

Active and Cooperative Learning Techniques for the Computer Science Classroom

Laurie Murphy (Moderator)
Pacific Lutheran University
Tacoma, Washington
murphylc@plu.edu

Kenneth Blaha
Pacific Lutheran University
Tacoma, Washington
blahakd@plu.edu

Tammy VanDeGrift
University of Washington
Seattle, Washington
tammy@cs.washington.edu

Seven Wolfman
University of Washington
Seattle, Washington
wolf@cs.washington.edu

Carol Zander
University of Washington, Bothell
Bothell, Washington
zander@u.washington.edu

Laurie Murphy: Introduction

There is a wealth of education research confirming the benefits of active and cooperative learning in higher education; benefits which include a deeper understanding of the material, higher academic achievement, increased student persistence and retention, positive peer relationships and higher self esteem [Felder and Brent, 1994]. Incorporating active and cooperative learning into the computer science classroom can be challenging but well worth the effort, particularly because it addresses issues that are unique to computer science education; it provides a mechanism for coping with the ever-increasing complexity and change inherent in our discipline, it can serve as a tool for dealing with the wide range of backgrounds and abilities exhibited by our students, and it can facilitate a safe and positive learning environment for students from underrepresented groups. This panel is a forum to discuss the challenges and benefits of active and cooperative learning and techniques for incorporating active and cooperative learning into the computer science classroom.

Reference:

R.M. Felder and R. Brent, *Cooperative Learning in Technical Courses: Procedures, Pitfalls, and Payoffs*. ERIC Document Reproduction Service, ED 377038 (1994).

Kenneth Blaha: From a Lecture Based Classroom to an Interactive Classroom

Most of us took undergraduate and graduate classes that were entirely lecture based. We are most familiar with this classroom environment and it is the style that I used for many years. This method of teaching is an effective way of distributing information to a large group of people. Unfortunately, the emphasis is typically on content and not on learning. Urged by a colleague and other educators I have attempted to move from a lecture base classroom environment to a student based active learning environment. This has been an interesting transition.

I will mention some of the techniques I have used and how students have reacted. I will also talk about the difficulty of making this transition and some of the pitfalls and successes.

Tammy VanDeGrift: Using Active Learning in Assessment

As computer science instructors, it is important to learn what, how much, and how well our students are learning. Classroom assessment techniques are learner-centered activities, designed by the instructor for formative assessment of his or her students [cite Angelo, Cross]. Many assessment techniques engage students in reading, writing, discussing and problem solving; these activities create active learning experiences while providing feedback to the instructor. Examples of classroom assessment techniques include minute papers, muddiest points, directed paraphrases and extended analogies. Many of these activities can be done in pairs or small groups, giving students opportunities to learn from each other.

I will describe specific classroom assessment techniques that I have developed for introductory computer science courses. I will address how an instructor would use these techniques, how students have used them in my experiences and how the students' responses informed instruction.

Reference:

Thomas A. Angelo and K. Patricia Cross. *Classroom Assessment Techniques: A Handbook for College Teachers*. San Francisco: Jossey-Bass Publishers, 1993.

Seven Wolfman: Motivating Active and Cooperative Learning

Active and cooperative learning have the potential to energize students about their own learning. Even the simplest application of these techniques may be more motivating than a well-crafted but passive lecture; however, a few rules of thumb can help to make your class's exercises even more engaging, evoking excitement about the material and passionate discussion:

- Craft problems that are intrinsically motivating. Some aspects of a problem that can motivate students are solutions that violate expectations, authentic experimentation, and connections to real problems or outside knowledge/experience.
- Encourage and capitalize on student diversity to foster discussion. Real problems in computer science rarely have a single answer; instead, CS problems define a rich design and implementation space encompassing fundamental tradeoffs and room for creative difference. Leave room for variance in your exercises. Recognize---and legitimize!---different students' values and backgrounds that lead to different choices within this space.

- Use Jigsaw Learning or other methods that create resource interdependence---distribution of the resources needed to solve a problem across group members---in order to foster discussion [Johnson and Johnson 1999].

Reference:

Johnson, D. W., & Johnson, R. (1999). *Learning together and alone: Cooperative, competitive, and individualistic learning (5th ed.)*. Boston: Allyn and Bacon.

Carol Zander: Issues with Student Group Work

Many computing instructors include group work in their courses as a way to simulate the work environment and to expose students to working with others. Most agree group work is beneficial both from the standpoint of learning how to work with others on technical problems as well as learning how to cooperate, collaborate, and deal with different personality and ability levels.

This work can take on many forms such as performing various aspects of the software life cycle by working together on large projects, or students' reviewing each other's writing, designs, code, etc. There are many ways to form and organize groups, many ways to evaluate groups, and many, many problems with student groups. I will examine the pros and cons of these different aspects of group work.