Kate Lockwood
Teaching Philosophy

I feel strongly that students need to feel like computer science is an accessible and useful discipline. To that end I develop assignments that are tied to current events or to student interest. I encourage students to come up with their own ideas for course projects and to think of ways to tie their CS work to projects in other courses. In addition, being able to explain clearly the methods used on a project and to describe its relevance are nearly as important as syntactic issues in my grading methods. This encourages students to think about the problem solving process involved in programming instead of focusing on syntactic details. These philosophies manifest themselves in several course goals that I strive to maintain:

1) As much as possible, I try to limit the amount of straight, uninterrupted lecturing that I do. In introductory courses, my personal preference is for class to be taught in a room where students have access to computers. Lecture can be interspersed with hands-on activities to keep students engaged and interested. For other classes I try to pepper lectures with short in-class assignments and activities to keep students engaged with the material.

2) I give students many opportunities to provide feedback on how the course is going and which topics they have questions on. I feel that students are more invested in a class if they feel like their feedback and participation is valued. I incorporate several different types of student feedback. I have had a small group analysis of a course conducted by an on-campus teaching group. I also periodically use quick (5 minute) ungraded quizzes or writing assignments to gauge which topics students need refresher lectures on. I try to address student questions and concerns and to provide explanations for course policies that they may disagree with.

3) I craft assignments that involve real-world situations and that appeal to many different disciplines. By creating assignments where students have "clients" in mind, I try to make homework appear more relevant to future job tasks and less like busy work. In most of my classes, I allow students to pick their own final project (with approval and guidance). This way students are allowed to tailor their work to something that actually interests them. They can see how computer science can work for them.

4) I incorporate "writing across the curriculum" activities whenever possible. Instead of just asking students to write a program, I will ask them to also explain their algorithm or design choices by writing a short paragraph. I also have them present their projects in class. This gives students the opportunity to try their hand at technical writing. It also makes them focus on the ideas and concepts they are learning and to develop communication skills.