

## Teaching Statement

**Stephanie Weirich**

I have had the privilege of teaching three programming courses, as well as assisting with the graduate programming languages course. Though it was not required, I took advantage of the following teaching opportunities, because I felt it important to develop my skills:

**CS 214, A Taste of UNIX and C,** taught Spring 1997.

My first teaching experience was a one credit, pass/fail, four-week course, with an enrollment of about 20 students. (A typical Cornell course lasts fifteen weeks and is worth three credits.) The lectures demonstrated various tools in the UNIX operating system, and the course grade was determined by a small project written in the C programming language. As well as teaching, I supervised two graders.

**CS 213, C++ Programming,** co-taught, Fall 1997.

The next fall, I worked with another graduate student, Tuğkan Batu, to teach a semester long C++ course for students who already knew how to program. This two credit, pass/fail course had an enrollment of about 50 students. Tuğkan and I collaborated in its design; we alternated preparing and giving the lectures and assignments, worked together to develop the final project, and jointly supervised three graders.

**CS 611, Advanced Programming Languages,** assisted Greg Morrisett, Fall 1998.

This core graduate course covered the semantics of programming languages. As a course assistant, I had the responsibility of grading homework, answering student questions and assisting Greg in the creation of the homework assignments and exams.

**CS 212, Java Practicum,** designed and taught, Fall 2001.

Most recently, I designed and taught a project course to accompany Cornell's second programming course for undergraduate CS majors. In past semesters, this second course has involved a large programming component, but the department felt that it would be more flexible for the students to separate the programming project into a separate course.

Because this semester was the first that Java Practicum was offered, I had a considerable amount of freedom in its design. The goal of the course was to introduce students to the principles of software engineering through several large programming assignments. I designed the semester project, a route planning tool similar to MapQuest or Yahoo! Maps, to be broken into several assignments over the semester. In that way, the students got the experience of working on a large program (about 4000 commented lines of code) that evolved in functionality over the semester. In support of this programming project, this course included weekly lectures and recitation sections. The recitation sections were led by my teaching assistant, and I also supervised three graders. The course enrollment was about 60 students.

**Teaching Philosophy** Through these experiences, I have been developing a methodology and philosophy of teaching. Learning occurs when the students become interested in the course, when they are engaged by the lectures and challenged by the assignments. The more that they interact with the learning environment, both the course staff and their peers, the more they will absorb and understand. When I teach a class, I try to instill a class community, through the use of interactive lectures, group work, course newsgroups, and web pages.

I believe that a computer science department has an obligation to the university to teach more than what has been traditionally been categorized as computer science. I want my students to be creative problem solvers and abstract thinkers with the ability to apply general principles to a wide variety of problems and to reason logically about their solutions. I also want them to be strong communicators, able to express their thoughts with confidence and to collaborate and learn from each other.

However, I am still learning how to be an effective teacher. I look forward to teaching a wide variety of courses and further developing my skills. I also look forward to the energy and excitement of teaching, and to learning from my students.