

Fall 2015 Syllabus

This syllabus is subject to change. In particular, the midterm dates will not be finalized until a week or so into the course.

Day	Topic	Reading	Assignment	Due
Th 8/27	Introduction to AI: Past, Present, Future	Ch. 1, 26.3, 27.4	P0: Tutorial	F 9/4, 5pm
Tu 9/1	Agents and Environments	Ch. 2	HW1	M 9/7, 11:59pm
Th 9/3	Uninformed Search	Ch. 3.1-4		
Tu 9/8	Informed Search and Heuristics	Ch. 3.5-6	HW2	M 9/14, 11:59pm
Th 9/10	Local Search and Agents	Ch. 4	P1: Search and Games	F 9/25, 5pm
Tu 9/15	Game Playing	Ch. 5.1-5	HW3	T 9/22, 11:59pm

Day	Topic	Reading	Assignment	Due
Th 9/17	Constraint Satisfaction Problems	Ch. 6.1, 6.3-5		
Tu 9/22	Propositional Logic: Semantics and Inference	Ch. 7.1-5, 7.6.1	HW4	M 9/28, 11:59pm
Th 9/24	Logical Agents	Ch. 7.7	P2: A Logical Planning Agent	F 10/16, 5pm
Tu 9/29	First Order Logic	Ch. 8.1-3, 9.1	No HW	
Th 10/1	MIDTERM		Practice Midterm (solutions)	T 9/29, 12:30pm
Tu 10/6	Probability	Ch. 13.1-5	HW5	W 10/14, 11:59pm
Th 10/8	Bayes Nets: Syntax and Semantics	Ch. 14.1-3		

Day	Topic	Reading	Assignment	Due
Tu 10/13	Bayes Nets: Exact Inference	Ch. 14.3	HW6	W 10/21, 11:59pm
Th 10/15	Bayes Nets: Approximate Inference	Ch. 14.4	P3: An HMM-based Agent	F 10/30, 5pm
Tu 10/20	Markov Models	Ch. 15.1-3, 22.1	HW7	W 10/28, 11:59pm
Th 10/22	Dynamic Bayes Nets and Particle Filtering	Ch. 15.5		
Tu 10/27	Decision Theory	Ch. 16.1-3, 16.5-6	HW8	W 11/4, 11:59pm
Th 10/29	Markov Decision Processes I	Ch. 17.1	P4: Decision-making and Learning Agent	F 11/13, 5pm
Tu 11/3	Markov Decision Processes II	Ch. 17.2-3	HW9	W 11/11, 11:59pm

Day	Topic	Reading	Assignment	Due
Th 11/5	Reinforcement Learning I	Ch. 21.1-3		
Tu 11/10	ML: Classification and Regression	Ch. 18.1-4, 18.6	HW10	W 11/18, 11:59pm
Th 11/12	ML: Neural Networks	Ch. 18.7	P5: Learning and Classification	F 12/4, 5pm
Tu 11/17	ML: Statistical Learning	Ch. 20	HW11	W 11/25, 11:59pm
Th 11/19	Reinforcement Learning II	Ch. 21.4-5		
Tu 11/24	Advanced Applications: Natural Language Processing	Optional: Ch. 23	No HW	
Th 11/26	Thanksgiving Break			

Day	Topic	Reading	Assignment	Due
Tu 12/1	Advanced Applications: Computer Vision	Optional: Ch. 24	No HW	
Th 12/3	Advanced Applications: Robotics & Conclusion	Optional: Ch. 25		
F 12/18	FINAL EXAM (8-11am)			

TIMES AND LOCATIONS

- **Lecture:** Tuesdays and Thursdays 12:30-2 pm in 155 Dwinelle
- **Discussion Sections:** See "Sections" tab above. You are expected to attend discussion sections every week (it is a significant portion of your participation score). You may change discussion sections up until Tu 9/22. In the case of an over-crowded section, priority goes to students that are officially registered for that section in Telebears.
- **Exams:**
 - Midterm: Thursday, October 1, 12:30-2 pm, 155 Dwinelle (in class)
 - Final: Friday, December 18, 8-11 am, Location TBD

COMMUNICATION

- Announcements will be posted on Piazza.

- The main mode of electronic communication between students and staff, as well as amongst students, will be through Piazza. It is intended for general questions about the course, clarifications about assignments, student questions to each other, discussions about material, and so on. We strongly encourage students to participate in discussion, ask, and answer questions through this site. The course staff will monitor discussions closely.
- If you need to contact the course staff privately, please make a private post on Piazza, so we can make sure to follow-up with you. If needed, you may email cs188-staff AT lists.berkeley.edu or contact the instructor or GSIs directly.

COURSE DESCRIPTION

Ideas and techniques underlying the design of intelligent computer systems. Topics include search, game playing, knowledge representation, inference, planning, reasoning under uncertainty, machine learning, robotics, perception, and language understanding.

You will learn to build intelligent agents - systems that perceive and act - for fully observable, partially observable and adversarial settings. Your agents will generate provably successful plans in deterministic environments, draw inferences in uncertain environments, and optimize actions for arbitrary reward structures. They will learn from observation and from rewards. The techniques you learn in this course apply to a wide variety of artificial intelligence problems and will serve as the foundation for further study in any application area you choose to pursue.

See the syllabus for slides, deadlines, and the lecture schedule.

PREREQUISITES

- **CS 61A or 61B:** Prior computer programming experience and a good understanding of clean, elegant implementations of nontrivial algorithms and data structures.
- **CS 70 or Math 55:** Facility with basic concepts of propositional logic and probability are expected, as well as the ability to understand and construct proofs. CS 70 is the better choice for this course, but Math 55 could be

substituted. Some linear algebra and calculus will be used, but the necessary content will be covered in the course.

This course has substantial elements of both programming and mathematics, because these elements are central to modern AI. You should be prepared to review basic probability on your own if it is not fresh in your head. You should also be very comfortable programming on the level of CS 61B even though it is not strictly required.

CS61A AND CS61B AND CS70 is the recommended background.

LANGUAGE

Course programming assignments will be in Python. We do not assume that students have previous experience with the language, but we do expect you to learn the basics very rapidly. Project 0 is designed to teach you the basics of Python, but if you want to get a head start here is a good tutorial: [ACM Python Tutorial](#)

ASSIGNMENTS

This class includes **five programming projects** and **regular written and electronic assignments**.

Collaboration: Project 0 is to be completed alone. Projects 1 through 5 can be completed alone or in teams of two. When completing in a team of two it is important that both team members submit their team's project files into edX to ensure a grade is registered for both team members. Homework is to be submitted individually but may be discussed in groups.

Slip days: Programming projects must be turned in electronically by 5:00pm (unless noted otherwise) on the listed due date. You will have a total of **5 slip days** for these projects, up to **two of which can be used for each project**. Note that slip days are counted at the granularity of days, rounded up to the nearest day. For example, for a project due at 5pm on Friday, any submission from Friday 5pm - Saturday 5pm will use up one slip

day, any submission from Saturday 5pm - Sunday 5pm will use up two slip days, and any submissions after Sunday 5pm will receive no credit. The other homework assignments do not have late days!

Project grading: Projects will by default be graded automatically for correctness, though we will review projects individually as necessary to ensure that they receive the credit they deserve. To calculate your total project grade, we will consider all possible allocations of slip days across all projects to give you the best score achieved. For example, suppose you submitted a 24/25 on Project 1 on time, and then resubmitted a 25/25 on Project 1 after using two slip days. If by Project 5, you used up all your slip days, but you need to use an additional slip day on Project 5 and wish you had not submitted two days late for the one point in Project 1, you can submit into the late module for Project 5 (edX does not block you from going over the number of slip days) and we will automatically redistribute your slip days usage.

Projects will also be manually inspected through the project check off process. You will meet with your GSI to review and answer questions about your project code for select projects.

Participation: Participation in the course is worth 1% of your final grade. Participation primarily refers to active engagement in discussion sections (and will be reported by your section GSI), but also includes active involvement during lecture, on piazza, and in office hours.

Ethics: Submissions should acknowledge all collaborators and sources consulted. All code and written responses should be original. We trust you all to submit your own work, but to protect the integrity of the course from anyone who doesn't want to play by the rules, we will actively be checking for code plagiarism (both from current classmates and previous semesters). I'm not lenient about cheating; I sympathize with [Kris Pister's policy](#).

GRADING

Overall grades will be determined from:

- Programming Assignments (25%)
- 145 points total
- 5 projects, 25 points each
- 2 project check-offs, 10 points each
- Weekly Homework Assignments (15%)
- 15% = 10% written + 5% edX

- Midterm Exam (20%)
- Final Exam (40%)
- Participation (1%)

Grades are on the following fixed scale:

A	[90 – 101]%
A-	[85 – 90)%
B+	[80 – 85)%
B	[75 – 80)%
B-	[70 – 75)%
C+	[65 – 70)%
C	[60 – 65)%

C-	[55 – 60)%
D+	[50 – 55)%
D	[45 – 50)%
D-	[40 – 45)%
F	[0 – 40)%

These cutoffs represent grade minimums. We may adjust grades upward based on exceptional class participation, extra credit, etc. The grade of A+ will be awarded at the professor's discretion based on exceptional performance.

Regrade Policy: If you believe an error has been made in the grading of one of your exams or assignments, you may resubmit it for a regrade. Regrade requests must include a detailed explanation of which problems you think we marked incorrectly and why. Regrades for cases where we mis-applied a rubric in an individual case are much more likely to be successful than regrades that argue about relative point values within the rubric, as the rubric is applied to the entire class. Because we will examine your entire submission in detail, your grade can go up or down as a result of a regrade request.

TEXTBOOK

The required text is [Artificial Intelligence: A Modern Approach](#) by Stuart Russell (UC Berkeley) and Peter Norvig (Google).