.1-11.3 (Deductions in Symbolic Logic, The Situation Calculus, Logic Programming) | **November 5**<br>***ADVISING DAY - NO CLASS*** |

| | |
|---|---|
| ◦ Poole &Mackworth 5 (Propositions and Inference)<br><br>In class:<br>• core.logic<br>• Demonic Coding | |
| **November 10**<br>**Demonic Coding**<br><br>In class:<br>• Demonic Coding | **November 12**<br>**Knowledge**<br><br>Before class:<br>• Read:<br>◦ Nilsson 6 (Semantic Representations)<br>◦ Poole & Mackworth 13 (Ontologies and Knowledge-Based Systems)<br>◦ Optional: Wikipedia article on Deep Learning<br><br>In class:<br>• Knowledge |
| **November 17**<br>**Portfolios/Demos**<br><br>Due: Assignment #4:<br>• Submit a portfolio containing:<br>◦ All previously submitted code and new, original Clojure code for controlling Pucks in more complex worlds (ai.world3-ai.world8) **and/or** new, original Clojure code applying core.logic to problems of your choosing.<br>◦ 1-2 page reading reaction paper; Prompt: How do you see logic programming influencing future developments in AI? What specific ideas do you have for applying ontologies in AI?<br>◦ A README file describing each element in your portfolio.<br><br>In class:<br>• Demos | **November 19**<br>**Evolution**<br><br>Before class:<br>• Read:<br>◦ Nilsson 25.6 (Simulating Evolution)<br>◦ Spector Evolution of Artificial Intelligence<br>◦ Poole & Mackworth 4 (Features and Constraints); **Note**: It is okay to skim most of this chapter, with the exception of 4.9 (Population-Based Methods), which you should read with care.<br><br>In class:<br>• Genetic Programming |
| **November 24**<br>**Philosophy**<br><br>Before class:<br>• Read:<br>◦ Turing Computing Machinery and Intelligence<br>• Listen:<br>◦ Spector Rethinking Computer Intelligence<br><br>In class:<br>• Discussion | **November 26**<br>***THANKSGIVING - No Class*** |
| **December 1**<br>**Demonic Coding**<br><br>In class:<br>• Demonic Coding | **December 3**<br>**Future and Implications**<br><br>Before class:<br>• Read:<br>◦ Poole & Mackworth 15 (Retrospect and Prospect)<br>◦ Future Progress in Artificial Intelligence: A Survey of Expert Opinion<br>• Watch:<br>◦ Barrat: Book Discussion on Our Final Invention<br>◦ Stephen Hawking explains killer robots to John Oliver |

| | |
|---|---|
| | • Optional:<br>  ◦ Listen: Robots! (and Artificial Intelligence)<br><br>In class:<br>• Discussion |
| **December 8**<br>**Portfolios/Demos**<br><br>Due: Assignment #5:<br>• Submit:<br>  ◦ All previously submitted code and a new, substantial revision of at least one of your previously submitted code projects.<br>  ◦ 1-2 page reading reaction paper; Prompt: How do you see logic programming influencing future developments in AI? What specific ideas do you have for applying ontologies in AI?<br>  ◦ A README file describing each element in your portfolio.<br><br>In class:<br>• Demos | **December 10**<br>**Demos**<br><br>In class:<br>• Demos, continued |