

# Don't Panic: Friendly Advice on Teaching Your First Course as a Graduate Assistant

Shannon Steinfadt  
Kent State University  
Computer Science Department  
Kent, OH 44240  
330-672-2074  
ssteinfa@cs.kent.edu

Ellen L. Walker  
Hiram College  
Computer Science Department  
Hiram, OH 44234  
330-569-5250  
walkerel@hiram.edu

## ABSTRACT

As a new graduate or teaching assistant it may be difficult adjusting to the dual role of student and instructor, since most of a graduate students' academic career is as a student. Moreover, the majority of undergraduates in the sciences do not study teaching methods as their main focus is on their major. The dual role of student / instructor brings with it some new responsibilities—educating others is now part of the job. New issues to deal with include setting up a course, getting through the first day, presenting the material effectively, and encouraging all students, including women. Entire manuals have been written devoted to teaching methods, classroom management, and general advice for more effective teaching. Therefore, this paper is not comprehensive; instead we share ideas of where to start, and present some references for further information.

## Categories and Subject Descriptors

K.3.0 [Computers and Education]: General

## General Terms

Management

## Keywords

Education, pedagogy, assignments, classroom management

## 1. INTRODUCTION

Making the transition from student to instructor as a Teaching Assistant or Graduate Assistant (GA) can be a bit of a shock. Your goal after graduate school may be to become a researcher, not an educator. However, regardless of intent it is likely that you will teach for support while in graduate school. It is also true that no one is ever really ready for teaching. Do not be scared; educating others has its challenges but also has its own rewards including a great amount of joy and satisfaction.

A brief background of the authors: Shannon Steinfadt is a Ph.D. candidate who has been teaching labs and lecture courses since her first year in graduate school back in 2000. Dr. Ellen Walker has 17 years of teaching experience in graduate and undergraduate computer science courses.

## 2. Before You Teach Checklist

Whether it is your first day of a new class or your first class ever, some work and reconnaissance is necessary before you

step in front of your eager audience. You need to get an idea of where you will teach and what you need to bring with you.

The following questions come from Rishel's helpful "Before You Teach" checklist [1] with a few additions based on our experience teaching computer science. Our additions are in italics.

- Do you have keys for your office or your classroom?
- Are the classrooms going to be open?
- Will you find chalk and erasers in the classroom? *Is there a white board and do you need to bring markers?*
- Where is your office? What is your office number?
- What is your telephone number? Your email address?
- Can you get a desk copy of the textbook for your course?
- What is the policy on making copies of materials for your class?
- *What facilities are available after hours?*
- Can you get old syllabi for your class? How about last year's exams? *Are there common exams?*
- Is the exam schedule made up in advance? Is the syllabus and exam schedule set in advance? Can you make changes to it?
- Can you get an overhead projector and transparencies? Do you even want these things? *Will you be using PowerPoint presentations?*
- *Will you be using a computer? Is there a computer and a projector available?*
- How many students might you expect to see in your class?
- *Are you responsible for putting up a course web page? If so, where?*
- *Do you have to use a course management system? Are there any available?*

Other graduate assistants, the course coordinators, and departmental assistants are all valuable resources for you. Those individuals can be very helpful but try to be reasonable with your questions as their time is valuable.

## 3. The Syllabus

The syllabus is a plan for the course that states how the course will be run. It should include your name and contact information, the name, location and time of the course, and information about the course content and course policies. Do

not start developing a syllabus from scratch, as your university probably has specific formats to follow—ask someone for a syllabus to modify. By changing the personal contact and schedule information to reflect your section of the course, you have an instant syllabus.

### 3.1 Course Schedule

The department or the course coordinator often sets the course schedule. If someone else sets the schedule you need to adhere to it, not spending too much time on earlier topics and finding at the end of the course there were things you should have covered and did not. For that matter, that outside someone may specify the number of quizzes, etc., so find out the rules and follow them!

To set the course schedule you should know what material absolutely has to be taught in the given amount of time. In other words, what chapters are students expected to know for subsequent courses? If the only information you are given is what has to be covered, a good rule of thumb is to apportion that material evenly throughout most of the semester. Leave the last two weeks for topics of student interest. However, there are good sources of additional information including your course coordinator, old syllabi, syllabi used at other schools that use the same book, and even the foreword of most textbooks. In any case, the book itself is a great resource. Typically the textbook author has taught the course many times, so it is a good policy your first time through a course to use the order of presentation in the book.

To aid the student, the syllabus should minimally include a weekly schedule of the chapter(s) to be covered, the due dates of the major assignments, and the dates of all exams. If you plan to give unannounced pop quizzes or in-class assignments, a statement that indicates this should be included on the syllabus. It may seem like a lot of work to set up all these details in advance, but it makes life easier for both you and your students.

### 3.2 Course Policies

Another important aspect to the syllabus is to provide course policies upfront so that when an issue arises you can easily handle it. Some policies to consider are late assignments, academic dishonesty and attendance (including exam attendance).

There are many options for dealing with late assignments. Some courses may not accept late assignments at all and others may take off a certain percentage of the grade per late day. While students clearly prefer the option to submit work late, remember that late submissions can prevent you from going over the answers or handing back work, and that time-management skills that students learn by meeting deadlines in school are valuable after they graduate.

Academic dishonesty can have serious consequences, from failing the assignment through failing the course or even expulsion from the university. Therefore, it is very important to define what constitutes academic dishonesty for your class. For example, can students discuss algorithms? Can they debug each other's code? Can they share code snippets with each other? It is also important to determine and clearly state the penalty for dishonesty in your class. Many colleges and universities have predefined language about academic dishonesty for you to include in your syllabus.

Attendance is another area where a policy is needed. If attendance is required, consider in advance what an excused absence is and what proof will need to be provided by the student. Even if attendance is not required in the class, you will need to think about what the policy is if a student misses a scheduled exam. Many instructors allow students to negotiate makeup exams for absences that are known in advance, although you are not required to do so.

These policies are best borrowed from instructors with more experience until you gain your own preferences. Look back at syllabi from previous courses you have taken or from other instructors who have taught this course.

## 4. Making It Through the First Day

Now that you have a teaching assignment, a syllabus and you have gone through the checklist it is time to show up for your first class. Do not let panic ensue!

### 4.1 Attitude

Your attitude sets the tone for the rest of the course, so there is no pressure, right!? The “Instructor Attitude” section in [2], quoted below, summarizes some of the most important points succinctly.

- Be confident; remember, even if you're uncertain about the material, you know more than your students do.
- Keep a positive attitude; it's contagious.
- Be honest with students about the course and your experience.
- Tell students you will make mistakes, and don't be afraid of making them as class proceeds. (After all, they might learn something from them ...)
- Don't apologize *too* much.

As new instructors, we were surprised how much our confidence and attitude influenced a class. The more positive the instructor, the more positive the students and the more excited they are.

### 4.2 First Day Logistics

The first day of class sets the tone for the class as whole. To help you get over the inevitable first day jitters you should have a specific plan.

- 1) Dress for success. Wearing somewhat more formal clothes will increase both your confidence and the respect you get from your students. This is especially important if you are a woman or if you are young-looking.
- 2) Introduce yourself. Give your name and how you would like to be addressed by the students, such as “Ms. Steinfadt”, “Ms. S”, or Shannon. This was skipped one semester by the first author, and students were confused. They would avoid using a personal address or guess an incorrect one; everything from Dr. Steinfadt, Professor Steinfadt, Mrs. Steinfadt, to Shannon. Avoid the discomfort and confusion.
- 3) Call the roll and get the correct pronunciation of the students' names. Learning the student's names shows your interest in the students, which leads them to feel more responsibility for the course. This sense of responsibility can lead to better attendance and help prevent other problems such as cheating [1]. While

learning the names of everyone in a large class can be difficult, two helpful methods are to use a seating chart and handing back assignments personally instead of having students pick them up.

- 4) Review the syllabus. Hand out a copy and read through it with the students. Discuss the “nuts and bolts” of homework, exams and grading. Now is the perfect time to cover cheating and late assignment submission policies.
- 5) Start covering material. This might seem obvious or obnoxious depending on whether you are an instructor or a student. The tone of the course is set from the first day. By using the complete class time, you send the message that this is a serious course with a lot of important (and interesting) material to cover. However, beware of trying to cover too much on the first, or any, day. Nerves and unfamiliar material cause instructors to speed up [1].

In between these activities remember to breathe. Remember that you are allowed to make mistakes and you will. It's ok.

## 5. Using Classroom Time

Congratulations, you have made it through the first day! There is a lot of material to cover in the days and weeks that stretch out in front of you. In planning classroom meetings, the structure of each meeting should be clearly defined. A common method for introducing new material is the lecture. Other sessions might include student presentations, group work, labs, quizzes and tests. There is an enormous amount of literature that describes effective ways to use classroom time. In this paper, we can only begin to scratch the surface. The different structures and some ideas of how to actually cover the different types of information are outlined below. For a rich set of teaching and assignment ideas look at [3], [4], and the Online Resources listed in [2] including SIGCSE's education links.

### 5.1 Lectures

Lectures are often the primary structure used throughout the course. When lecturing there are several ways to present the materials including “chalk and talk,” overhead slides, from a web page, or PowerPoint slides. The topics of the course and prepared materials already available, along with personal preference, often determine your choice. Having another textbook in the area is a great way to have backup information and coverage.

Textbook authors often make PowerPoint slides available, perhaps on a password-protected web site. As an instructor teaching a course using their textbook, you can get access to that site. If you do use someone else's slides, (1) make sure you give them credit (i.e., do not pass the slides off as your own), and (2) make sure you understand the slides you are using (i.e., do not download them an hour before the class and read them then for the first time).

If you are using an electronic medium such as notes on a web page or PowerPoint slides, it is advisable to make them available for students, either on paper or online. With notes, students can focus on what is being said instead of frantically trying to write everything down. The depth of the information in your notes should not be textbook coverage. Notes are a guide to lead your presentation; they are not your entire lecture. Some instructors will leave out crucial words or

arrows in figures so that students will have to pay attention and fill them in.

Even when lecturing, it is good for the class to be interactive [5][6]. Asking questions and then waiting for a student response is important. As a new instructor it is difficult to wait long enough to give students a chance to think about the question and find their voice. Avoid the temptation to answer your own question, as painful as it may seem.

For lectures that involve teaching a programming language, another powerful technique is to use the compiler or interpreter being taught. Introduce a concept, but *show* the students how it works. Pose a problem that makes use of the new topic. For example, if you are discussing for loops you can ask the students how to count up to a given value and output each intermediate number. Have them tell you what to program while you sit at the keyboard or stand at the board. Help them out, but they will learn more if you let them make mistakes and have them figure out what went wrong.

Reading through and running example code from the textbook is another interactive method for teaching programming. The programs themselves can be posted on the course webpage for the students' access.

### 5.2 Peer Learning

Peer learning refers to many different types of group interaction both in- and outside of the classroom setting [5]. Start small. Peer learning is not a way for the instructor to do less work. In fact, peer learning activities require careful planning and preparation [7].

#### 5.2.1 Student Presentations

The focus here is on student presentations to the class. Individuals or small groups of students are assigned to give presentations to their classmates over a particular topic. They have to introduce the concept, show examples and make an appropriate conclusion to their presentation. The instructor is there to help out but not to present the information for the student(s). This requires being able to relinquish a level of control in the classroom.

#### 5.2.2 In-Class Collaborative Learning

Group work is a powerful tool that is often overlooked or viewed as inefficient. Students commonly overestimate their comprehension of complex topics. A worksheet or a series of questions answered in a group session allows students to interact and help teach each other. In-depth comprehension is often achieved through explanation to others. These small group activities engage students to become more active. Collaborative learning is a good way for new instructors to introduce peer learning and change the pace of the course without a dramatic shift in classroom dynamics and little chance for failure [7].

Know ahead of time what size groups you want. Ideally groups are two to four students. Groups of three give enough variation of opinions and make it difficult to not participate.

The group activity should be short and focused. Give the students a worksheet or a set of questions with a central theme. Limit their time and help keep them on track. Giving immediate feedback, either by going over the questions as a class or by handing out an answer sheet helps the students lock in the correct answers while the material is still fresh in their mind.

Students should have had ample time to read and understand the topic before holding an in-class group session. Presenting a new idea during a lecture session and going right into group work can be difficult for the student, especially if they were not prepared for the change in class structure.

The instructor is the most active participant in the group sessions, visiting the different groups. She should listen and be available to answer any questions, acting as a problem solver and not a group leader. When instructors circulate in a non-evaluating context, students in small groups are more likely to ask greater and more detailed questions.

#### 5.2.2.1 Example

One chapter in the text of an introductory course was an overview of C++ and structured programming. Group work was used to help the students learn the finer details of the topic with instructor's help.

A worksheet was provided to each student that covered C++ programming syntax and concepts. A short program was given that output the maximum and minimum of 100 user input numbers. It used for loops, Boolean logic, conditional statements and input/output. Students were asked to comment each line with a description of what it did. A separate exercise had students create a function on paper to input two integers and output their sum for practice with functions. The two short written exercises covered declaration of a constant and a yes/no question about case sensitivity.

The instructor circulated among the groups of three to four students to answer questions about C++ syntax and concepts. The questions that the students asked were much deeper than those asked in lecture and showed students' active participation and concentration in the topic. At the end of the working time, answer sheets were handed out and briefly discussed in class as the worksheets were "self-graded" by the students.

### 5.3 Labs

Depending on the course you teach, you may be responsible for a lab. Even if no formal labs are scheduled, using lab techniques in class can be useful. Labs generally involve longer and deeper assignments than the group work mentioned above. Students can work individually or in small groups, depending on available facilities and instructor preference.

A lab can begin with a quick review or an introduction of the concepts that will be used during the lab session. Students are given a lab handout and work at their own pace.

A structured lab handout can consist of problems from the text or an assigned laboratory manual, or it can be original. The tasks specified in the handout should be manageable and build upon each other. As an example, if one exercise captures user input, the following exercise might manipulate the user input. Take small steps. What is obvious to an individual with a bachelor's degree in the subject area is usually anything but obvious to an introductory student. Help your students by building their understanding, one clearly defined step at a time. This will help them build their confidence and competence in the material.

The lab assignment should be adequate to fill up the allowed time and discourage idleness, but it should not be impossible to complete within the allotted session. Gauging the amount of time students need to complete a lab can be difficult. If the

students do not finish during the lab period, consider assigning the unfinished portion as homework.

Like group work, labs require that the instructor be available. This does not mean that you sit at your computer and do your own work. Interact! Walk around and monitor progress. Offer help when you see that things are not going as expected. Offer encouragement when you see good work. Be aware of balancing time between students as fairly as possible.

## 6. Test Their Knowledge

How can you tell whether or not your students comprehend what you are teaching? Personal experience has shown that tests or assignments that count towards the course grade are the one of the best indicators. A graded assignment reveals the depth or lack of students' knowledge. Graded evaluations force students to read and comprehend the material covered so far. They also help students to evaluate their own progress.

Smaller and more frequent tests and assignments can mean more work for you, the instructor. However, they also allow for more manageable chunks of information that the students can handle and better understand. Balance these two concerns, but whichever decision you make, hand back graded materials promptly, ideally within one week. Frequent evaluations that are not handed back for many weeks are of little value in providing feedback to the students.

It is helpful to have a graded assignment early in the term. A student can understand the instructor's style of grading early on to make adjustments early enough to achieve their desired grade in the course. It also sends the signal that this is a course that requires work. Countless times students have mentioned that they did not realize the amount of time and effort required even in a lower level computer science / computer engineering course. An early assignment of some depth can be a clear indicator of this. This can prepare the students for success in the course.

### 6.1 Assignments

When you create an assignment you should have an objective in mind. Think about what your students should learn by having completed the assignment.

Assignment types vary widely. Uncompleted labs or in-class worksheets can become an assignment. Programming assignments may be for individual students or use the Extreme Programming (XP) method of pair programming [8][9]. Pair programming is another form of peer learning with many advantages for both the students and instructors including better code, more satisfied and self-sufficient students, and less overwhelmed instructors in regards to helping students and with grading [10].

In addition to programming assignments, many courses have written assignments where students solve problems or answer questions without using the computer. While the textbook often provides good examples of problems, be aware that answers for these questions are widely available on the Internet. One way around the available solution sets is to personalize the assignment for the student, e.g. if doing string manipulation have the student use his or her own name or initials.

### 6.2 Exams and Quizzes

Giving a balanced, reasonable test that examines a broad spectrum of the material covered is difficult. New instructors

are strongly encouraged to look at exams from other instructors. Use them as guideposts in order to write your own exam. Be sure that the treatment of topics and their emphasis match the coverage given in your own class, not that of the other instructor.

A shortcut to writing exams is to write one or more exam questions after each lecture. While the material is still fresh in your mind, one or two test questions come quite easily. Keep a running log or file of these questions. They can be any style of question such as true/false, multiple choice, matching, fill in the blank, fill in the code, or completion. When it comes time for an exam, it is simply a matter of choosing and refining questions from your own test bank. Using another textbook is also good source for producing homework assignments and ultimately quiz and test questions.

Quizzes are a useful tool for evaluating students' progress often without the overhead of a full exam. A quiz typically consists of a few short answer or true/false questions. Aim for quizzes that take no more than five minutes.

If you are thinking about giving quizzes or pop quizzes you should list them on the syllabus and how they figure into the grade, i.e. as a homework, as part of the exam scores, etc.

## 7. Encouraging Female Students

In the field of computer science, women are highly underrepresented (15.1% of new Bachelor's degrees in Computer Science, according to the latest Taulbee Survey [11]). As an instructor, you have the power to create a classroom environment that encourages women to excel and continue in computer science. While these ideas are targeted toward women, many are helpful for underrepresented minorities and can benefit the majority students as well.

### 7.1 Choosing Relevant Topics and Assignments

Women tend to be interested in applying computers to a problem that is relevant to them, rather than simply studying computers for computers' sake [12]. Introduce algorithms in the context of applications that require them. For example, you can introduce sorting by asking students to consider how they would find a book if the library were entirely unsorted. Develop assignments that reflect real-world applications as well. Another idea that emphasizes the utility of what students are learning is to have them use their newly acquired skills to help their community. If teaching how to build a web page, students could work on pages for a local charity or non-profit organization.

Your choice of examples affects students and can balance the stereotypes that exist in society. Although many network problems in textbooks involve roadmaps or building networks, using an example of social networks counteracts the stereotype of the isolated asocial computer user and also yields interesting assignments. Multimedia (i.e. working with digital image and sound files) is a good source of examples that many women find attractive [13].

Even in spoken examples, be aware of their relevance to everyone in the class. Avoid the use of stereotypes, not only of computer use, but also of computer users. Instead of using "he" for all programmers, computer designers and users, consider using "she", "you", or avoid pronouns entirely. In examples and assignments where names are used, use a mixture

of masculine and feminine names, or use names like "Chris" and "Pat" that could be interpreted either way.

## 7.2 Encouraging Participation

Your behavior as an instructor, even unconsciously can either encourage or discourage participation by less-confident students, which often means women in the class [14]. Some students can be intimidated by the instructor's or other students' styles of interaction, especially given that males and females tend to have different communication styles [15]. As an instructor, you can use several techniques to encourage participation by all and discourage intimidation. The following ideas are adapted from a workshop handout from Princeton University [16].

Behaviors to avoid:

- Looking directly at the students (men) who you expect to answer a question
- Calling men by their last name but women by their first name
- Allowing students to interrupt the one who is giving an answer or to dominate discussion
- Offering more substantive responses to men's answers than to women's
- Making "helpful" comments that disparage women's abilities, such as "I know math is difficult for you..."

Strategies to encourage participation:

- Tell students on the first day that you expect everyone to participate and you will be calling on all of them over the course of the semester. Then, keep track over time and be sure to call on the "silent" students.
- Call students by name, and refer to their contributions by name (e.g. "Remember the point that Shannon made...")
- Do not only call on students that raise their hands. In addition to picking out students by name, you can encourage participation with statements like, "we haven't heard from the left side of the room in a while..." or even "Who else besides Shannon wants to answer?"
- Give students time (5-10 seconds) to get their hands up before calling on someone.
- Ask follow-up questions (of everyone) such as "Why do you think that?"
- When a student deserves it, offer praise.

Especially in introductory courses, be sure to encourage the stronger students to continue their study in computer science. Many women have cited attention from one of their early instructors as the reason that they chose (or remained in) computer science. Without such encouragement, women are more likely to lose confidence and leave the field entirely [14].

## 8. Flexibility and Reuse

It is important to remain flexible. If something is not working for you, do not be afraid to change it. If you are going too fast you can slow yourself down by adding more examples or additional figures. You may think you can cover the concepts of the Von Neumann architecture in two class periods because you have only a set number of slides but you may need to re-visit or expand your coverage of such complex topics. Check in with students by asking them questions and watch their

reactions. Often confused students will give up rather than ask questions.

Reuse may not seem important now but it will be in the future. Personal experience has shown it takes at least two hours of preparation for every one hour of class time, in addition to grading time. The preparation time can be well spent when you teach a course again. Do not be afraid to re-use notes, assignments, or test questions from different courses if the topics are the equivalent. At the same time, you can help yourself out by keeping clear notes and even a log of what works and what does not [2]. In this light, it is important to correct any problems or errors right away. That way you (or the students) do not have to re-discover the issues the next time you teach the same course.

## 9. Conclusion

Breathe, relax, and prepare—you can do a great job teaching. Planning and structure go a long way to help create a varied and interesting course.

As a new graduate assistant, teaching is a challenge but not impossible. Teaching has its own set of rewards. It is amazing when you excite your students in about your subject area. Your enthusiasm is contagious, as is theirs. When you teach you are also learning, not only from the material taught but also from the students. When you see a student start to understand and learn, it is a huge thrill. Hopefully some of the ideas presented will help in the next (or first) course that you teach.

## 10. ACKNOWLEDGMENTS

We are grateful to all of our students who help us while we continue to learn about teaching.

## 11. REFERENCES

- [1] Rishel T.W. *A Handbook for Mathematics Teaching Assistants: Preliminary Edition*. The Mathematical Association of America, Washington, DC, 1999. Also available at <http://www.maa.org/programs/tahandbook.html>
- [2] Huggins, J., Bergin, J., Caristi J., and Walker, E. Survivor: Getting Through That Class the First Time. *ACM SIGCSE 2003 Panel Discussion* (Reno, NV February 21, 2003). Taken from <http://www.cstc.org/data/resources/300/survivor.html> on July 24, 2006.
- [3] Pollard, S. and Duvall, R. C. 2006. Everything I needed to know about teaching I learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes. *SIGCSE Bull.* 38, 1 (Mar. 2006), 224-228.
- [4] Parlante, N., Wolfman, S. A., McCann, L. I., Roberts, E., Nevison, C., Motil, J., Cain, J., and Reges, S. 2006. Nifty assignments. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06. ACM Press, New York, NY, 562-563.
- [5] Lopez-Herrejon, R. E. and Schulman, M. 2004. Using interactive technology in a short java course: an experience report. In *Proceedings of the 9th Annual SIGCSE Conference on innovation and Technology in Computer Science Education* (Leeds, United Kingdom, June 28 - 30, 2004). ITiCSE '04. ACM Press, New York, NY, 203-207. DOI= <http://doi.acm.org/10.1145/1007996.1008051>
- [6] Rodger, S. H. 1995. An interactive lecture approach to teaching computer science. In *Proceedings of the Twenty-Sixth SIGCSE Technical Symposium on Computer Science Education* (Nashville, Tennessee, United States, March 02 - 04, 1995). C. M. White, J. E. Miller, and J. Gersting, Eds. SIGCSE '95. ACM Press, New York, NY, 278-282. DOI= <http://doi.acm.org/10.1145/199688.199820>
- [7] Wills, C. E., Deremer, D., McCauley, R. A., and Null, L., "Studying the Use of Peer Learning in the Introductory Computer Science Curriculum", *Computer Science Education*, Vol 9, No 2., August 1999, pp. 71-88.
- [8] McDowell, C., Werner, L., Bullock, H., and Fernald, J. 2002. The effects of pair-programming on performance in an introductory programming course. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education* (Cincinnati, Kentucky, February 27 - March 03, 2002). SIGCSE '02. ACM Press, New York, NY, 38-42. DOI= <http://doi.acm.org/10.1145/563340.563353>
- [9] Pair Programming Web page. <http://www.pairprogramming.com/>. Accessed on July 26, 2006.
- [10] L. Williams & R. Kessler, "Experimenting with Industry's "Pair-programming" Model in the Computer Science Classroom," *Journal of Computer Science Education*, vol. December 2000, 2000.
- [11] Computing Research Association. 2004-2005 CRA Taulbee Survey. Taken from <http://www.cra.org/CRN/articles/may06/taulbee.html> on July 31, 2006.
- [12] National Center for Women & Information Technology, Promising Practices -- Compelling Education: Introductory Courses, Taken from [http://www.ncwit.org/pdf/INTRO\\_COURSES\\_ED\\_practice.pdf](http://www.ncwit.org/pdf/INTRO_COURSES_ED_practice.pdf) on July 28, 2006.
- [13] Guzdial, M. and Forte, A. 2005. Design process for a non-majors computing course. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (St. Louis, Missouri, USA, February 23 - 27, 2005). SIGCSE '05. ACM Press, New York, NY, 361-365. DOI= <http://doi.acm.org/10.1145/1047344.1047468>
- [14] National Center for Women & Information Technology, Inclusive Pedagogy: Classroom Climate, Taken from [http://www.ncwit.org/pdf/Classroom\\_Practice.pdf](http://www.ncwit.org/pdf/Classroom_Practice.pdf) on July 28, 2006.
- [15] D. Tannen, 1990. *You Just Don't Understand: Women and Men in Conversation*. William Morrow, New York, NY.
- [16] Princeton University, "An Introduction to Teaching at Princeton." Taken from <http://www.princeton.edu/~aiteachs/workshop/gender.html> on July 28, 2006.