CS 4300: Artificial Intelligence

Fall 2017 Syllabus

Course Description

Required of students pursuing a Computer Science degree or emphasis. Introduces the broad field of artifical intelligence in computer software followed by specific applications in computer gaming strategies. Students will complete programming assignments.

Prerequisites

CS 2420 and CS 2810 both with a C- or better

Course fees

Course fee: $25, used to assist in maintaining CIT infrastructure.

Disability Statement

If you suspect or are aware that you have a disability that may affect your success in the course you are strongly encouraged to contact the Disability Resource Center (DRC) located in the North Plaza Building. The disability will be evaluated and eligible students will receive assistance in obtaining reasonable accommodations. Phone # 435-652-7516.

Sections

1. MWF 8:00 - 8:50 am in Smith 117
   Final exam December 11 at 7:00 - 8:50 am

Instructor

Curtis Larsen

Objectives

The student will be able to discuss the principles of:

- problem solving using uninformed search,
- problem solving using informed search,
- problem solving using local search,
- propositional and first order logic,
- probability and probabilistic reasoning,
- machine learning,
- select topics in artificial intelligence.

The student will be able to:

- design problem definitions suitable for use in search algorithms,
- implement and use search algorithms,
- design and implement propositional logic to solve problems,
- design and implement baysian networks to solve problems,
- implement large software projects using third party libraries,
- work in large software projects with legacy code.

Resources

Textbook

There is one text for this course, available from the campus bookstore:

Computer Labs

You may use the computers and software in the Smith Computer Center. Some lab assistants may be able to help with assignments and pass off homework assignments for introductory courses.

The homework resources provided require the student to work in Linux. The 64-bit Ubuntu 16.04 version is required. This software is available in the lab. Students choosing to work in another setting will need to provide a version of this OS, with the GNU C++ tools for software development. Note that the Linux development environment inside Windows 10 as worked for previous students.

Course Web Site

Assignment submissions and grades will be managed in the Canvas System.

Assignments and Exams

Reading

The student is responsible for reading the material in the textbook. A reading schedule is provided with the class schedule on the course website. The student is expected to read the material before the class in which it is discussed. The book also includes material beyond what we will discuss in lecture, which you are encouraged to study on your own. Feel free to bring questions from the reading to lectures or to office hours.

Assignments

There will be assignment requirements due each week. Most assignment requirements will combine into larger projects to create software agents to perform rationally in a simulated environment.

Exams

There will be two written midterm exams scheduled in the testing center during the semester as shown in the class schedule.

There will be a final exam as scheduled during finals week. The final will include extensions to the course projects. The student is responsible for keeping working backups of all code submitted during the semester.

Grading

Assignments will count for 40% of your point total. The midterm exams will count for 30% of your point total (15% each). The final exam will count for 30% of your point total.

Letter grades are assigned based on the percentage of possible points attained, according to the following chart:

<table>
<thead>
<tr>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>94</td>
<td>A</td>
<td>84</td>
<td>B</td>
<td>74</td>
<td>C</td>
<td>64</td>
<td>D</td>
</tr>
<tr>
<td>90</td>
<td>A-</td>
<td>80</td>
<td>B-</td>
<td>70</td>
<td>C-</td>
<td>60</td>
<td>D-</td>
</tr>
<tr>
<td>87</td>
<td>B+</td>
<td>77</td>
<td>C+</td>
<td>67</td>
<td>D+</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

Course Policies

Attendance

Students are responsible for material covered and announcements made in class. School-related absences may be made up only if prior arrangements are made. The class schedule presented is approximate. The instructor reserves the right to modify the schedule according to class needs. Changes will be announced in class and posted to the website. Exams and quizzes cannot be made up unless arrangements are made prior to the scheduled time.
**Time Commitment**

Courses should require about 45 hours of work per credit hour of class. This class will require about 135 hours of work on the part of the student to achieve a passing grade, which is approximately 9 hours per week. If you do not have the time to spend on this course, you should probably rethink your schedule.

**Late Policy**

Each assignment has two due dates. The earliest due date is the required date. The second date is the absolute latest date to submit the assignment. Late work will not be accepted after the second date.

**Collaboration**

Limited collaboration with other students in the course is permitted. Students may seek help learning concepts and developing programming skills from whatever sources they have available, and are encouraged to do so. Collaboration on assignments, however, must be confined to course instructors, lab assistants, and other students in the course. Students are free to discuss strategies for solving programming assignments with each other, but this must not extend to the level of programming code. Each student must code his/her own solution to each assignment. See the section on cheating.

**Cheating**

Cheating will not be tolerated, and will result in a failing grade for the students involved as well as possible disciplinary action from the college. Cheating includes, but is not limited to, turning in homework assignments that are not the student’s own work. It is okay to seek help from others and from reference materials, but only if you learn the material. As a general rule, if you cannot delete your assignment, start over, and re-create it successfully without further help, then your homework is not considered your own work.

You are encouraged to work in groups while studying for tests, discussing class lectures, discussing algorithms for homework solutions, and helping each other identify errors in your homework solutions. If you are unsure if collaboration is appropriate, contact the instructor. Also, note exactly what you did. If your actions are determined to be inappropriate, the response will be much more favorable if you are honest and complete in your disclosure.

Where collaboration is permitted, each student must still create and type in his/her own solution. Any kind of copying and pasting is *not* okay. If you need help understanding concepts, get it from the instructor or fellow classmates, but never copy another’s code or written work, either electronically or visually. The line between collaborating and cheating is generally one of language: talking about solutions in English or other natural languages is usually okay, while discussions that take place in programming languages are usually not okay. It is a good idea to wait at least 30 minutes after any discussion to start your independent write-up. This will help you commit what you have learned to long-term memory as well as help to avoid crossing the line to cheating.

**College Policies**

Additional college policies, calendars, and statements are available online at [http://new.dixie.edu/reg/syllabus/](http://new.dixie.edu/reg/syllabus/).