

**Complex Adaptive Systems**  
**UNM CS 423/523 Spring 2013**  
**MW 11:00 – 12:15**  
**Sara Reynolds Hall 101**  
**[cs.unm.edu/~melaniem/courses/CAS13.html](http://cs.unm.edu/~melaniem/courses/CAS13.html)**  
**email list: [cs523@cs.unm.edu](mailto:cs523@cs.unm.edu)**

**Professor:** Melanie Moses  
Email: [melaniem@cs.unm.edu](mailto:melaniem@cs.unm.edu)  
Office: Farris Engineering Center 327  
Office Hours: Monday 12:30-2:00pm  
Thursday 9:00-11:00am

**TA:** Matthew Fricke  
Email: [mfricke@cs.unm.edu](mailto:mfricke@cs.unm.edu)  
Office: Farris Engineering Center 323  
Office Hours: Wednesday 12:30-2:00pm  
Tuesday 8:30-10:30am

**Textbook:** We don't have a textbook for this course, but we'll use as a guide *Complexity, A Guided Tour*, Melanie Mitchell, Oxford University Press 2009. Each week we'll read journal articles to explore concepts from the book in more depth.

**Course Description:** A graduate level introduction to selected topics in complex adaptive systems. The course focuses on computational tools to simulate and measure complexity, and analysis of biological and computational complex adaptive systems. Topics include definitions of complexity, evolution and genetic algorithms, cellular automata, dynamical systems, scaling and fractals, ant colonies & ant colony optimization algorithms, immune systems & immune inspired computer security and swarm robotics.

**Course Assignments and Grading:**

The course requires extensive reading, participation in discussions and in-class exercises, attending lectures, and completing programming assignments and written reports. In addition to the primary textbook, students will read papers from the primary scientific literature or chapters from supplemental textbooks each week.

- 20% of the course grade will be based on class participation, including short pop quizzes to ensure that you have completed the reading, leading and participating in discussions of assigned readings and participation in in-class exercises. Graduate students will work in teams to present a paper to the class. The oral presentation (usually with slides) will include a paper summary and questions to facilitate discussion. Students must meet with the professor beforehand to review their presentation.
- 20% of the grade will be based on a midterm.
- 60% of the grade will come from three programming projects. The first project will require you to write simple programs, analyze data and present it in a report. You are strongly encouraged (but not required) to use Matlab for this assignment. For the second project you will modify genetic algorithms and a robot simulator written in C++. You will be responsible for designing your third project, which will be due on the final day of class and count as your final exam. You will work in teams for all assignments and document the contribution of each team member.

For each project you will turn in your code, a readme file describing how to run your code, and a report describing how your code works, results and analysis and answers to specific questions. Your grades will be based primarily on the quality of your reports which should be clear, concise, free of typos and grammatical errors and contain clear and meaningful figures. Your reports should indicate an understanding of relevant concepts covered in lectures, readings and discussions. You should spend at least as much time writing your report as writing your code.

Undergraduates will have (slightly) simpler assignments, will not lead paper discussions, but will be expected to complete all readings and participate in all discussions.

Projects turned in late will be penalized 10% for each late day. Students who have a true emergency must contact the professor before the due date. No exceptions.

**Academic dishonesty will not be tolerated.** If you cheat, you will fail the class. In collaborative work, the contributions of each student must be documented clearly in an author contributions section of the report. Your report must clearly document all downloaded code and how you have modified or incorporated it into your own code. Failure to document the source of any code that you did not write yourself constitutes cheating. Similarly, you must cite all journal articles, books, web pages and other online sources for your reports in a references section.

### **Course Topics**

**Introduction:** Definitions of complexity; dynamical systems, evolution (Jan 14 -30)

**Genetic Algorithms** (Feb 4 - 11)

**Cellular Automata** (Feb 13 - 20)

**Swarm Robotics** (Feb 20 – 27)

**Ants & Ant Colony Optimization** (Mar 4 – 6)

**Spring Break** (Mar 11 – 15)

**Midterm Review & Midterm** (Mar 18 - 20)

**Brains, Neural Nets & Analogies** (Mar 25 – 27)

**Natural and Computational Immunology** (Apr 1 – 3)

**Modeling & the Prisoner's Dilemma** (April 8)

**Networks, scaling & fractals** (Apr 10 - 17)

**Complexity Revisited** (Apr 22 – 24)

**Review & Final Project Due** (Apr 29 – May 1)

### **Tentative Due Dates**

Project 1: Monday February 4 at the START of class

Project 2: Friday March 7 by NOON

Midterm: Wednesday March 20 during class

Final Project Proposal: Wednesday April 3 at the START of class

Final Project: Wednesday May 1 at the START of class

**Assigned Readings** will be posted on the course web page.

Graduate students must sign up to present a paper listed on the web page.