

Andrew H. Fagg

Teaching Statement

With my background in computer science, I can teach most undergraduate computer science classes. For the more advanced classes (both graduate and undergraduate), my preference is to teach classes in the areas of robotics (including mobile systems, manipulators, and computer vision), computational neuroscience, embedded systems, machine learning, artificial intelligence and wearable/ubiquitous computing.

Courses Taught

In my time at UMass I have taught several core and seminar-style classes:

- Fall 2003: Graduate/Undergraduate Embedded Systems (CS 503/591c), 15 students.
This course provides an introduction to integrated hardware/software solutions in computational systems with sensing and actuation. Topics include digital and analog electronics, PIC microcontroller architecture and programming, scheduling, interrupt handling, timing, sensor interfaces and processing, actuator control, and inter-processor communication. In addition to the lecture, the class requires a significant amount of laboratory work. In this semester's class, we focused on the design and construction of components for an interactive room. I developed a majority of the course material.
- Spring 2003: Undergraduate Operating Systems (CS 377), 70 students.
The undergraduate operating systems course is required for computer science undergraduates at UMass. This class covers a wide range of topics, including: process control and scheduling, multi-threading, process synchronization, memory and disk management, inter-process communication, and I/O handling. I extended the existing course material and designed a new set of laboratory assignments.
- Spring 2003: Co-instructor (1 of 10) of the Graduate Computational Psychology Seminar (Psych 891E), 10 Students.
This course covers a range of topics in the area of computational modeling methods as applied to the psychology domain. My component of the course focused on the experimental and computational issues surrounding the encoding of movement in motor cortex and on the question of formulating and testing computational hypotheses in general.
- Fall 2002: Undergraduate Operating Systems (CS 377), 60 students.

- Spring 2002: Wearable Computing Seminar (CS 691w), 8 students.

The wearable computing seminar examines many topics in this new area of research, including multi-modal user interfaces, sensing, representation of user context, machine learning, and application design and testing. The students are required to perform a significant semester project. I developed the material for this course.

- Fall 2001: Computational Neuroscience Seminar (CS and Neuroscience and Behavior Programs: 691c), 12 students.

This seminar introduces students to the language and computational issues of the neuroscience and motor control domains. Topics include the formulation and testing of computational hypotheses, basic anatomy and physiology, experimental techniques, movement and sensory encoding, neural plasticity, and motor learning. I developed the material for this course.

- Spring 2001: Wearable Computing seminar (CS 691w), 12 students.
- Fall 1999: Computational Neuroscience seminar (CS and Neuroscience and Behavior Programs: 691c) co-taught with Prof. Andrew G. Barto; 12 students.

Future Courses

I am willing to teach and to develop undergraduate and graduate courses based on the needs of the department. I am particularly interested in transforming my computational neuroscience and wearable computing seminars into full-fledged courses. In addition, I am looking forward to developing new classes in robotics and artificial intelligence/machine learning.

Teaching Philosophy

One of the most effective ways I have found to teach students is to make it possible for them to take an active role in their own education. In this context, I see my role as structuring the class experience so that the students have the tools and the interest in the subject matter to do this in a productive manner. This process takes a variety of forms:

First, I explicitly engage the students in discussion during class. One of the devices that I have found effective is to open many of my classes with an opportunity for students to discuss recent news events that relate to the class topic. This gives the students an opportunity to direct the class conversation and to appreciate how the material about which they are learning relates to the external world. I have found that this works well even in classes as large as 70 students.

Second, I structure class assignments such that the students must interact with one-another, both inside and outside of class. Something that has been lost in the *computer in*

every dorm room culture is that students in the same class often do not have contact with other students outside of the classroom, thus losing out on the opportunity to learn from each another. I find that giving collaborative assignments helps this contact, but that the effects are particularly improved when the assignments are designed such that students must work side-by-side outside of the classroom (even when they are not collaborating directly).

For example, my Embedded Systems class (designed for graduate and advanced undergraduate students) is heavily oriented toward laboratory work. For the most recent semester, the class project was to construct an interactive room that could sense the presence of individuals in the group lab space, infer something about their activities, and customize information delivery and other services. I found that the hands-on work was very engaging to the students. In general, they were responsible for selecting a set of their own mini-projects, which were graded based on the difficulty of the problem and the quality of the work. I provided enough structure to ensure that the students satisfied a minimum set of requirements and that they made regular progress through the course of the semester. I also structured the skeleton of the projects such that the students started the class by building components in pairs, but later projects could involve an arbitrary number of students. This led to class and lab discussions about larger group goals and about standards in the way that individual software and hardware modules are constructed. Having students work together also gave them the opportunity to learn from each other. This proved to be the case even when the students were not directly collaborating on a common project.

Training Students in Research

In addition to classwork, I have supervised the research activities of 7 graduate and 13 undergraduate students in a range of research projects in the areas of robotics, computational neuroscience, and wearable computing. The undergraduate research has been performed either in the context of a Research Experiences for Undergraduates (REU) funded program (NSF) or through supervision of honors theses.

At all levels, my first goal is to teach the students some of the fundamentals of working in science: how to transform a problem into a scientific question, how to design an experiment from a question, how to analyze the results, and how to express oneself in spoken and written form. I also focus on making sure that the students develop an appropriate tool set with which to tackle the problems in their research area. With the younger students, I am typically careful about setting the direction of the research and writing processes. But as the students mature, I think it is most valuable to them (and to me) for them to take a more active role in determining the direction of their own research. In this context, research and writing becomes more of a collaborative process. I also expect senior students to take on leadership and mentoring roles. This gives the students a better sense of investment within the laboratory and prepares them to direct research in their own labs.

I have served and am now serving on a variety of Masters and PhD committees. Although most of these students are from the Computer Science Department, I have served as an outside member of two committees in the UMass Psychology Department. In addition, I

have served as a PhD committee member for students from the Universitat Jaume I in Spain and the University of Queensland in Australia.

Teaching Reviews

I was nominated for the University-level Outstanding Teacher Award for the 2002-2003 academic year.

Below, I have included the reviews from my recent Operating Systems class (Spring 2003).

Numerical Scores

	Mean (highest: 5)	STD
The instructor was well prepared for class.	4.76	0.57
The instructor explained course materials clearly.	4.52	0.78
The instructor cleared up points of confusion for you.	4.46	0.86
The instructor used class time well.	4.33	0.82
The instructor inspired interest in the subject matter of this course.	4.28	0.89
The instructor showed a personal interest in helping you learn.	4.48	0.75
The instructor provided useful feedback on your performance.	4.07	0.83
The methods of evaluating your work were fair.	4.04	1.13
The instructor stimulated useful class participation.	4.13	1.11
Overall, how much do you feel you learned in this course?	4.3	0.81
What is your overall rating of this instructor's teaching?	4.5	0.78
What is your overall rating of this course?	4.35	0.77

Individual Comments

Students were asked four different questions:

1. What do you like most about this course and/or the instructor's teaching of it?
2. What about this course and/or the instructor's teaching of it needs change or improvement?
3. What suggestions can you offer that would have made this course a better learning experience for you?
4. Any additional comments?

Below is the entire set of responses to these questions; they are unedited except where noted.

[LIKE MOST] Prof. Fagg did a good job of making the material understandable. Also, the mixture of homeworks and labs forced me to learn the material.

[LIKE MOST] Really took an interest to teach us.

[SUGGESTIONS] If the TA's had not been allowed to teach because it was ineffective and sometimes boring.

[LIKE MOST] The instructor was very helpful and showed a personal interest in helping students learn.

[LIKE MOST] The programs were the best! Challenging and rewarding, as well as subject appropriate. He also showed an amazing personal commitment to seeing his students really learn the material.

[NEEDS CHANGE] He thinks in unusual ways and so sometimes exam questions have an unusual spin on them he does not see.

[OTHER] One of my favorite classes [in] CS. He is a great teacher.

[LIKE MOST] Very good textbook and lots of material covered.

[NEEDS CHANGE] Test questions were significantly more difficult than [the] homework. Perhaps more practice would help.

[LIKE MOST] An excellent teacher; interested in making students learn.

[SUGGESTIONS] Smaller class size. I know this is not possible, but it makes a BIG difference.

[OTHER] With the exception of one Prof this instructor helps students get better grades and is willing to bend over to help. Above all the instructor is amiable and dedicated in teaching. Lab and homework assignments were good. I think more CS courses should encourage working in pairs (groups).

[LIKE MOST] The lectures were very organized and easy to understand (because of the organization and all the visual diagrams).

[SUGGESTIONS] If reading the sections in the text discussed in lectures was required (as if you had to read it before the lecture) then it would've been easier to understand a lot of things. More quizzes.

[LIKE MOST] Andrew Fagg is a great guy, always welcoming students. Great teacher, explains course material well and good homework and programming assignments. He is [a] dedicated teacher and [an] excellent instructor.

[LIKE MOST] The interest that the professor took to making sure I understood and the effort put into helping me personally.

[NEEDS CHANGE] My biggest problem is the strict grading scheme.

[SUGGESTIONS] I can't think of anything. It was all I could have asked from an OS class.

[OTHER] Excellent job Andy.

[LIKE MOST] Lecture notes are useful.

[LIKE MOST] The instructor was friendly and really tried to help people learn the theory and application of the theory.

[NEEDS CHANGE] Move exams to evening exams instead of in class.

[SUGGESTIONS] Better TA's.

[OTHER] Very enjoyable class.

[SUGGESTIONS] Interactivity - write code together in class; don't use text only slides; and don't just read what they say.

[LIKE MOST] Very interesting material, and it was always well taught, with a good amount of time spent on difficult topics.

[NEEDS CHANGE] Great, so no improvements needed.

[SUGGESTIONS] I would have liked to program more core operating system components, like the file system, but all labs were very interesting and I learned a lot.

[LIKE MOST] The networking project interested me highly.

[NEEDS CHANGE] The instructor taught very effectively, however the TAs were heavily taxed with workload and I strongly feel it affected my grades.

[SUGGESTIONS] More death rays.

[LIKE MOST] Attitude and participation.

[NEEDS CHANGE] Should use more visual aids, especially for algorithms. Also spend less time talking about slashdot news.

[SUGGESTIONS] Level of detail and the amount of material challenging and interesting.

[NEEDS CHANGE] Make labs worth less, exams worth more. More time for exam.

[SUGGESTIONS] It was a great experience.

[OTHER] As much as I enjoyed [it] I hope I don't have to take it again.

[LIKE MOST] Great lecture notes. Very good instructor.

[SUGGESTIONS] Good Professor.

[LIKE MOST] A lot of work but manageable and interesting.

[OTHER] Only good review I gave this semester.

[LIKE MOST] Network.

[SUGGESTIONS] Provide more examples and illustrations on complicated topics.

[LIKE MOST] Interesting labs, VERY helpful teacher.

[NEEDS CHANGE] Long class periods get really boring especially when it goes over time.

[LIKE MOST] We were give[n] a lot of resources to work with.

[NEEDS CHANGE] I think there was just too much information to cover. There is nothing to improve, it is actually amazing that we covered all of it.

[SUGGESTIONS] Less material to cover.

[LIKE MOST] I like OS; homework helps understand material. Appreciate the way the prof tries to make sure the student understand[s] the material.

[LIKE MOST] The third lab was good because it inspired more creativity and problem solving than the others. The other labs seemed too structured, especially lab four.

[NEEDS CHANGE] Accepting late homework would be nice. I forgot to upload an assignment and was disappointed to realize that I could not even receive partial credit for submitting it the next day.

[SUGGESTIONS] I think that using a real operating system, other than Java, for the labs would have been more helpful.

[LIKE MOST] OS stuff is just so money and cool. I learned the most from cool or sweet labs.

[NEEDS CHANGE] N'much homie

[SUGGESTIONS] My brain can only absorb so much money stuff in one semester. Not much could be done to improve it, unless my brain was implanted with special chips.

[OTHER] The course is MONEY!

[LIKE MOST] The programming assignments really helped me increase my confidence with creating programs - something I've felt was lacking in other courses in the major.

[NEEDS CHANGE] Use class time better. This includes showing up at least a few minutes early, ending class on time (and accounting for tests back, etc.), and cutting off questions when we're running out of time.

[OTHER] One of my favorite CS classes.

[LIKE MOST] The course was great. It was a lot of work but if you kept up with it, it was manageable. He was always excited about the material.

[LIKE MOST] Professor Fagg's notes were very readable/available and he explained things very very well.

[OTHER] Great class; well taught.

[LIKE MOST] He was great! He learned almost everyone's name, which hardly any other CS teacher has done. He really wanted us to learn. He was very patient.

[NEEDS CHANGE] The programming assignments were a little too hard for me, especially #2 and #4. I learned more from doing #3 because it was more manag[eable]

[OTHER] No matter what I asked the TAs, they told me it was the other TAs responsibility or that they didn't know the material (though [NAME DELETED] was often very helpful in the Edlab)

[LIKE MOST] The labs were very helpful

[OTHER] Thank you!

[LIKE MOST] The humor.

[NEEDS CHANGE] Clearer explanation of materials.

[SUGGESTIONS] More examples on board or more detailed examples on the slides.

[OTHER] No need to raise an interrupt :-)

[LIKE MOST] Obviously as a CS major I have an intimate relation w/ my OS, along w/ an enormous curiosity for its innerworkings. Fagg is very witty, very intelligent. He teach[e]s in a fun manner, and he is very comprehensible.

[NEEDS CHANGE] Sometimes we might need a mutex on the bad jokes. I hate going over the time limit in class. You always begin w/ last classes lecture.

[SUGGESTIONS] Its a hard course no doubt. Its also a fair course. I never used the resources of the TA's ... not once.

[OTHER] This guy rules. You're lucky if you have a class with him. I didn't get an A in it, but it challenged me as much or as more than some of my other favorite courses. I really enjoyed this class and will miss it.

[OTHER] Thanks to one of the programming assignments I got an internship for the Summer 2003, thus I loved this class!!!