

Computer Science and Engineering 150
Programming Languages for Artificial Intelligence
SECTION ID: 588775

Instructor: **Gary Cottrell** Phone: 858-534-6640
Office: EBU3B (CSE Building) 4130
email: gary@ucsd.edu
Office Hours: Mondays 1:15-2:15PM, Friday 10-11AM, or by appointment.

TA: Matt Tong

Tues 5-6pm @ EBU3b B225 - Office hour
Wed 1-2pm @ WLH 2206 - Discussion
Thurs 2-3pm @ EBU3b B250/B250A - Lab/Office hour

Course Description

CSE 150 is an introductory course in artificial intelligence. Familiarity with calculus, statistics, computer programming, including procedural abstraction, recursion, list and tree data structures, and elementary tree traversal algorithms is presupposed (see below).

Prerequisites

You must have all the prerequisites listed below. This is the BARE MINIMUM. If that is all you have, you will have to work hard to keep up. You cannot make up for any missing prerequisite with extra work or a friend who can help you or even the grace of God. In determining the bare minimum, we already figured in all those friends and extra help. **IF YOU DO NOT HAVE ALL THE PREREQUISITES DO NOT TAKE THIS CLASS.**

- (1) CSE 100. (This supposes that you have taken the prerequisites for 100!). *CSE 100 may be taken concurrently.* If you took a course someplace else it matters little what programming language you used but you need to have covered at least the following topics: procedures, local variables, arrays, recursion and pointers. If you are not very comfortable with recursion in some language, then you do not have the prerequisites and you should not take this class.
- (2) Enough knowledge of UNIX to move files around and to use the vi editor. (You will receive absolutely no UNIX instruction in this class, because we will assume that you are already comfortable using UNIX.)
- (3) Knowledge of very basic data structures: If you understand both depth first and breadth first searches of binary trees and can implement them in some programming language, then you probably know enough data structures to get by. If you do not feel comfortable with the notions of depth first and breadth first searches, then you should not take this class.
- (4) Knowledge of Calculus, in particular, partial derivatives, and statistics.

This course is independent of CSE 151.

Computer Accounts

See the TA as soon as possible to obtain your computer account if you do not have an OCE account.

Required Text

Artificial Intelligence: A Modern Approach second edition by Russell & Norvig.

We will place this text on reserve at the S&E library.

Required work

You will be given a series of approximately 3-5 programming assignments. Usually, you will have 7-14 days to finish the assignment. Programming assignments will *not* be accepted late. There will also be one in-class midterm and a Final. The various components are weighted as follows:

Programming Assignments: 40% Homeworks: 15%

Midterm: 20% Final: 25%

Class schedule

The following schedule is mildly optimistic:

04/03/07: Read chapters 1-2
04/05/07: Read chapter 3, Homework 1 handed out
weekend: PA 1 handed out
04/10/07: HW 2 Handed out.
04/12/07: HW 1 due.
04/17/07: HW 2 due. HW 3 Handed out.
04/19/07: PA 1 Due. PA 2 Handed out.
04/24/07: HW 3 Due.
04/26/07: Study guide for midterm handed out.
05/01/07: PA 2 Due. PA 3 Handed out
05/02/07: Review session (evening)
05/03/07: MIDTERM
05/08/07: HW 4 Handed out.
05/15/07: PA 3 Due. PA 4 Handed out.
05/17/07: HW 4 Due. HW 5 Handed out.
05/24/07: HW 5 Due. HW 6 Handed out.
05/29/07: PA 4 Due. PA 5 Handed out.
05/31/07: HW 6 Due.
06/05/07: Study guide for final handed out.
06/09/07: PA 5 Due.
06/12/07: FINAL (TUESDAY) 11:30a - 2:29p

Programming assignments are generally due at midnight on the specified date. (That is, 11:59PM plus 1 minute -- people ask!). The midterm and final are closed book. *You must bring a picture*

ID to the midterms.

Programming Assignments

NOTE: Many messages regarding programming assignments will appear on the discussion board for the assignment. YOU ARE EXPECTED TO KNOW THAT INFORMATION! Check the discussion board frequently!

When you have a question about the programming assignment, use the discussion board to clarify points and ask brief questions. *Do not post your code and ask what is wrong with it.* If you do, you will receive a zero for that assignment! Very small fragments of code may be appropriate to post. For big problems or particularly stubborn bugs see the TA or instructor in person. If it is about your program, bring a listing with you.

Naming of Functions

Part of the grading will be done automatically by grading programs. Be sure to spell function names exactly as the exercise states, otherwise the grading program will not find your function and will mark it incorrect. This applies to the choice of uppercase versus lowercase letters as well as the words and their spellings.

All programs must be easy to read and well documented. If a program is not easy to read or if it is uncommented, the grader will take off half of the style points for each one.

A large part of your grade will be determined by the following: We expect you to have function headers on your functions, and a page at the top of the file explaining your approach - data structures, your algorithm (if not specified in the assignment), explanation of your heuristics (if heuristics are asked for in the assignment) etc. The page at the top of your file is the "documentation". The style you use for your programming, and comments, are "style".

Here is an example. Your TA may provide others:

```
;;; Function: make-open-init
;;; Arguments: initial_state
;;; initial_state should be a legal state of the puzzle.
;;; Returns: a legal OPEN list, with one path on it,
;;; consisting of the null move from the initial state.
;;; Side effects: none
...
```

Turning in Programs:

Use the "turnin" facility to turn in your programs. To use it, first "prep cs150" if you have an OCE account. (Do nothing otherwise!). Then, simply type "turnin <filename>".

You **MUST** execute this command **BEFORE MIDNIGHT ON THE DUE DATE** (that is, 11:59PM plus one minute). Programming assignments will *not* be accepted late.

NOTE: HALF OF YOUR GRADE FOR PROGRAMS IS DOCUMENTATION AND "STYLE". DOCUMENTATION MEANS: A roughly page-long comment at the top of the file explaining what the program does, how it does it, and examples of its use. STYLE MEANS: A well-commented program written in an appropriate style for the language. See the example above. IT IS USUALLY BETTER TO TURN IN A NON-WORKING PROGRAM ON TIME THAT IS WELL-DOCUMENTED THAN A WORKING PROGRAM LATE, WHICH WILL RECEIVE A ZERO!

Rules for collaboration on Programming Assignments:

You may not work in groups unless the assignment explicitly says that you may. You must write your own programs. **The attached "Integrity of Scholarship Agreement" (ISA) is the final arbiter in cases of suspected cheating.** A good rule of thumb is the "Gilligan's Island Rule": You may discuss assignments with classmates *but you may not take away any written notes* and you must watch an hour's worth of Gilligan's Island (an old, pretty mindless TV show. A modern equivalent: "Who wants to be a millionaire") or equivalent before writing anything down. A second rule of thumb is, if your program logic ends up identical to your friend's, you have not followed the rule. If a question regarding cheating arises, we reserve the right to give you an oral exam on the homework to see if you understand your answer. If you do not understand the assignment, you will receive a zero for that assignment, even if the program you have is perfect in every way. If you are caught blatantly copying we will give you an F for the course.

Cheating Policies

Unfortunately, cheating has become a serious problem in CSE. A number of students have complained that they are being hurt by others cheating in classes. This is correct. Cheating lowers the value of honestly earned grades. Unfortunately, we have no alternative but to enforce strict rules against cheating.

But there is no reason to cheat! *If you are having trouble with an assignment, see me or the TA during office hours or make an appointment!!!* UCSD was recently ranked the number 1 Public Science University in the US, based on things like grants awarded, papers published, etc. I expect you to learn to become independent thinkers and problem solvers while you are here.

Exams: No talking during exams. No exceptions! We can easily tell if you are talking, but we cannot listen to and evaluate everything that anybody in the room might say. Similarly, books or notes may not be with you at your seat during an exam. If you bring books into the exam room, then you must leave them in the front of the room and not pick them up until after the exam. For the final, calculators will NOT be allowed. You must remember how to multiply decimal numbers!!!

Programming Assignments: You must follow the Gilligan's Island rule above (but this, again, is just a rule of thumb - the fine print is in UCSD's Integrity statement).

Anybody caught clearly cheating on tests or programming assignments will receive an F in the class and will be reported to their Provost's office.

CSE 150: Programming Languages for Artificial Intelligence Integrity of Scholarship Agreement

STUDENTS' RESPONSIBILITIES

Students are expected to complete the course in compliance with the instructor's standards. No student shall engage in any activity that involves attempting to receive a grade by means other than honest effort, for example:

1. No student shall knowingly procure, provide, or accept any materials that contain questions or answers to any examination or assignment to be given at a subsequent time.
2. No student shall complete, in part or in total, any examination or assignment for another person.
3. No student shall knowingly allow any examination or assignment to be completed, in part or in total, for himself or herself by another person.
4. No student shall plagiarize or copy the work of another person and submit it as his or her own work.
5. No student shall employ aids excluded by the instructor in undertaking course work.
6. No student shall alter graded class assignments or examinations and then resubmit them for regrading.
7. No student shall submit substantially the same material in more than one course without prior authorization. A student acting in the capacity of an instructional assistant (IA), including but not limited to teaching assistants, readers, and tutors, has a special responsibility to safeguard the integrity of scholarship. In these roles the student functions as an apprentice instructor, under the tutelage of the responsible instructor. An IA shall equitably grade student work in the manner agreed upon with the course instructor. An IA shall not make any unauthorized material related to tests, exams, homeworks, etc. available to any student.

The following are additional examples not listed in the General Catalog specific to programming classes:

8. No student shall provide their assignments, in part or in total, to any other student in current or future classes of this course. No student shall procure or accept assignments from any other student from current or prior classes of this course.
9. All programming code and documentation submitted for evaluation or existing inside the student's computer accounts must be the student's original work or material specifically authorized by the instructor. The course accounts are authorized for course work only.
10. Collaborating with other students to develop, complete or correct course work is limited to

activities explicitly authorized by the instructor. Use of other student's course work, in part or in total, to develop, complete or correct course work, including documentation, is unauthorized. Use of "Instant Messenger" or similar systems to share files in part or in total is unauthorized.

11. For all group assignments, each member of the group is responsible for the academic integrity of the entire submitted work.

Each student is responsible for knowing and abiding by UCSD's policies on Academic Dishonesty and on Student Conduct. Any student violating UCSD's Academic Dishonesty or UCSD's Student Conduct policies will earn an 'F' in the course and will be reported to their college Dean for administrative processing. Committing acts that violate Student Conduct policies that result in course disruption are cause for suspension or dismissal from UCSD.

Authorized course assistance is available in person and electronically from the course instructor, instructional assistants, and OASIS.