CS 550 Spring 2019 / Schedule # 38081

Course and Office Hours
Class meets: Meets M/W 4:00-5:15 PM HT-183
Professor Marie Roch, GMCS-533, tel: 619 594 5830
e-mail: marie●roch a_t sdsu●edu (use period for ●), office hours posted at https://roch.sdsu.edu.

Course Overview
This is an elective course for computer scientists that teaches you the fundamentals of artificial intelligence. We will touch on a broad array of topics such as learning how an agent can search a problem space for sequences of actions that solve a problem (e.g. how to maneuver an autonomous vehicle), how certain problem configurations and actions can constrain the what can be done next (e.g. solving crossword puzzles), how to reason logically, and how machines can learn to predict categories from data.

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

- Construct intelligent agents capable of interacting with their environments.
- Understand and implement heuristic searches for problem solving and game playing (adversarial search)
- Understand first-order logic and its application to theorem proving.
- Analyze constraint satisfaction problems and resolve them through search.
- Understand and apply machine learning algorithms for classification tasks

Enrollment Information
Prerequisites: CS 108, Mathematics: Discrete Math (245) or Mathematical Logic (523)

Course Materials

Programming in this course is in Python 3. While we will briefly introduce Python in class, we will not be devoting much time to how to program in Python as upper division computer scientists should have the skills to learn new languages without an extraordinary effort. See the course web site for recommendations on optional books and videos on Python that are freely available to students online through the University’s Safari subscription.

Course Structure and Conduct
This is a lecture-based course. Assignments are a combination of problems, essay-like answers, and programs.

Relationship to CS Program Course Outcomes
CS 550 addresses the following CS Program course outcomes:
  a) An ability to apply knowledge of computing and mathematics
  b) An ability to analyze a problem, and identify and define the computing
requirements appropriate to its solution

c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

d) An ability to function effectively on teams to accomplish a common goal

h) Recognition of the need for and an ability to engage in continuing professional development

i) An ability to use current techniques, skills, and tools necessary for computing practice

j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

k) An ability to apply design and development principles in the construction of software systems of varying complexity.

Topics Covered: intelligent agents, theorem proving, search, machine learning, constraint satisfaction problems

**Course Assessment and Grading**

Materials are evaluated using coarse grading. It is very difficult to justly and systematically determine that one answer is worth N points and another is worth N±ε. Consequently, points are frequently assigned based upon broad categories that indicate the degree of mastery:

- A+: Excellent (E)
- B: Mostly right (MR)
- ~40%: Valiant effort (VE)
- A: Good (G)
- C: Right track (RT)
- F: Not much effort

Percentage contribution of assessments to your final grade:

**Exams (40% of grade):** There will be two closed book exams. The exams are non-cumulative, but the second will build on material covered in the first. Generally, I tend to emphasize concepts and you should expect short essay questions in addition to any short answer or quantitative questions. If you cannot attend an exam, contact me before the exam. No make-up exams will be permitted without prior approval.

**Assignments (60%):** Problem sets consist of questions and programming assignments. The number of problem sets varies from semester to semester, but there are typically five to six problem sets. Questions may be quantitative or qualitative in nature. For qualitative questions you are expected to give responses in grammatically correct complete sentences. For quantitative problems, you must show your work in order to receive credit.

*Note on programs:* Whether as part of a lab or problem set, programs must be well structured and commented. Frequently, students make their lives more difficult by simply writing the program without thinking first. The use of good program design will make your life significantly easier. In addition, an emphasis should be made on making your programs readable. Use meaningful variable names and comment as you write the
program rather than adding comments at the end. If you encounter difficulties in your projects, I will assist you provided that you have made a good-faith effort to resolve the problems first. Functionality may be evaluated automatically, be sure to follow specified interfaces and naming conventions.

Good design and structure are expected; programs are not assessed as excellent or good simply because they work. You are not expected to comment every line nor to comment obvious lines of code (e.g. \texttt{counter = counter + 1}), but you must provide enough documentation such that a reasonable programmer can easily follow your code. Failure to do so will result in the program earning a lower number of points. See the course FAQ for guidelines on appropriate program structure.

Assignments are due at the \textbf{beginning} of class on the date posted in the course calendar. Assignments may be submitted \textbf{up to one class period late with a penalty of 10\%}.

This course uses a coarse grading system. It is very difficult to justly and systematically determine that one answer is worth $N$ points and another is worth $N \pm \epsilon$. Consequently, points are assigned based upon broad categories that indicate your mastery of the concept:

- \textbf{A+} - Excellent
- \textbf{A} - Good
- \textbf{B} – Mostly right
- \textbf{C} – Right track
- \textbf{VE} – Valiant effort
- \textbf{F} – Unacceptable

Weighting of grades: Assignments 36\%, Exam I 32\%, Exam II 32\%

\textbf{Schedule}

A week by week schedule may be found on the course web site as well as a Google calendar that lists due dates for assignments. Typically, five to six problem sets will be assigned with deadlines varying from one to two weeks from assignment depending upon the complexity. Students are encouraged to use the pair programming paradigm. Details on pair programming will be discussed in course.

\textbf{No early finals will be given without and excuse}, so if you are booking plane tickets do not schedule a date before the final exam (Mon May 11\textsuperscript{th} from 3:30 to 5:30 in our normal room).

\textbf{Other Course Policies}

\textbf{Accommodations}: The learning environment should be accessible to all. SDSU provides reasonable accommodations in the following situations:

- 
  \textbf{Disability}: If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to contact the Student Ability Success Center at (619) 594-6473. To avoid any delay in the receipt of your accommodations, you should contact the center as soon as possible. Accommodations are not retroactive.
Religious and official university activities (e.g., athletics): Within the first two weeks of classes, notify your instructor of planned absences for religious or university activities. If scheduling changes occur, immediately notify the instructor.

**Your Well Being:** University life is stressful for many people. If you find yourself having problems, please feel free to avail yourself of SDSU’s Counseling and Psychological Services center (619) 594-5220. SDSU also offers services to Dreamers and other people with undocumented status.

**Academic Honesty:** You are free to discuss ideas and strategies for approaching problems with others, but students must complete work on their own. Using other people’s work in any form (i.e. the web, other students) will result in disciplinary action. **Plagiarism is unacceptable and will not be tolerated. You are responsible for understanding plagiarism**; the library has a tutorial ([https://sdsu.libwizard.com/f/Plagiarism-tutorial](https://sdsu.libwizard.com/f/Plagiarism-tutorial)). If you think that copying an answer from the web is okay, take the tutorial. If you think that paraphrasing with citations is okay, take the tutorial… ignorance is not a valid excuse. If you have any questions about plagiarism after taking the tutorial, I will be happy to assist you. Students who have plagiarized or cheated will be reported to the Center for Student Rights and Responsibilities. In addition to the academic penalty (0 on the assignment with no possibility of redemption on first offense, failing the course on the second), the university may decide upon additional sanctions such as expulsion.

**Classroom policy:** Turn cell-phones off before class and refrain from chatting during class as both disturb the students around you. If you find yourself unable to resist chatting, you will be asked once to be quiet. A second time will result in your being asked to leave for the day. If you need to work on assignments for another class, do it somewhere else.

**Conflicts/Issues:** Should you have any concerns about the course, please see me during my office hours or make an appointment and we will try to resolve the problem together. If you are not satisfied with the resolution after having discussed the issue with me, you may contact Dr. Shangping Ren.