NORTHERN ARIZONA UNIVERSITY
College of Engineering, Forestry & Natural Sciences
School of Informatics, Computing, and Cyber Systems

# Course Syllabus

| CS 470 – Artificial Intelligence | | Spring 2018 |
|---|---|---|
| **Credits:** 3 credits | **Pre-reqs:** CS249 (Data Structures) with C or better | **Co-Reqs:** N/A |
| **Section#: 1** | **Co-convened with:** CS570 | **Cross-listed with:** N/A |

## Academic Catalog Description:

Introduces fundamental principles of artificial intelligence, including knowledge representation, planning, game playing, learning, and adaptive algorithms.

## Course Purpose:

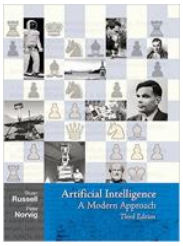
This course is a CS upper-division undergraduate elective, and is designed to provide an introduction to the specialized area of Artificial Intelligence. It is therefore suitable for students interested in pursuing graduate studies or industrial work in intelligent or self-adapting/learning systems, or those wishing to gain a broad understanding of the state of the art and programming approaches used in this rapidly growing area.

## Expected Student Learning Outcomes:

Upon successful completion of this course, students will be able to:

- Articulate the main areas of past and present activity within the broad domain of AI, including expert systems, game playing, machine learning, theorem-proving, and learning.
- Be able to trace the historical trajectory of AI development across key areas, including key milestones and remaining challenges in each.
- Describe and apply key computational techniques used in of AI systems, both on paper and in the Python programming language.
- Implement simple programming projects to explore basic AI techniques and algorithms.
- Identify AI techniques embedded within current software products, and speculate realistically on the application of AI in future software products.
- Implement one project that explores at least one AI technique in more detail, creating a system able to solve simple but realistic problems.

## Detailed Information for this offering




**Time and Location:** **9:35-10:50 TTH, Rm 224 Engineering (Bldg 69)**

**Readings and Materials:**

**Course Textbook:** *Artificial Intelligence: A Modern Approach*, Author: Stuart Russell and Peter Norvig. Publisher: Pearson

**Instructor's Name:** **Dr. Eck Doerry**

**Office Building/Room Number:** **SICCS Building (bldg. 90), Rm 217**

**Email:** **Eck.Doerry@nau.edu**

**Instructor Availability:**

| | |
|---|---|
| **Office hours:** | **Tues, Thurs from 14:00-15:00**<br>**Monday from 4:00-5:00**<br>We often change office hours by class vote during the term to accommodate your schedules better. Check the course website for current information! |
| **Other:** | • Although you should try hard to make it to scheduled office hours, I am also available at other times by appointment. To schedule, see me before or after class or send email. |

### Course Structure and Evaluation Mechanisms:

This course offering will consist of a number of elements: Lectures that meet three hours per week to present and discuss topics relating to Artificial Intelligence, readings from the textbook that support and expand on lecture themes, homework assignments that will sharpen students' understanding of the material, and programming challenges that provide a chance to apply concepts from lecture.

**Evaluation Mechanisms:** There are several mechanisms by which your course grade will be determined. Homeworks allow you to practice and develop the concepts and analytic skills that we discuss in lectures. Programming assignments allow you to gain hands-on experience in AI programming algorithms and techniques. Exams allow you to explore the extent to which you understand the material that has been presented.

**Homeworks:** Homeworks will be assigned for various topics that we cover, and give you a chance to apply some of the analytic approaches we've covered. All homework must be done individually.

**Programming:** Three to five programs will be assigned to more deeply explore some of the concepts discussed in class. Programming assignments must be done individually.

**Exams:** There will be two exams during the semester. Exams count much more than programs.

**Final Exam:** There will be a final exam that brings together what we've learned over the course of the semester. The final exam will consist of some combination of written exam and final project.

**Class participation:** Interaction -- both with the instructor and with other students -- is crucial for understanding and integrating the ideas presented in lecture. To emphasize this, a significant number of points are set aside to reward students who take an active role in the class. Participation points are also used to credit quizzes (in-class or on BBlearn) and other formative exercises.

## Grading System:

| **Weighting of Deliverables**:<br>The following percentages are used in weighting total points earned on programming, exams, and participation:<br>• Programming Assignments = 25%<br>• Homework and Quizzes= 20%<br>• Two midterms = 25%<br>• Participation/Quizzes = 5%<br>• Final Exam/Project = 25% | **Grading Scale**:<br>90-100% = A<br>80-89% = B<br>69-79% = C<br>55-69% = D<br>under 55% = F |
|---|---|
| Notes:<br>• Simply providing a mostly correct solution with little or no explanation will generally get you a "C". To get an "A" or a "B" score, you'll generally have to show "above average" or "excellent" mastery, which in this course is demonstrated by clear, complete analysis; demonstrated knowledge beyond factual basics; and robust, elegant implementations. | |

## Tentative Class Outline:

| Weeks | Topics | Readings |
|---|---|---|
| 1-2 | Introduction, Intelligent Agents, History | Ch.1 & 2 |
| 2-4 | Problems solving agents: state space search | Ch. 3 & 4 |
| | **Exam 1** (exact date TBD) | |
| 5-7 | Adversarial Search, Game Playing, Constraint probs | Ch.5, 6 |
| 8-10 | Logical Agents & reasoning (K-rep, 1st -order logics) | Ch. 7, 8, 9 |
| | **Exam 2** (exact data TBD) | |
| 11-13 | Reasoning under uncertainty | Ch. 13, 14 |
| 14 | Learning | Ch. 18 |
| 15-16 | Final Exam and demos | |

## Class Policies:

**Attendance:** You are responsible for all material covered during the lectures whether you attend or not. If you must miss a class, be sure to get the notes from another student. Late arrivals are very disruptive and are not acceptable --- plan to arrive five minutes before the start of class.

**Electronic Device usage:** All cell phones, PDAs, music players and other electronic devices must be turned off (or in silent mode) during lecture, and may not be used at any time. Laptops are allowed for note-taking only during lectures; no surfing or other use is allowed. I devote 100% of my attention to providing a high quality lecture; please respect this by devoting 100% of your attention to listening and participating.

**Late work and Make-ups:** Unless otherwise noted, all assigned work is due at the beginning of class on the date they are due. The following specific policies apply:

- **Quizzes:** No make-ups, no late work accepted. These will mostly be on BBlearn
- **Exams:** Make-ups only when scheduled/approved in advance or with proper documentation (note from physician, etc.), as required by NAU policy.
- **Programming assignments and written homeworks:** Late work is accepted, with appropriate deductions as specified in the Late Submissions Policy on the course website.

**Grade Challenges:** Although I try hard to grade as accurately and fairly as I can, mistakes do occur. If you feel that I owe you some points, or would like to discuss an evaluation, I encourage you to stop by office hours. To avoid loss of context, any grade disputes must be brought to my attention no later than 5 business days after the assignment was returned.

**Homework Submission and Format:** All homework where a packet of deliverables is required must:

- Have a cover page containing only the Your name and NAU ID number, the class name, the homework name, and the date of submission.
- Have problems and solutions presented in the order deliverables are listed in the assignment spec. All work must be typewritten, with the exception of figures and diagrams which could be hand-drawn, scanned, and included. Clear formatting and presentation counts; illegible work can not be graded.
- Hardcopy submissions must be stapled. No other fastening mechanisms (paper clips, dog-ears, etc.) are acceptable.

**Academic Dishonesty:** Cheating will not be tolerated and may result in serious sanctions, including immediate failure in the course. Serious incidents of academic dishonesty will also for brought to the attention of the university and may result in expulsion. All work in this class is meant to be an individual effort by the person receiving the grade. Any variation from this is considered cheating and all parties involved (giving or receiving) will be sanctioned.

## Other Important Course Information:

**Vital Skills and other Hints for success in CS470, Artificial Intelligence**

**Be active, not passive.**

One of the skills you should develop is an ability to read difficult material on your own - you will exercise this skill in the present course. Artificial Intelligence is a very broad ocean, with many fascinating sub-areas to explore. We will necessarily only have time to sample some subset of the chapters in the book…and will be moving along briskly to do that. Do NOT expect that I will lecture on every detail contained in the chapters; you **must** read the chapters before the lecture, and then come back later to deepen and refine the lecture discussion with material from the chapters. If you find this material as fascinating as I do, reading the chapters should be a knowledge-rich pleasure. In any case, you are responsible for material in any chapters (or sections of chapters) assigned in the course schedule.

**Strive for elegance.**

Those who have taken my courses before know that I'm a big fan of elegant solutions. This is especially true for programming challenges…and triply true for programming solutions in Artificial Intelligence. It is not a stretch to say that the way you factor up a programmed solution is a directed reflection of how well you really understand the concept you are trying to embody in your program. Take some extra time up front to really think it through so that you truly get what "tasks" your program will need to do…which you can then represent in your code (functions, objects, etc.). Careful thinking through of the problem up-front is the difference between three hours of programming where you just type in what you've planned, test it, and fix a few small things; and a marathon hacking session where you start off more or less blind, hit a roadblock, kludge around it, then repeat that ad infinitum until your code is an ungodly mess that you yourself can barely understand!

**Don't procrastinate.**

Don't delay working on homeworks and programming until the last minute. We are way past the days of CS126, where you could blow off the assignment until the night before it's due, then sit down and crank it out in a couple of hours. The programming problems in this course are designed to take student learners between 6-20 hours of work…depending on how carefully you plan and analyze the problem up front, versus hacking away at it. Leave time for exploring, failing, thinking, and learning. Unlike lower division courses, where the goal is always to simply hack up a solution to a narrow complete spec, the programming in this course is truly part of the thinking and learning process. And that just takes time…

## University Policies:

This course is conducted in accordance with all applicable university policies which can be found at: http://nau.edu/OCLDAA/_Forms/UCC/SyllabusPolicyStmts2-2014/ including Safe Environment, Students with Disabilities, Academic Contact Hours, Academic Integrity, Research Integrity, Sensitive Course Materials, and Classroom Disruption Policy. The student handbooks are also valuable resources for other policies. The undergraduate student handbook is at http://nau.edu/student-life/student-handbook/.