cs249 Course information

cs249 – Computational Science **Allen B. Downey**
Fall 2001 **Computer Science Department**

- Professor: Allen B. Downey, adowney@wellesley.edu, Science Center E106, x3318.
- Web page: Almost all material related to this class is available from the class web page. The URL is

  `http://rocky.wellesley.edu/cs249`

- Class-related email: I don't use a Macintosh or a Windows machine, so please don't send me attached documents in formats other than plain text.
- Class mailing list: I don't use FirstClass. At some point in the not-too-distant future we will have a mailing list for the class.

Coursework

Work in this class will include readings from materials I provide, weekly programming assignments, two one-hour exams, a final exam, and in-class activities including written quizzes.

The total course load is intended to be 11 hours per week (including class time); the load should be spread evenly across the semester.

- Grading: Final grades are determined by the weighted average of exam scores (20% for the two one-hour exams, 20% for the final), quizzes (10%), and homeworks (30%). The January Option is not appropriate for the format of this class, so it is not available.
- Exams: The two midterm exams are scheduled for Monday 1 October and Thursday 1 November. If you cannot attend one of the exams, please let me know as soon as possible so that we can make arrangements.

  If you miss a midterm for an extremely legitimate reason, your final grade will be based on the other exams. If you miss a midterm without a legitimate reason, you will receive a zero. If you miss two midterms or the final for any reason, you cannot pass the class.

- Assignments: We will have weekly assignments that are meant to give you an opportunity to apply and practice the material we cover in lecture. You will get the most benefit from these assignments if you work on them alone, although you are of course free to discuss the problems with other students and me, subject to the rules of collaboration below.

  Late assignments will be penalized by one point out of 10 for each 24 hour period after the due date. No assignment will be accepted after I have distributed the solutions. Each student is awarded 2 free late days that may be used at any point during the semester, but both may not be used on the same assignment.

Collaboration

In any class like this, it is difficult to draw a sharp line between accceptable and unacceptable forms of collaboration. Here are some guidelines:

1. In general, it is acceptable to talk about programs using natural languages, but not acceptable to use any formal language, and especially not MATLAB. In other words, you should not be looking at other people's code or showing them yours.

2. It is never acceptable to present someone else's work as if it were your own. Unless stated otherwise, I will assume that all work you hand in is yours and yours alone. If you work with another student, you must acknowledge that student's contribution in writing on your assignment. If you get help from me or a TA that constitutes a significant part of the assignment, you should acknowledge that, too. If you are not sure, err on the side of caution.

3. Copying other people's code is a serious violation and may be regarded as a disciplinary matter. It is sometimes tempting to make superficial changes to copied code to disguise it, but I should warn you that (1) similarity between programs is often more obvious than you think, and (2) an attempt to disguise cheating is evidence of guilt, and is a more serious offense since it compounds plagiarism with further deceit.

Topics

This class covers three kinds of topics

Programming: How to use Matlab and write Matlab programs.

Numerical methods: Techniques for getting approximate answers to problems without exact analytic solutions.

Modeling: Abstract models of real-world systems.

These are parallel threads. We will work through something like 10 modules, about one module per week, with some slack built in to deal with delays, and the possibility of adding 1-2 modules if time permits.

Each module contains a bit of new programming material, a numerical method, and a selection of problems we can solve with the numerical method.

Each module includes an assignment that involves modifying and writing Matlab programs, using the program to explore abstract models, and writing a report presenting your results.

At this point I have planned the first five modules in detail:

Module 1: MATLAB expressions and scripts, basic computation, debugging.

Module 2: MATLAB functions, errors, Fibonacci sequences.

Module 3: iteration, root-finding, Newton's method, compound interest, numerical minimization.

Module 4: conditional execution, recursion, vector arithmetic, fractals.

Module 5: sequences and vectors, Euler's method, the coffee cooling problem.

Other modules, not yet planned in detail, include:

Module 6: chaos, dynamic systems, predator-prey models.

Module 7: numerical integration, Monte Carlo simulation.

Module 8: arrays and permutations, combinatorics, statistical inference.

Module 9: recursion on arrays, discrete Fourier transform, fast Fourier transform, spectral analysis of speech.

Module 10: matrices, cellular automata, Game of Life.

Module 11: complexity, sand piles, earthquakes.

Depending on the interests of the class, we can choose among these modules and possibly add new ones.