Teaching Survival Tactics
CRA-W Workshop, SIGCSE 2015

1. Start your planning with the goals of the course. What do you want the students to be able to do when the course is over? Then craft the examples, assignments, and tests toward those goals. Don't forget non-academic goals like developing teamwork, becoming an independent thinker, or discovering computing careers.
2. Don't take anything personally, especially student evaluations. Just like your grading is not personal, their evaluations aren't either.
3. Find a way to assess your courses that is meaningful to you. It may be your own crafted evaluation, a comparison with other CS courses at other institutions, or an alumni survey.
4. Come to class 15 minutes early and just chat or discuss topical issues in the news. You'll get to know the students, and they'll get to know you.
5. Set the tone and expectations for the class on the first day. For example, trying to introduce active learning midway through the semester or only for one day doesn't seem to work. The students should know on the first day what kind of class you will run, what your goals for them are, and what participation expectations you have.
6. Ask for feedback at midterm. You can give a survey, hold a class discussion, or hold individual meetings with students. Talk about what things are working and what things may not be. This gives you a chance to change things before the end of the term. Also, the students feel heard, and they are less likely to still have problems when the final evaluations come around.
7. Prepare students for filling out student evaluations with an overview of the entire course. Students tend to remember just the most recent things that happened. Remind them of how far they have come since day one and be candid about how you plan to fix things you thought did not go well.
8. Document everything with students. If someone challenges a grade, be ready.
9. Find a Center for Teaching and Learning at your institution or a neighboring one and participate. They are a great resource for education resource and can help you apply it to your courses and to do education research of your own.
10. Do something that motivates you for each new semester. Some ideas are keeping good student evaluations to read, talking to an alumnus, or watching an inspirational teaching movie.
11. Learn something new. This will remind you what it is like to struggle with learning like your students do. It will help you remember the frustration of seeing someone “make it look easy.”
12. Begin and end class on time. It shows respect and value for the students' time.
13. Be comfortable with silence. Students need time to think when you pose a question to them. If you are uncomfortable in the silence, you will prematurely jump in with the answer.
14. Find a great teacher and observe them. It doesn't have to be someone in your discipline.
15. Don't sacrifice the experience of your class just to cover more material.
16. Check to see if the way you spend your class time & effort accurately reflects what you think is important for students to know or experience. If it isn't important to you, cut it.
17. Keep notes about how everything goes so you repeat only the good decisions.
18. Assume the best from your students, and set your expectations high.

Paper References:

“Reshaping the Image of Computer Science in Only Fifteen Minutes (Of Class) A Week.” Sara Sprenkle and Shannon Duvall. SIGCSE 2012.

“Creating a Games Class: A Walkthrough.” Shannon Duvall. Proceedings of the Foundations of Digital Games, 2009.

"Computer Science Fairy Tales." Shannon Duvall. CCSC 2008.

“A Picture is Worth a Thousand Words: Using Digital Storytelling in the Classroom” by Michele Kleckner and Shannon Duvall, IACIS 2007.

"Everything I Needed to Know About Teaching I Learned in Kindergarten: Bringing Elementary Education Techniques to Undergraduate Computer Science Classes." Shannon Pollard and Robert C. Duvall. SIGCSE 2006.

"Hands on Labs Without Computers." Shannon Pollard, Jeffrey Forbes. SIGCSE 2003.

“Making Lemonade: Exploring the Bright Side of Large Lecture Courses” by Steven A. Wolfman, SIGCSE 2002.

“Active Learning in Small to Large Courses” by Owen Astrachan, Robert C. Duvall, and Eugene Wallingford, FIE 2002.

“Toys Are Us: Presenting Mathematical Concepts in CS1/CS2” by Paolo Bucci, Timothy J. Long, Bruce W. Weide, and Joe Hollingsworth, FIE 2000.

“Concrete Teaching: Hooks and Props as Instructional Technology” by Owen Astrachan, ITICSE 1998.

“Active Learning and its Use in Computer Science” by Jeffrey J. McConnell, ITICSE 1996.

Online References:

Current events for Computer Science: http://computersciencenews.blogspot.com/

Kindergarten Computer Science: http://www.cs.duke.edu/csed/kindergarten/

Nifty Assignments: http://nifty.stanford.edu/

Computer Science Unplugged: http://csunplugged.org/

Active Learning Ideas: http://www-cs.canisius.edu/~mcconnel/active_learning.html

Kinesthetic Learning: http://www.cs.ubc.ca/~kla/

Pedagogical Patterns: http://www.pedagogicalpatterns.org/