

College Faculty Appointment and Review Committee Narrative

How to Use this Document

There are two versions of this document available. A printed version of the document is included within the red book. An electronic version is available on the CD-ROM included with the CFARC packet. The documents are identical except that the electronic version has hyperlinks to supporting material. Words boxed in blue are links to webpages external to this document. Words boxed in red are links to other locations within this document.

“The class has helped me a lot. I was thinking of changing majors because I didn’t know what I was doing at all.” — CS-183 student

“I think this is by far the best class I have taken so far at MSOE. Of course, that might just mean I’m a huge computer geek, but oh well!” — CS-182 student

“[I was surprised] that I’m actually being able to understand what we are learning in class whereas last semester I always seemed lost.” — CS-183 student

“The personality and wonderful humor of our professor in lecture that makes CS-182 a very interesting learning experience.” — CS-182 student

“Having dropped CS-200 last year for difficulty of programs, this class shows us every tool we need to write our programs quickly.” — CS-200 student

“I really like your teaching style. I want you to know that I have learned more from you in the last 4 weeks than I did in the entire term of CS-100.” — CS-200 student (in week 4)

“[The instructor] lectures extremely well on material, admits when doesn’t have the answer, but works to get one.” — CS-321 student

“Dr. Taylor was the *best* instructor that I had this quarter.” — CS-285 student

“I really enjoy class lectures, informative ... but useful, a lot of interesting facts have been thrown in.” — CS-321 student

“I liked Dr. Taylor’s teaching style. It is challenging for both novices and advanced programmers.” — CS-182 student

“[The instructor] is helpful with questions and is willing to change his teaching methods to benefit the students more.” — CS-280 student

“I really learned a lot this quarter. I know I could have understood even more if I had Dr. Taylor for CS-182.” — CS-183 student

“The fairest tests I have ever had, keeps class interesting, gets all the important information to the student.” — CS-285 student

“[CS-182] was interesting. I learned a lot. It was fun.” — CS-182 student

“[The instructor] relates complex concepts in a simple, clear way.” — CS-183 student

“Everything has been straight forward and well explained.” — CS-285 student

I Effective Classroom Instruction

A Courses Taught

EECS FACULTY MEMBER	SUBJECTS TAUGHT (CR.)	NO. OF SECTIONS	CLOCK HOUR LOAD		TOTAL CONTACT HOURS	NO. OF STUDENTS	STUDENT CREDIT HOURS	NO. OF PREP.
			LEC.	LA				
Taylor, Chris	<u>Fall 1998</u>							
	CS-200 ¹	1	2	2	4	14	42	6
	CS-321	1	3	3	6	15	60	8
	Total	2	5	5	10	29	102	14
	<u>Winter 1998-99</u>							
	CS-182	1	3	3	6	20	80	8
	CS-285	2	2	2	4	42	126	6
	Total	2	5	5	10	62	206	14
	<u>Spring 1999</u>							
	CS-183	1	3	3	6	17	68	8
	CS-280	2	3	2	5	20	80	8
	Total	2	6	5	11	37	148	16
<u>Summer 1999</u>								
CS-321	1	3	3	6	4	16	4	

The electronic version has links to each course's website. Each website contains the syllabus, schedule, supplemental material, selected handouts, and the lab assignments for the course.

B Courses Taught for the First Time

All of the courses taught during the 1998 – 1999 academic year were taught for the first time. It may also be interesting to note that of the six new courses taught, I had taken comparable courses to only one as a student. The reason for this is simply that my formal education was in electrical engineering while my teaching responsibilities are in the computer engineering area. I do not raise this issue as a complaint, for I was certainly well aware of the needs of MSOE before accepting my current position, nor should this come as a surprise for those who hired me, for I made an effort to be as transparent as possible about my previous experience. Rather, I raise this issue to acknowledge my dependence on other faculty in the EECS department and to indicate that a substantial amount of my preparation time involved learning the material. I have listed the courses in chronological order since this facilitates discussion of my personal development throughout the review period.

Fall 1998

CS-321 — Computer Graphics

Much of the materials used in this course were based on work done by Dr. Henry Welch. Both Henry and Dr. Mark Sebern provided me with copies of their course notes from previous quarters. These notes were very helpful in my lecture preparation. In addition, Henry, who was teaching two other sections of this course concurrently, developed the laboratory assignments which I used. Much of the success I experienced with this course may be directly attributed to Henry.

CS-200 — Engineering Software Design

In contrast to CS-321, I was on my own for this course. The official course syllabus appeared to be somewhat out of date and did not match the book we were using. I had significant experience programming in C, so the transition to C++ was relative straightforward. I created my own course schedule which followed more closely with the book and the objectives for the course. Some of course objectives were not addressed by our book, and brief discussions with Dr. Mark Sebern helped me develop additional course content to meet those objectives. I developed all of my lecture materials and all of the lab assignments for this course.

The lab assignments were well received by my students. Below is a quote taken from a lab report for a programming assignment involving a Hamming Coder:

“This was another fun lab that was extremely challenging, yet within the scope of our current knowledge. Once again, I was pleased that we were forced to learn about a few things that hadn’t been discussed in lecture. This is something I always look forward to in these labs.

“One additional comment I have is that I like the fact that these labs focus towards real world problems. Using parity bits and now this Hamming coder function are problems that exist in today’s world. I find these labs not only very interesting and beneficial to my learning, but also as a tool for learning about how our technology driven world works.” — [redacted], Lab 6, Oct. 26, 1998

Winter 1998-99

CS-182 — Computer Programming

I developed the lecture notes and most of the lab experiments for this course. Six sections of this course were offered in the Winter quarter. This provided an opportunity share in the lab development tasks. I used three lab assignments developed by Dr. Mark Sebern. In addition, I developed three lab assignments of my own (two of these were used by a number of the other faculty teaching CS-182). I also developed a final project that was used by most of the CS-182 sections.

Students responded positively to these lab assignments. The following are a few of the comments I received in regard to lab assignments I developed:

“This lab has taught me to consider alternative methods of achieving the same final product.” —
[redacted], Lab 3, Gas Mileage, January 10, 1999.

“I have learned more from this than any other lab in my life.” — [redacted], Final
Project, February 19, 1999.

“To have the opportunity to write a program of my own that does many of the same operations
of certain commercial software packages has been a special treat and a learning experience that I
will never forget.” — [redacted], Final Project, February 19, 1999.

“I know how to write multiple function calls, use a vector of vectors, and do useful calculations
on a graphical image, which I wouldn’t have believed I was going to be able to accomplish in the
start.” — [redacted], Final Project, February 19, 1999.

A more complete subset of student comments related to specific lab assignments may be found
in the student work section.

CS-285 — Fundamentals of Data Structures

This was the first time that this course was offered with a laboratory. Previously it had consisted of three lecture hours per week. The current arrangement is two lecture hours and one two hour lab per week. As a result, I developed lecture and lab materials to fulfill the course objectives under this new format. Dr. Mark Sebern was very helpful in providing feedback on my course schedule and helped brainstorm potential lab assignments with me. This proved to be very valuable. Mark's assistance in the planning stages helped to make my experience with this course a success. Prof. Jeff Blessing was also very gracious in providing me with the material he used when he taught the course previously. However, I found that the change in structure was significant enough that I did not take full advantage of the materials that Jeff provided.

I had a course similar to this one as a student (although it was not taught with C++), so I was better able to design material that would minimize busy work and maximize learning. While it is difficult to cover as much material with one fewer lecture hour per week, the laboratory proved to be an especially fertile learning experience for many of my students. As with many of the other courses, students found the lab assignments to provide an opportunity to reinforce the concepts discussed in lecture. A full summary of the comments related to labs may be found in the student work section.

The following are a few of the student comments related to the lab:

“This lab just seemed to be set up perfectly. It offered a great review, the base project was very reasonable in difficulty, and it offered the ambitious student the opportunity to add nearly unlimited extra features. Over all nearly a perfect lab (I am sure something could have been better, I just can’t think of it).” — [John G. Smith](#), Spell Checking using a List, December 17, 1999.

“This lab was a good way to introduce the use of templates. The lab itself isn’t too difficult to do, which really let me concentrate more on understanding the concept of the templates.” — [Ed Griffiths](#), Generic Algorithm Implementation, January 12, 1999.

“I believe this was a good lab that made people think.” — [John G. Smith](#), Spell Checking using a Set, February 4, 1999.

“This lab was really neat.” — [John G. Smith](#), Hash Table Implementation, February 11, 1999.

“I feel that the spell checker application that we have been developing this quarter provided an excellent example of many of the objectives of CS-285. It demonstrated the use of, advantages of, and disadvantages of many different types of data structures. I would definitely recommend that the same 3 step application development (list, set, hash table) be used in future CS-285 courses.” — [John G. Smith](#), Spell Checking using a Hash Table, February 15, 1999.

“I give this lab an A+, mostly because it shocked me so.” — [John G. Smith](#), Spell Checking using a Hash Table, February 15, 1999.

Spring 1999

CS-183 — Software Design

I developed the lectures and helped develop the two multi-week lab projects. Dr. Mark Sebern did most of the laboratory development for this course. I helped mainly with brainstorming ideas, but Mark ended up doing most of the implementation work.

CS-280 — Embedded Systems Software

I developed the lectures for this course. Dr. Steve Barnicki developed all but one of the lab assignments which I used. Steve and Dr. Henry Welch were both very helpful whenever I had questions related to the course material or the 68HC11 briefcases that have been developed for use by MSOE students.

From my perspective, all of my courses went quite well except for this one. I was disappointed with the level of teaching I was able, or perhaps more correctly, I was not able to achieve. As a result my students struggled as well. The subject matter represents a significant shift in thinking from students' prior courses. Unfortunately, I did not do a good job of helping them make the transition. A comparison between the numerical ratings for this course and the other course I taught in the Spring quarter (CS 183) highlights the need for improvement.

While students in both classes seemed pleased with my presentation (3.9/4.0 for both courses on evaluation question number three), there was a significant difference in the scores given for the other questions. For the balance of the questions, I received an average score of 3.1/4.0 from my CS-280 students while I received an average score of 3.9/4.0 from my CS-183 students. There are a number of things that I plan to do different the next time I teach the course. Hopefully, these changes will improve the level of learning.

C New Teaching Techniques

As a new teacher to MSOE, most of the teaching techniques I used were new. Teaching effectiveness is something that has interested me for the majority of my life. As a student, I studied the techniques of my teachers. As a teacher, I have an opportunity to experiment with my own ideas. I have developed a statement of my teaching philosophy. Due to its length, I have included it in Appendix A.

Numerous discussions with Drs. Mark Sebern, Lisa Milkowski, and Henry Welch have helped me develop what I perceive is an effective teaching style. All of them have been very willing to share their own experiences and listen to me talk through some of my challenges. The openness of my colleagues, particularly these three, to assist me in any way they could has far exceeded my expectations, and I am very grateful to them.

Throughout the year I have tried an number of teaching methods. These methods are designed to achieve the objectives set forth in my Teaching Philosophy. I have selected some of the successful methods and listed them below. After each item I have included student comments that pertain to the particular method. There are three sources for these comments: 1) Anonymous feedback forms (see Appendix B) given periodically throughout the term¹, 2) Comments included in lab reports², and 3) CFARC evaluation forms³.

- Get to know my students and use their names in class

“The personal level of attention is very much appreciated.”¹

“You knew mine and everyone else’s names after the first week. Very impressive and adds a personal touch.”¹

- Encourage questions

“The Prof. has been very receptive to questions.”¹

“I like the free style with which you are teaching. You don’t discourage questions, comments, suggestions and aren’t afraid to joke around a little.”¹

- Encourage feedback

“As a new professor at this school I think that you have done very well at listening to the needs of the class and teaching based on that.”¹

“[The instructor] evaluates himself and questions the class to be sure his teaching is effective.”³

- Encourage critical thinking

In response to, “What will you do differently in future courses as a result of taking this class?”:

“This course has helped me focus on the problem solving algorithm. In the future this will help for all classes.”¹

and

“Have a better plan of attack.”¹

- Be available to students

“It is very nice to have you so available for help. There were a few instances where I needed some assistance, and you always responded promptly.”¹

“I like the way the class is taught. I like that the instructor is available for help in lab, class, and office.”¹

“[The instructor is] always available to help.”³

“[The instructor is] great with students names and making office meetings.”³

“[The instructor is] always there to help, [has an] excellent knowledge of C++, [and provides] clear explanations.”³

“Answers emailed questions promptly.”³

“[The instructor] has patience with people. [He] makes sure everyone understands what is going on. [He] makes it known he is ready and available for help.”³

In response to, “What most surprised you?”:

“Your willingness to help us. My teacher for CS-182 told us to read the book instead of trying to explain things to us.”¹

- Retain student interest during lecture

“[The instructor] keeps class interesting and somewhat informal, yet maintains control and focus of class.”³

“Classes are interesting and fun.”³

“The instructor explained the material well and thoroughly. He kept the class interesting.”³

“[The instructor] keeps the class involved.”³

“He gets the class involved well [and] keeps it interesting.”³

“[The instructor] keeps students interested using his personality and wit.”³

“[The instructor] makes the class fun to learn.”³

- Incorporate humor

“I also like your jokes/humor in class, it makes the material interesting.”¹

“I like your sense of humor. Also when you incorporate little stories into your lecture, however short, are always amusing.”¹

“[The instructor] uses time very wisely; follows syllabus exactly, and makes use of every minute; makes classes interesting with his sense of humor.”³

“Your jokes and enthusiasm are what keep me awake.”¹

In response to, “What most surprised you?”:

“That programming lectures could actually be not boring, thanks to your dry wit.”¹

and

“How much fun this class is.”¹

- Incorporate group work in lecture

“I enjoyed the group work, it kept me awake, it made us think, and not just copy from the board.”¹

In response to, “At what moment were you most involved (excited, enthusiastic, ...) in class or lab?”:

“In class when we had to work in groups in writing a program.”¹

- Incorporate examples in lecture

“I like the class the way it is with examples, explanations, and the labs having to do with material we learn in class.”¹

“[The instructor] explains things well, uses good examples.”³

“He relates somewhat dull topics to more interesting scenarios which helps both class attention and (realistic?) applications.”³

“[The instructor] shows a variety of useful examples.”³

- Use technology to improve learning

“The email updates of things associated with the lab are helpful; good idea.”¹

“Electronic submission is excellent idea. . . less paperwork hassle.”³

“Love email lab submission.”³

In response to, “What was the most helpful action taken by anyone in class or lab?”:

“Having the website as a resource.”¹

- Use quizzes to keep students current

“I believe the quizzes are good to keep us current on material.”¹

“Weekly quizzes are a great idea.”³

In response to, “What was the most helpful action taken by anyone in class or lab?”:

“The weekly quizzes – force us to keep up with the class discussion.”¹

- Create stimulating lab assignments – see Sec. I.B and the student work section.

D Ideas to Improve Future Teaching Effectiveness

Listed below are a number of ways that I can improve my effectiveness as a teacher:

- Further develop my expertise in the area of computer engineering. Continued experience teaching will naturally enhance my knowledge. In addition, I have embarked on an aggressive reading program to deepen my understanding (see Sec. IV.E).
- Pursue industrial experience through consulting activities. This will provide for a richer pool of experiences to share with my students. This past year I have done a limited amount of consulting for Direct Supply, Inc. and a significant amount of consulting for the Mitra Corporation. My work with Direct Supply has provided additional exposure to software engineering practices. My work with Mitra has given me a number of additional practical experiences in the areas of computer graphics and numerical methods (see Sec. IV.F).
- Continue fruitful discussions with other faculty on teaching techniques. I have benefitted from a number of suggestions from other faculty in the Electrical Engineering and Computer Science department. I look forward to gaining further insights from the perspectives of others.
- Continue seeking ways to create an active classroom environment in which active learning is commonplace. In particular, seek to incorporate laptop use into the classroom.
- Respond to constructive suggestions from faculty members, graduates, employers, and students.
- Continue to review scholarly educational material such as ASEE Prism, ASEE Journal

of Engineering Education, and a number of conference proceedings related to engineering education.

- I will be participating in the MSOE mentor program this year. I hope that this will allow me to build rapport with students and remain relevant.

E Affect of Student Evaluations on Teaching

As is evident from the volume and content of student comments in this document, I am very interested in using my students' ideas to improve my teaching effectiveness. In addition to the "MSOE issue" student evaluations, I have solicited student feedback one or two other times throughout each quarter. My most often used questionnaire is based on a similar form which Dr. Mark Sebern shared with me. The feedback gleaned from these forms, along with the questions, are summarized in Appendix B.

I have been pleased with the usefulness of the comments I have received. By creating a formal mechanism for student feedback as the quarter progresses, I have been able to either make adjustments when appropriate or clarify my reasoning for a particular policy or teaching method when change is not appropriate. I have found that these additional opportunities for student input have a number of advantages.

1. I am able to make adjustments in my teaching methods to better address the educational needs of my students.
2. Students appear more likely to view me as an ally in their learning process as opposed to someone who is bent on determining how much of the material they didn't learn.
3. Students are more willing to provide me with unsolicited feedback on how my classes could be better.

In general, I have received very positive feedback from my students. In addition, I have received a number of helpful suggestions for improving my effectiveness.

F Summary of Student Work Samples

Samples of student work for all the classes I have taught are included. The samples included consist of quizzes (where appropriate), exams, and laboratory reports. In general, I have included the high, median, and low scoring submissions for each quiz and exam. Beginning in the winter quarter, I graded all lab reports electronically. Electronic copies of *all* of these lab reports are included. In addition, I have included the high, median, and low scoring submissions for the lab reports submitted in the fall quarter that were not graded electronically. The histogram in Fig. 1 indicates the number of comments I made on each report that I graded electronically. For this histogram, comments were defined to include the final score,

spelling/grammar corrections, and any more substantial comments. I averaged 5.73 comments per report. Of course, this means very little without actually looking the graded reports, so please do take some time to scan through them. I am always interested in getting feedback on how my grading could be improved.

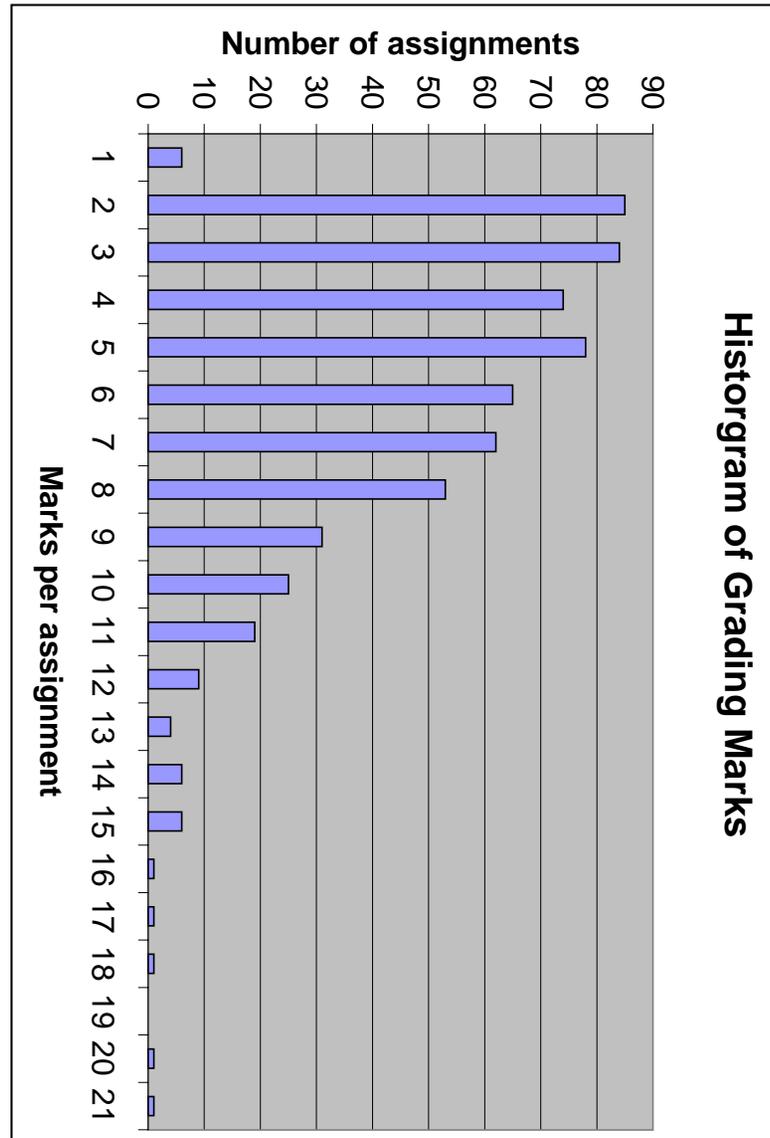


Figure 1: A histogram of the number of comments made on each lab report that was graded electronically.

II Department Activities

A Committees

1998-1999	CE curriculum committee
Spring 1999	CE/SE assessment committee
Spring 1999	CS-100/CS-150 textbook selection committee
Winter 1998 - Spring 1999	CS-100 and CS-200 course development committee
Winter 1998 - Spring 1999	CS-150 course development committee
Fall 1998	CS-182/CS-183 textbook selection committee

B Course Development

I made significant changes to the course syllabus for CS-285. The course was changed from a three lecture hour course with no laboratory to a two lecture hour course with a two hour laboratory. As a result, I made a number of changes to the syllabus to accomodate the change in format.



C Non-teaching Activities

- June 1999 Freshman advising
- May 1999 Industrial advisory committee meeting
- March 1999 Spring registration for international/transfer students
- November 1998 Industrial advisory committee meeting

III Institutional Activities

A Committees

October 1998 Reported to the MSOE assessment committee on my experience at the “Best Assessment Processes In Engineering Education II” symposium

B Administrative Duties

5/10/99 Faculty Council
4/30/99 Faculty Council
4/9/99 Faculty Council
10/30/98 Faculty Council
9/3/98 Faculty Council

C College Mission Activities

- 4/17/99 MSOE Open House – Computer/Software Engineering representative
- 3/27/99 MSOE Open House – Computer/Software Engineering representative
- 3/12/99 MSOE Campus Visit Day – Computer/Software Engineering representative
- 2/6/99 MSOE Open House – Computer/Software Engineering representative
- 11/24/98 MSOE Opportunity Conference – Programming contest judge
- 11/23/98 MSOE Opportunity Conference – Registered teams

D Student Activities

- 5/10/99 MSOE/UNICEF Carnival – Pie-in-face participant
- 4/20/99 College Quiz Bowl – Judge
- 3/18/99 Faculty vs Student Basketball Game – Player

IV Professional Activities

A Professional Society Memberships

- American Society for Engineering Education
- Institute of Electrical and Electronics Engineers
- IEEE Computer Society

B Conferences Attended

B.1 August 1999

I attended the “Rational Rose/Unified Modeling Language” seminar in Brookfield, WI.

B.2 May 1999

In addition, I participated in a “Personal Software Process” workshop here at MSOE. The workshop was conducted by Drs. Mark Sebern and Steve Barnicki as a way of introducing PSP to MSOE faculty.

B.3 October 1998

“Best Assessment Processes in Engineering Education II – A Working Symposium,” Rose-Hulman Institute, October 16-17, 1998.

I attended this symposium for three primary reasons:

- Become familiar with the assessment process;
- Provide extra an knowledge base on assessment for MSOE; and,
- Become familiar with assessment strategies that I could apply to my own courses.

A summary of my experience is available upon request.

C Professional Journals Read Regularly

- *ASEE Prism*
- *IEEE Spectrum*
- *IEEE Computer*
- *ASEE Journal of Engineering Education*
- *Proceedings of the IEEE Frontiers In Education Conference*

D Professional Journals Read Periodically

- *IEEE Transactions on Biomedical Engineering*
- *IEEE Transactions on Image Processing*
- *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- *Circulation*
- *Proceedings of the International Conference on Engineering Education*
- *Journal of Cardiovascular Research*
- *International Journal of Cardiac Imaging*

E Books Read

The following is a list of books from which I have read significant portions. I have done this reading in order to improve my knowledge base in computer and software engineering. This list does not include textbooks used in courses that I have taught.

- S. R. Schach, *Classical and Object-Oriented Software Engineering*, McGraw-Hill, 1999.
- D. R. Musser and Atul Saini, *STL Tutorial and Reference Guide*, Addison-Wesley, 1996.
- S. Meyers, *Effective C++*, Addison-Wesley, 1998.
- J. D. Greenfield, *The 68HC11 Microcontroller*, Saunders College Publishing, 1992.
- J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, 1996.

- A. S. Glassner, *Graphics GEMS*, Academic Press, Inc., 1990.
- R. L. Kruse and A. J. Ryba, *Data Structures and Program Design in C++*, Prentice Hall, 1998.
- J. L. Antonakos and K. C. Mansfield Jr., *Practical Data Structures Using C/C++*, Prentice Hall, 1999.
- L. Nyhoff, *An Introduction to Data Structures*, Prentice Hall, 1999.
- M. L. Marcus *et al.*, *Cardiac Imaging*, Saunders Company, 1991.

F Consulting

I have performed consulting work for two companies. I have done a handful of hours consulting with Dr. Mark Sebern for Direct Supply, Inc. This work has focused on the implementation of software engineering processes within Direct Supply. This work was performed in the spring and summer of 1999.

In addition, I have logged over 500 hours consulting for the Mitra Corporation. My work for Mitra has involved developing software algorithms for quantitative coronary analysis. The project involved image analysis, computer graphics algorithms, numerical methods, physiology, and fluid dynamics. It proved to contain a nice balance between my background in image processing and my current role in computer engineering. This work was performed in the summer of 1999. I hope to continue these activities in the future.

G Scholarly Activities

In the summer of 1999 I reviewed a chapter on color models from the CS-321 textbook, D. Hearn and M. P. Baker, *Computer Graphics, C Version*, 2nd edition, 1997, for Prentice Hall.

H Research Publications

In February 1999 I submitted, with Drs. Jan Allebach and Zygmunt Pizlo, a paper to the *IEEE Transactions on Image Processing*. The paper is entitled “The Image Fidelity Assessor,” and is based research performed as a Ph.D. student.

A Teaching Philosophy

In an information-rich age it would be easy for educators to assume the role of information vendors. Many courses require the transfer of vast amounts of information from teacher to student, but information transfer alone does not produce quality engineers. There is much more to being an engineer than possessing knowledge, and there is much more to teaching an engineer than peddling knowledge. I begin this essay by describing my idea of an effective teacher. I then discuss strategies that I have used and plan to further develop in order to teach effectively.

1 Effective Teachers

Effective teachers purposefully seek to develop certain qualities in their students. Their students should enjoy learning, develop critical thinking skills, take personal responsibility for learning, and have a vision for their future.

Effective teachers are caring, encouraging, organized, enthusiastic, creative, deliberate, visionary, and self-improving. Students are first and foremost human beings. They are more likely to respond positively when they know that someone cares about them. Developing such an environment can be particularly difficult in large classes, but it is no less important. Organized

professors provide clear instruction, convey to students that they care about the class, and serve as models for students to emulate. Appropriate lectures, activities, assignments, and tests are crucial for successful learning. Effective teachers are aware of their students' current level of competence and design material that will move students to the next level. They design:

- in-class activities that relate well to students, foster interest in course material, promote student involvement, encourage critical thinking, cater to multiple learning styles, and follow a logical structure;
- out of class assignments that minimize demoralizing “busy work,” focus on concepts to be learned, facilitate self discovery, encourage critical thinking, challenge students at all levels of competence, reward students according to their effort and level of mastery, and provide a sense of accomplishment when completed;
- techniques for evaluating students that separate students according to their level of mastery of the material, promote studying for future exams, challenge students to think, make cheating difficult, and do not trick students.

Effective teachers have specific objectives for each activity. They have a vision for how their students will be different as a result of their class and are able to transfer that vision to their students. Above all, effective teachers continue to evaluate and improve their strategies in an effort to become even more effective.

2 Maximizing My Teaching Potential

It is critical for students, particularly for first and second year undergraduates, to view their professors as allies in the learning process. A number of steps can be taken to encourage rapport, such as knowing the names of one's students. At MSOE my class sizes are small enough that I can learn my students' names within the first week or two. Another step is to solicit student feedback. I formally solicit student feedback twice a quarter and then review their comments with them in the following class period. This allows me to make changes when appropriate and let my students better understand the reasons for course policies that I do not change. Contact outside of the classroom is also valuable. I anticipate taking on advising responsibilities in the near future, and have volunteered to be part of the mentor program. These will allow me to further develop relationships with my students. I continue to search for ways demonstrate to students that I am interested in them as people, not just as sponges soaking up information in my class. Furthermore, by getting to know my students better, I am better able to design course materials that will appeal to them.

Teaching is a learning process for both the teacher and the students. There are many things that I have yet to learn and relearn as a teacher. I strive to be purposeful and creative in designing activities and material that fulfill the in-class activity, out of class assignment, and student evaluation objectives described in the previous section.



This page was intentionally left nearly blank.

B Anonymous Student Comments

1 CS-200 Feedback in Week 4

At what moment were you most involved (excited, enthusiastic, ...) in class or lab?

- During the 2nd quiz review. That was a tricky quiz, not tough! (It was easy to miss the " \n\n") It would be better if spaces were represented by "b"
- When I first began programming (writing my first program) was fun and still is.
- After finishing my first C++ program (Lab #2), it was a great feeling knowing that I figured it out. I also like your jokes/humor in class, it makes the material interesting.
- Seeing a new teacher. Day one or two of class was very interesting having a younger teacher. ... open to new idea/suggestions. Identifies with what we are thinking. Specifically when explaining need for pseudocode.
- Once I actually set my programs up and running. Also, when student participation/involvement is integrated in lecture.
- When we had done our first lab.
- When we started actually programming in C++. (7)
- Probably class, but the lab is harder and it tends to take me extra time to finish labs.
- A little bit of both, class goes smooth and lab is challenging.
- When I got the second lab to work out.
- Some of the labs have required us to use programming techniques that weren't discussed in lecture. Making these discoveries and getting them to work was exciting.
- Programming the lab.
- No comment (1)

At what moment were you least involved (bored, disconnected, ...) in class or lab?

- During most of the initial class sessions, until you started getting into the fun stuff ... handling input and output files and output formatting. (2)
- The first few days of class were boring, but that is to be expected.
- After lab # 3 session. I felt overwhelmed, I wasn't sure if I was headed in the right direction.
- When lecture becomes just you talking and no one has questions it is easy to sort of drift into other thoughts.
- When the lecture is just one-sided, though I do understand that that is sometimes necessary to get information across

efficiently. (3)

- Going through arithmetic, but I do realize it is different when writing programs.
- Discussing methods and operations in class.
- Well, lab, but I think I'm beginning to understand them better now.
- In my room, working on the lab, debugging.
- When I did the first lab.
- The monotony of lecture, although important, gets boring at times.
- The overemphasis placed upon what the computer will do if you make a logic or syntax errors.
- No comment (1)

What was the most helpful action taken by anyone in class or lab?

- Everyone voting to make the test open book and open notes. After all, in the real world, programmers refer to textbooks on the job!!
- When people asked questions that helped me in the process.
- When Mark decided to drop – He was distracting and made it hard for me to focus.
- Basically when you requested that Mark not comment if he didn't have questions.
- When someone requested that the lab assignments be made available to us a little earlier. Also, when Mark decided to drop.
- Someone pointing out the fact that if I set a fix, I have to unset it as well.
- Explanation of the syntax needed for certain commands.
- I think me and [student] studying the night before the test helped me a lot.
- When questions are asked, gives another view. (3)
- When you (Dr. Taylor) would stop and answer questions instead of going on.
- The weekly quizzes – force us to keep up with the class discussion.
- Good examples.
- No comment. (1)

What was the most confusing action taken by anyone in class or lab?

- Trying to figure out the labs.
- Lab #3 – it confused me.
- Anytime that Mark talked. Ok, but seriously, On the first lab, the requirements for the write-up were kind of unclear. But that situation has already been rectified.
- When we just flew through the file stream part of the book.

- The quiz on the out from a program. (# 2)
- Questions on test and quizzes are vague and leave room for interpretation.
- None yet
- No comment (6)

What most surprised you?

- 10 on the 4th quiz!!
- How much C++ I already knew from last year.
- The quizzes following the first one. They were hard to understand. (I know the info, just wasn't prepared for the format.)
- Having dropped CS-200 last year for difficulty of programs, this class shows us every tool we need to write our programs quickly. I'm no longer spending 6-7 hours a week to finish the lab.
- That programming lectures could actually be not boring, thanks to your dry wit.
- Some of the quizzes that I was given. (2)
- The fact that this class was teaching how to program and not just talking about it.
- The amount of programming required to pass the course.
- The rate at which we are moving. I feel we are moving a little too fast.
- Your ties.
- How quickly we jumped into writing code. This was a good thing and helped me to learn things on my own.
- How much I already knew. (2)
- How are test questions that involve syntax errors easily determined when compiled valid test questions?

What would be the first thing you would do differently if you were teaching the class?

- Go faster through the initial class material in order to start with the "pointers" section.
- nothing... just try to keep it fun.
- Set aside half of the class period on Thursdays to discuss the lab material. That way we have a good start come Monday. And possibly tell students what type of quiz they will be taking.
- Try to get a room with actual tables. Trying to have a notebook and the text open at the same time is not very easily done.
- The only things I would change would be to give explicit requirements for lab write-ups and make the labs available earlier, but both of these have already been taken care of.
- Have quizzes every other lab.
- Attempt to spend more time in lab and less in class.
- Take a little more time to explain some thing about programming.

- Lecture more in lab, about assignment. (2)
- Go over labs more before the lab itself.
- Try to discuss the lab a bit more in detail during Thurs. lecture. This will help us to use Monday's lab for actual programming. (2)
- Nothing.
- Speed the pace up.
Give different examples than the book's.
Flowchart instead of Programmer's Algorithm.

Additional comments:

- I really like your teaching style. I want you to know that I have learned more from you in the last 4 weeks than I did in the entire term of CS-100. Thank you for taking time to answer questions and help me. Your lecture notes are a great resource and your humor makes it more enjoyable.
In summary – The open book/notes test was a plus. I would like to see partner work, because in the engineering field the majority of the projects will be done with partners. Also, seeing another persons view point might make us learn more → different approaches.
- This class is much more enjoyable than my earlier C++ class. Things are explained much better and assignments fit pretty well with what we are doing.
- So far so good, let me see my test.
- I enjoy the class a lot. I feel that I understand C++ somewhat.
- Excellent professor! But he needs a bit more experience teaching this subject.
- Condense the lab write-ups:
 - Title page
 - Summary
 - Source code
 - Output codeTest and quizzes are too picky and do not measure knowledge of material.
- No comment. (7)

2 CS-200 Feedback in Week 8

How well do you think you have learned the material covered in this class?

- At first I thought I wasn't learning it as well as I should. (Based on weekly quiz scores.) But through labs and going over quizzes I was able to raise my test average. Now I think I have learned the material pretty well.
- Very well, no problems.
- I feel I have learned a lot about C++ programming through actually programming in the lab.
- I think that I sort of understand things, but its just the little things that get me.
- Just well enough to understand the basics.
- Maybe about 75%. I'm not a good programmer and may never be, so you move a little to fast for me. I'm sure others who knew more about programming before the test had an easier time than I so far.
- Quite well! It would be better if we could do some real world examples using "objects" and "class."
- Although I don't know it cold, I know it well enough to be able to find out anything I don't know.
- Personally, I have learned this material very well.
- I think I probably learned more in this course than any previous class.
- Very well, I understand the language and syntax to the level we have covered.

How well do you think your peers have learned the material covered in this class?

- Some of them don't have a clue, but the majority have a good understanding as to what's going on.
- Ok, some seem to have problems.
- I think that everyone in this class learned a lot.
- They have got to be doing better than me! So I think that they understand.
- Better than others; depends on individual strengths.
- Some better, if they knew how to program before the class. I came into the class with virtually no clue.
- I am not too sure about that.
- My best guess would be that they're at about the same place I am, except less cool.
- In discussing some of the labs with other students, it is my opinion that not everyone knows the material.
- Very well.
- Some very well, others hardly at all.

What could you have done differently to have improved your learning experience?

- Make more time to study for the quizzes. It always seemed that I either didn't study enough (not enough time) or studied the wrong material. some extra things that weren't in the lecture.
- Eliminate the little mistakes!!!
- I tried hard in the beginning. I've slacked off a little more later. I wish it had went slower and more complete in the beginning.
- I would have got my computer account fixed first.
- Taken more time to flat-out learn the material, time allowing.
- It may have helped if I had read the textbook. I relied on lecture rather than the text. (5)
- Unknown.

What will you do differently in future courses as a result of taking this class?

- This course has helped me focus on the problem solving algorithm. In the future this will help for all classes.
- Use the problem solving techniques.
- Read the book a little more and not wait until the day before to do my programming.
- Break down problems!
- Have a better plan of attack.
- Use C++ whenever required in other classes. . . (I don't have to take any more classes here!)
- Always try to take more time to thoroughly understand the material, not just learn enough to get by.
- Hopefully, I will understand C++ enough to be able to write the programs necessary in future courses. Also, I hope to incorporate the general idea of programming when I do other types of programming (microcontrollers, etc.)
- Make section selection based more on professor. A good professor definitely makes a difference.
- Probably nothing, however I will know that I could write a C++ program for some problems. The algorithm steps will help with any problem with multiple steps.
- No comment (1).

What could I have done differently to have improved your learning experience?

- Tell us what to study for the quizzes → make a deal with us so that if we do well (or well enough) on the tests that the quizzes won't hurt our grade.
- Show us other ways to write our programs. (Ex: If there is a more complicated way to code something, show us and give us the option of figuring it out.
- Spent more time on passing by reference!!
- More forgiving on tests.
- Just have taken the time to make sure people who aren't good programmers understand what's going on instead of teaching

at the other student's pace.

- Word the test very clearly, so that every student understands what exactly you have in mind.
- Ya done good, buddy-boy. (West Side Story)
- It may have be beneficial to move a bit quicker through the material in an effort to cover more topics.
- I don't know. Examples and humor made class interesting, and up to now, the programs and tests have been fair. This lab may prove otherwise?
- No comment (2).

Please add any additional comments you have. Indicate if you do *not* wish to have them appear in the summary returned to the class. Use the back of this page if necessary.

- Overall I think the labs have been a bit too advanced.
- It is very nice to have you so available for help. There were a few instances where I needed some assistance, and you always responded promptly.
- The personal level of attention is very much appreciated.
- As a new professor at this school I think that you have done very well at listening to the needs of the class and teaching based on that.
- The reduced lab reports saved a lot of time for me.
- No comment (6)

3 CS-321 Feedback in Week 5

At what moment were you most involved (excited, enthusiastic, ...) in class or lab?

- When we help work or find algorithms on the board.
- I am primarily involved in lab because I am having trouble with the labs.
- I find myself involved with the labs on a whole. I enjoy the challenge but would like them explained more thoroughly.
- Labs tend to be quite difficult, yet challenging. I was most involved so far in lab #1.
- I was enthused when there was discussion on line, circle, etc. Algorithms.
- When topics directly related to lab were discussed. Even the slightest hint can trigger hours of productive work.
- When I got the dots to work in the shell after about an hour.
- So far I have enjoyed pretty much the whole class. You are good at teaching.

- In class – while discussing how to generate the actual drawing point.
In lab – when I actually understood how upcasting actually works.
- Coding is the most enjoyable aspect of this class. Unfortunately, it is also the most *frustrating* aspect of the class.
- I would say in lab once I got things to work, but that's obvious.
In class, I was involved in things I haven't learned yet, like the fast way of drawing circles.
- Most of the time. It seems that most of the topics discussed are interesting.
- Not too exciting.
- No comment.

At what moment were you least involved (bored, disconnected, ...) in class or lab?

- I was most bored during the mathematics review lecture, this should be eliminated. (5)
- These labs are different from the ones in the past classes... easy to get frustrated over.
- When Welch was explaining polymorphism. (2)
- In lab – While reading the overly vague lab assignments that want us to be experts with things we've never been introduced to. (2)
- Not too boring.
- No comment.

What was the most helpful action taken by anyone in class or lab?

- The Prof. has been very receptive to questions.
- On lab 2 the whole class worked together to make sure everyone understood what we were doing.
- Someone showed me how to and what to edit in my "Makefile".
- It is very helpful when everyone discusses the problems that they are having with the lab. I think this is a natural reaction so I'm sure it wouldn't work if forced.
- In lab – comment made by previous graphics students: "Forget everything you ever knew about programming because none of it is true for this class."
- Combining Lab 3 and 4.
- Anyone who gives me a tip to solve a problem in lab.
- A certain classmate helped me figure out lab 3. This was probably the most helpful. Having the guest lecture from Dr. Welch was nice.
- None seen so far.
- No comment. (3)

What was the most confusing action taken by anyone in class or lab?

- Labs... are just handed out, and although they are somewhat descriptive... They are confusing, and written by someone else. Who do we ask for help? (5)
- Dr. Welch is an extremely intelligent man; however, he thinks that everyone is at his level, when he lectured in our lab I thought he was speaking Greek. (2)
- I was a little lost on the filling algorithm stuff when you first went over it.
- None seen so far.
- No comment (3).

What most surprised you?

- That my program ran!
- I really enjoy class lectures, informative ...but useful, a lot of interesting facts have been thrown in.
- The inefficient and error filled coding of the graphics shell.
- Simplicity of the lectures, difficulty of the labs. Amount of work is quite high in addition to work in other courses.
- The night and day difference between this class and the previous programming classes. I think it is too big of a jump. This would be the case with any other teacher also.
- Dr. Taylor's knowledge of L^AT_EX and his willingness to admit I may know some more about Unix than he does.
- That the labs were really as bad as they were made out to be.
- The lack of support from Welch in his labs. No info on Makefiles. *[This is my fault. I thought that I had gone around to make sure everyone understood the Makefiles, but it appears that I missed a few.—CCT]* Have labs more structured and less time hogs.
- The difficulty of the labs.
- How little direction we are given on certain lab assignments.
- No comment (2).

What would be the first thing you would do differently if you were teaching the class?

- Explain the process behind the labs more. Discussion on Makefiles would help.
- – Spend more time discussing the Shell program. Since most of our work is being done using this, it would help having more insight.
- Spend more time discussing essential X Windows functions that we would be using for labs and how to access/use them.
- More exams. 1 mid-term has too much stuff to study for. Either weekly quizzes or 2-3 exams to balance the study time involved.
- Upgrade textbook from C version to C++ version.

- Aid the students more in lab. Ultimately the students are going to obtain the necessary skills needed, but with more help, the students may not feel so intimidated by the projects and could possibly be able to extend beyond the basic requirements. (5)
- I would be using the upper-left corner and $+y$ would point down.
- Get a prereq. to programming in X-windows and that environment, as opposed to handing pre-written code and saying you *will be required* to use this code.
- I would ditch Welch's labs. I would have the same ideas, but developed more so a retard like me could understand it.
- Learn to use humor effectively, it is a powerful tool. Try to come up with more real world examples.
- No comment (2).

Additional comments:

- I understand this is your first experience not only with this class, but also C++. You have been doing a great job, and I can tell you are really trying. But remember our coding on labs have been broken up... no continuous and we have forgotten a lot of the upper level stuff.
- The class thus far is enjoyable (of course we haven't had our first test yet).
- Make the labs accomplishable for those of us who are full-time students (19 credits) and work. We have other class and things we must do besides coding for graphics.
- Remember an outline is not a schedule. Don't be afraid to slow labs or speed up sections students understand. Good work, keep it up.
- I have no doubt about your abilities, but MSOE should have not thrown you into teaching this class so fast. Possibly before next term, if possible, review Welch's labs and understand them so that you and the students will have easier time. Also possibly familiarize yourself with the questions the students had this term for next term.
- I think I would change from individual lab assignments, and form groups of 3 or 4 to do the project. In real life you rarely have to work alone. Also lab and class should be more tightly integrated. More should be taught on X, Motif, and Unix library functions.
- The labs are too difficult at times because they're somewhat vague and not many resources are available to use. Don't recommend a book that is out of print!!!
- I am not sure if you are doing this or not but since you don't know C++, I think it would be a good idea to do the labs on your own at the same time we're doing them. Then we could learn from each other.

4 CS-182 Feedback in Week 3

At what moment were you most involved (excited, enthusiastic, ...) in class or lab?

- In the lab coding. (7)
- During the third lab. That's when it started to get interesting. (5)
- Explaining each details and be sure that all the student got and understood the main idea.
- In class when we had to work in groups in writing a program. (4)
- When we worked on Lab 2 together as a class. (2)
- Participating in planning the lab programs (the approach to the lab).
- Class – every class period has good explanations and opportunities to ask questions.

At what moment were you least involved (bored, disconnected, ...) in class or lab?

- Classes run smoothly keeping it interesting. [I'm not sure why this comment was made here. – CCT]
- Repetitive numbering types in the class.
- During some class periods where basic programming was discussed.
- During lectures, sometimes I get bored going over the same thing over and over when I already understand. (3)
- When I ask a question and I don't understand the answers. (personal problem, I shut down).
- Waiting to be able to write a program in lab.
- In lab when some of the material and stuff on quizzes were not making sense to me.
- During chapter 1.
- Going too fast when we were discussing one part of a lab assignment.
- When we were learning the order of operations because we have learned that since 5th grade.
- Lab – definitely yesterday's lab – lab 3 – explanation of program involvement got *very* confusing and I couldn't keep up with what was being discussed – a lot of info in the program is unclear – “junk” for example.
- Going over machine language.
- I have yet to become truly bored in this class. While I can pretty much hack out a program if given the time, I am learning that doing so is not only inefficient, but error prone. (3)
- No comment (3)

What was the most helpful action taken by anyone in class or lab?

- People asking questions. (7)

- Sharing notes.
- In lab, help with programming errors.
- One kid explained what parts were entailed in a while loop.
- When the student share their idea with the teacher.
- Programming syntax examples on the board.
- Class involvement putting things on the board.
- When you explained ++i and i++. Those are very useful shortcuts.
- Having the website as a resource.
- Having others spot check code for bugs.
- I think when one of my fellow classmates spotted our error in the “Celsius to Fahrenheit” lab proved to be a most helpful moment. He suggested using floats so that the fraction would not evaluate to 0.
- No comment. (3)

What was the most confusing action taken by anyone in class or lab?

- None. (2)
- Trying to figure out what the heck other people are thinking sometimes.
- The command for C++ get me confused.
- Trying to explain how Microsoft C++ worked... meaning getting a program file, name, etc.
- It is confusing for when the program itself (Visual C++) doesn't work or has problems.
- The most confusing action was taken by my computer in the lab when it suddenly decided that I shouldn't be let into the MSOE network.
- In class putting single and double quotes on an expression and saying why.
- Going over hexadecimal, octal, etc.
- How using just one = sign sometimes produces 1.
- When students ask questions from past programming experience that the rest of this class doesn't understand.
- None that come to mind.
- The electronic submission is still a little confusing. (I don't need another explanation, I just need to do it a few more times.)
- Electronic submission deserved more explanation.
- No comment (5)

What most surprised you?

- I thought there would be more days on the computer.
- How different C++ is from BASIC.

- That C++ is easier than I expected.
- How much I understand without ever having this before.
- Your strange sense of humor.
- The misunderstanding for the questions in the quizzes.
- The flexibility of the class.
- The quickness of learning the language of C++ in class.
- The fact that there are escape operators and I had never seen that.
- How easy it is to use character strings.
- That someone could keep track of 6 years of filling up your car with gas... and a nice challenging lab.
- Having to declare every variable and how dividing 6 by 5 gives you 1.
- Lab #3 was quite a bit more complicated than Lab #2, although I don't think that is a bad thing.
- Questions and ability level of others in class – about same as me – and their questions relevant to mine.
- How incredibly picky the compiler is.
- The addition of file stream I/O surprised me greatly. I think it's really interesting.
- No comment (4)

What would be the first thing you would do differently if you were teaching the class?

- Handouts of sample code relevant to what is going on with material.
- Nothing, I like the method of instruction.
- Relax, sometimes you seem nervous up there. I think you need to feel more comfortable with people.
- Focus more on programming than talking about how to program. More time at the computer. Thing discussed can be seen in action instantly.
- I would use the overhead with program segments typed on transparencies to save time writing on board. Use of overhead also allows changes to be made when everyone can see them on the projection screen. (2)
- I would make the quizzes as an extra grade rather than making them as a part of the whole grade.
- Less coughing. Just kidding. I would try to get more hands on, and a little better explanation of how visual stores programs, and what it does when it is rebuilding all.
- Have a quick review at the beginning of each class of what was learned in the previous class, and take questions for about 5 minutes.
- I would get more group work. Explain the material a little more. Curve grades.
- Nothing.
- Maybe explain syntax more than theory a little more.
- I would teach the loop much faster because you can't do much without it.

- One on one lab assistance.
- Teach labs *slower* and explain each “foreign” term.
- Make sure everyone has a good start on the lab before they leave.
- I don’t believe I have the knowledge to come close to teaching this. That would be like telling Mark McGuire or Sammy Sosa how to hit a home run.
- Try to get a different lab. The current one is too cramped, hot, and the computers are slow.
- No comment (2)

Additional comments:

- Sorry for the vagueness of some of my answers but, I think you are doing a good job.
- More lab times open, hard to get into a lab that has C++ to work on projects/labs.
- No complaints thus far. You seem to have a handle on what we need to know.
- I would like it to go faster, but I understand that everyone has had different levels of experience.
- Quizzes should be counted a little less.
- So far everything seems fine. C++ is not as hard as I thought it would be.
- Overall pretty good – keep things *slow* and *simple*.
- I believe the quizzes are good to keep us current on material.
- No additional comments at this time.
- The class has been perfect so far.
- No comment. (8)

5 CS-182 Feedback in Week 7

At what moment were you most involved (excited, enthusiastic, ...)?

- In lab. (3)
- While we are group programming. (4)
- Maze lab where their was extra credit. Satisfaction of going beyond the requirements.
- I love writing code, it’s sometimes puzzling which makes it fun to solve.
- Class.
- Most involved and enthusiastic at the start of the term.

- When we started talking about GUI and during group projects.
- When I woke up one morning and I knew how to do that maze program.
- I was most involved in everything but the most exciting part so far is the labs.
- When I got the mileage program to work.
- Watching my programs work.
- Creating lab #5 (hamming coder), working out the algorithms. (3)
- When I found out we were going to work with images.
- The “tag-team” activity we did in class was also fun.

At what moment were you least involved (bored, disconnected, ...)?

- Lectures on C++ that don't involve examples. Not so much anymore because all new keywords have accompanying examples.
- It's hard to concentrate in lab.
- Recently after the break, it all seemed to go downhill. No longer captured interest.
- When we talked about libraries.
- When somebody asks a question after you just explained it. Why can't they pay attention?
- When I decided that I didn't want to sit in front of a computer programming for the rest of my life.
- Sometimes during lecture. (4)
- Never a dull moment. (2)
- During the first 1.5 weeks of class.
- The last time we did evaluation forms!
- Trying to think of all the possibilities of data for the hamming coder.
- When you started talking about the pixels on the window or something of that sort.
- No response (2)

What was the most helpful action taken by anyone?

- Helping others with programs.
- Talking through a small program when we were in groups of three. (2)
- Questions. (3)
- Classmate's suggestion that class was moving too fast.
- People throwing out different solutions to the same problem. (2)
- Finding the lab 5 webpage for me.
- Class, explaining topics.

- When someone sat down and went step-by-step through a program with me. That helped me understand exactly what was going on.
- Answering questions in class and hands on help in the lab.
- Dr. Taylor explaining functions: scope, initialization, etc. . .
- No response (3)

What was the most confusing action taken by anyone?

- Sometimes when stuff on the board is written too lightly and I'm too far away, but I suppose I could move to fix this confusion.
- My programs are probably very confusing looking, but the work. . . I think.
- The creation of the labs seem at times too complex for what we are learning, but that isn't exactly a bad thing. Just go over it more.
- When someone said a formula would not work because of the order of operations and it did.
- When people say things and they have no clue what they are talking about.
- When someone tried to explain the ++i and the i++.
- When I accidentally deleted a curly brace and sent the compiler into convulsions.
- Learning the difference between call-by-value and call-by-reference.
- When you introduced the ampersand. I really didn't know what you meant by it, until later on.
- None (3)
- No response (6)

What most surprised you?

- Simplicity of C++.
- Final project requirements. . . but hopefully looking forward to the challenge.
- The personality and wonderful humor of our professor in lecture that makes CS182 a very interesting learning experience. (2)
- The fact that you have to prototype functions. Why is that anyway?
- How lost I am after missing one class.
- How fast the level of difficulty in our labs increased. (3)
- Open book tests.
- GUI's in CS-182 (that's a good thing).
- How easy this language is to learn.
- Lab #5 compiled and executed flawlessly on the first try.

- How difficult it is to have the hamming coder work correctly.
- I was surprised by two oddly rather simple things: `assert()` and `clrscr()`. I never had heard of them before.
- How the compiler is really annoying.
- No response (2)

What would be the first thing you would do differently if you were teaching the class?

- Keep it the same.
- Have the class taught in a computer lab (A/C, of course).
- Require flowcharts or pseudocode before a lab began.
- More group work.
- Give out more sample code. The book has such fragmented chunks that it would be helpful to see an implementation of an entire sample program.
- Spend more time addressing concepts regarding labs.
- I would have more programs that used the GUI structure.
- I am still anti-write everything on the board. I am a big fan of overheads. It takes too long to write out. Of course, you are probably not as lazy as I am.
- Less tests, more labs. (2)
- I really don't know.
- Explain more on the *how* to implement a program than the what.
- Nothing, you're doing great. (2)
- Be more proactive. Keep students informed as to what they should know by this point. Give the class a little more direction. (The final project helps a lot with this!)
- Have the correct output for the labs available for lab verification.
- For the most part, I feel the class is exceptional. However, I think it would be helpful to the students (this is not regarding CS-182) if an archive of extra lab projects was made available so that students may hone their programming skills over summer vacation.
- I would keep the group work. I think that the final project is good too. But what I would do differently is to try to explain things in ways that even people who don't know programming would understand them.
- No response (1)

Additional comments:

- Look, my comment is on the sheet and yours isn't.
- I think this is by far the best class I have taken so far at MSOE. Of course, that might just mean I'm a huge computer

geek, but oh well!

- Its very uncommon that a college professor knows all the names of their students. I think its great that you do.
- I think everything is going pretty well.

6 CS-285a Feedback in Week 3

At what moment were you most involved (excited, enthusiastic, ...) in class or lab?

- When I think I understand the material, I get more involved.
- I was most involved during the review of C++, since it had been awhile since I had programmed in C++. (2)
- When I knew I would learn new features about C++.
- Finding out that we will be learning about the STL. Also, creating programs that have some value, use.
- Going over Big O , or other new stuff. (3)
- Actual programming – lab. Going through examples – class. (2)
- First day of lab seeing how much programming I could remember.
- When we got the first assignment for the spell check program, I immediately began coding. Couldn't wait to start programming again!
- I wasn't. (2)
- We've only had four class periods and on functional lab period. Nothing particularly exciting has happened in that time, but I think that's just because it's all been review or introductory material. (2)
- I felt most involved during lecture in class, when we collaboratively answer questions asked by the teacher.
- I was enthusiastic about taking this class because I really do like C++. I was excited one day in class when I actually understood what you were doing.
- In class

At what moment were you least involved (bored, disconnected, ...) in class or lab?

- I am least involved when we are going over the basics, or things that are too complex for me to understand at the time.
- Only last lab when I had projects due in other classes. I really had to leave early and I was pretty “antsy” or that Big O stuff.
- Big O (4)
- The C++ refresher part of the first wee was a little boring. “Card game.” (6)

- When going over all the code from book example.
- When you started talking about recursion (ick) but mostly that very first lab.
- Finding out that we were not going to use GUI.
- I started feeling a little disconnected when I didn't understand a concept, but it all made sense when I asked my question after class.
- When you were talking about complicated algorithms without discussing their development and without showing us how they really worked. I cannot just look at something and know how it works, I have to study it very well and I felt that you were talking over my head.
- In class (2)
- No comment (1)

What was the most helpful action taken by anyone in class or lab?

- When you pose problems and help us to think why we need a specific solution and then offer the new idea/function etc. . .
- When you make it *interactive*.
- The C++ review was helpful.
- I had a problem with one of my labs and a friend helped me understand.
- Reminding me the basics needed for a header file.
- Questions.
- Doing examples in C++ on board, not just theory.
- Talking about the logarithmic search.
- Sending in the emails with the questions about our lab.
- Nothing yet. (2)
- Contributing to discussion, because when they say something that isn't totally correct, I feel less intimidated to ask my question.
- Stopping you in the middle of your explanation to ask a question that was bugging me about your code.
- No comment. (7)

What was the most confusing action taken by anyone in class or lab?

- Confusing the lab due date (one of my friends did this, seriously)
- Electronic submission of labs. Never done it and I don't really use email that much.
- The electronic submission was something new to me. I would rather attach my source code as a ZIP file or print out a hard copy, as I have in the past. Formatting the email correctly is very time consuming.
- Part of the report.

- I am sort of confused with the algorithm right now.
- When class isn't involved...
- None. (2)
- Not that I remember. (2)
- I missed the second week of lab, so I can't really answer these questions pertaining to lab.
- The only confusing things happen when there is code on the blackboard that I do not understand and you go ahead to the next topic.
- No comment (9)

What most surprised you?

- A Prof with a sense of humor??? (that's a good thing...) (2)
- Having any review at all. Most classes don't. It's a good idea though, since most of us haven't programmed in 6 months.
- If I say, it may come back to haunt me!
- That we are going to do things in this class at a very fast rate.
- This class seems to have a very precise reason... algorithms.
- How much this sheet is like Dr. Sebern's... is this a CE dept. requirement? I wasn't any good at these then, and I didn't get any better.
- You knew mine and everyone else's names after the first week. Very impressive and adds a personal touch.
- How much I had forgotten. (2)
- How much I remembered about C++.
- You being 25 with a PhD. I admit you have skill.
- Nothing. (2)
- They added a lab to the course.
- The way our labs are all graded electronically in a neat and formal manner.
- The way you explain concepts is a lot different than what I am used to. (I am only comparing you to my other computer programming teachers.) I always felt reassured because these teachers had a great deal of expertise in C++. I have heard that you do not have this expertise, but you have done other types of programming. Do you feel that you know C++ well enough to teach a higher level class? And if you do, would it be possible for you to explain the concepts in a simpler manner?
- No comment (3)

What would be the first thing you would do differently if you were teaching the class?

- The best CS Prof I've had runs through thought processes with us... He led us into understanding why we needed specific

ideas/functions/solutions etc. . . Then we understood why we need to know what we're learning and in turn are much more interested.

- Dress casual – jeans. Nothing too important.
- I would probably show the entire code of the WAR game program to show how it all fits together.
- Couldn't tell you.
- Try and get the class more involved in the class. Force question, make sure you are getting positive feedback.
- Probably make the first program due in a shorter period of time. (was not too hard) (2)
- More groups not just all taking notes.
- I would make transparencies of programs that are written on the board. It would save time and make the class move somewhat smoother.
- For review, go over as much basic stuff as possible (reading from files, sorting, comparing). It's been a long time since some of us have looked at C++.
- I'd try to be really cool you know. Educate, but be on the students level. You just left this level, you can easy like come back.
- I don't know, really. I've never had to teach a class, and I can't say that you're doing anything wrong. Just because I feel the intro stuff is dull doesn't mean it's not necessary.
- No offense. . . teach for today's use which includes programming for windows interface.
- Make handouts that we can use as reference when studying or writing a lab. Maybe the way to drive a point home is repetition. Maybe small homework assignments that would make us read the book would help.
- No comment (5)

Additional comments:

- So far so good. I *am* learning.
- 4 gallons a week?!
- Loosen up a little.
- The email updates of things associated with the lab are helpful; good idea.
- As we go through programs can you show us how to make it "user friendly" things like assert, error and such?
- Good job, be more confident.
- Uhh. In all honesty (or maybe laziness) I don't like the extra submission thing, but I guess I'm not going to like a lot of things in the work place, but. . .
- It's much too early for this kind of thing. I'd probably have more to say after a few more classes, and particularly after more labs.
- I think you are doing a great job keeping the classes attention and keeping the boring material fun.

- No comment. (11)

7 CS-285b Feedback in Week 3

At what moment were you most involved (excited, enthusiastic, ...) in class or lab?

- When we started covering new info on the different STL containers.
- None.
- During the war card game.
- At no real point was I very excited in class or lab but I've never fallen asleep in class or lab.
- When we were doing the war game.
- Testing the Greatest Common Denominator (divisor) Function (see answer to surprised).
- When we went over things that I remembered from last year but not fully. The refresher was good.
- I really liked the review the first few days you explained well and I had forgotten a lot since I hadn't been using C++ for awhile.
- When my program actually worked.
- In lab, working out the program.
- When we have lab, it's "hands-on" and more involved. The lectures are good, but can anyone really get enthusiastic about lectures? Love the small "jokes" you throw out.
- I was most focused when I was in lab working on my project. It made me think and not just copy the blackboard.
- In lab when I learned how to start an X session with eXceed. Much cooler than using telnet.
- I have been kind of excited to get back into a computer-related class, after having 2 quarters without any.
- I like your sense of humor. Also when you incorporate little stories into your lecture, however short, are always amusing.
- Just getting back into the programming mode after a year of inactivity in it.
- A little of both. Each have their ups and downs but mostly in class. Possibly lab later on when we get into more stuff.
- When I'm programming in the lab.
- During the development of simple algorithms and determining "Big-Oh" values for them.
- No comment. (1)

At what moment were you least involved (bored, disconnected, ...) in class or lab?

- When talking about the "Big-Oh" stuff. I was disconnected because I was somewhat confused by it.

- Can be a little boring going through the big-oh but that also has to do with the subject matter.
- I haven't been very involved at all but I haven't been bored.
- Studying $O()$.
- When we went over the concepts of the previous classes, it was very slow.
- Not really ever. I think class is running very well.
- O notation, I don't understand.
- Filling out this survey. I'm not too good at taking surveys.
- O notation is a complete loss for me.
- No jokes being announced. At the beginning of the quarter when we reviewed, even though I needed it.
- In class there is not much involved besides sitting down, taking notes, and learning. Face it, we aren't learning exciting stuff here.
- During the C++ review lectures.
- Some of the lectures have been boring – they don't always keep my attention. Big O notation, for example, seemed to just drag on.
- When you're just writing code on the board, however I understand it is necessary.
- Big " O " notation. I understand what it is, but become confused when applying it to programs.
- A little of both, but at a very minimum. I love this stuff.
- Haven't been yet.
- The first few review days.
- No comment (2)

What was the most helpful action taken by anyone in class or lab?

- I think you responding to everyone over email when you received a question.
- The most helpful was going through the WAR card game because it helped refresh my memory of syntax.
- Asking questions and then having you answer them to everyone through email.
- The most helpful action was when people asked questions (Error 4786) in lab and you emailed the answer.
- When *censored* asked why it is called "Big-Oh" notation.
- Posting the questions asked by others by email.
- No one did anything helpful.
- I don't know, I haven't had a problem yet.
- Even though some topics have been uninteresting, they have been thoroughly explained, which has been helpful.
- Someone picked up my pencil when it dropped. Just kidding. I don't really know.
- The WAR example.

- Someone helped me remember how to do some stuff and learn new stuff with VC++ 5.
- When they ask questions and they get emailed back to *all* of us with the answers.
- The emails sent in questioning the lab.
- No comment. (7)

What was the most confusing action taken by anyone in class or lab?

- I am generally confused.
- Nothing confusing so far.
- Everything has been straight forward and well explained. Nothing has been confusing to this point.
- `random_shuffle` command in the WAR.
- Trying to answer this question.
- I didn't understand notation.
- When *censored* tries to show off by talking about stuff nobody else knows about.
- Again, *O* notation.
- No one really confused anything.
- The only confusing thing so far were MSVC++ compiler's warning messages.
- The most confusing part of class is trying to remember how to write code, since it has been a while. I have forgotten a lot, although it has been coming back to me more quickly as we go on.
- Don't know.
- Whenever *censored* asks a question.
- I saw code that changed capital letters to small which took something like 20 lines of really confusing code.
- The first lab until I got some help. Didn't remember too much at first.
- I don't understand this question.
- No comment (4)

What most surprised you?

- That very few people are talking or asking questions in class. Also, that there has been so little programming (code/syntax) so far in class.
- Nothing really about how I expected it.
- My surprise birthday party. Relating the question to the class, nothing has been a surprise.
- I wasn't surprised yet.
- That the Greatest Common Denominator (divisor) function, it blows my mind that it works!
- The fact that you just learned C++ not too long ago and already know it so well.

- That you emailed answers to questions to everyone.
- That I actually remembered so much C++ in lab.
- That I have been able to write this program.
- Your recent experiences to C++. Way you can throw out funnies—makes it interesting and make me more comfortable to be in class.
- How fast you picked up C++ is amazing!
- The fact that the school is still making us use Micro\$oft's product.
- I thought this class was going to be something completely new, when it is actually just going deeper into stuff we already learned.
- It surprised me that you are just learning C++, but yet you are teaching us.
- Nothing, yet.
- A CS teacher that is actually entertaining.
- How much I had forgotton since my last CS course.
- How fast you can learn this stuff.
- The fact that Big-Oh values really don't tell you how fast an algorithm is – just how fast it is relative to something else. I figured there would be an actual time value.
- No comment (2)

What would be the first thing you would do differently if you were teaching the class?

- I would make handouts of my lecture notes, so students could have all the code (and what you have) in front of them. Then they could listen more.
- I would not make any major changes to classes.
- Nothing except if I were the professor, I would write a different name on the board the first day.
- Related $O()$ notation to real time(s).
- In addition to the regular lab, have an optional portion which is more challenging (not necessarily worth more credit).
- Do more things not in the book to give more variety in what is taught.
- I think you do a pretty good job of teaching the class, but your spelling could use some work.
- Labs that are creative. Everybody makes up boring labs that you can't have fun with, we need something a little more open ended and fun.
- Not too much. This is way better teaching than I had in CS182/3.
- I would hold class in the computer lab so when we learned new things, we could try them out on small, test programs.
- Shorten the C++ review to one day.
- I would try and make things a little more exciting, but I don't know how.

- Nothing I can think of.
- Explain Big “O” to programs that are similar to labs we did in 182 and 183.
- Not much. I believe that you are doing an excellent job.
- I would be a little more familiar with the subject.
- Use some examples that aren’t in the book – this gives us more to look at since we can see what Budd does and what you do.
- No comment (2)

Additional comments:

- On the first few days, provide a *very quick* but *very thorough* review of C++. I had forgotten almost everything about C++ over the summer and had to look at old programs to remember how to do things.
- I like your subtle humor.
- The class has not been all too difficult so far. I have understood topics for the most part. The hardest part, as I have mentioned above, has been remembering code and C++.
- You’re doing a good job.
- Some examples which aren’t in the book would be good, but not all would need to be.
- I think you’re doing a good job. Especially for first year. The electronic submission of labs is nice. A bit more of an explanation of documentation would have been helpful at the beginning (possibly first lab).
- No comment. (14)

8 CS-285a Feedback in Week 7

At what moment were you most involved (excited, enthusiastic, ...)?

- I was excited when I got my test back and found I had done well.
- Generic algorithms/algorithm improvements.
- When my `list` program crashed NT, 95, Linux, and dumped core on Unix, all from the same code.
- Mostly in lab. In lab it’s more hands on, and you are actually accomplishing something, not that lecture doesn’t accomplish anything, but lab is much more exciting.
- When you took examples and put them on the board with the entire code, not just the function parts.
- When we were working on the span program, it made sense.

- Been pretty involved recently.
- I enjoyed the group work, it kept me awake, it made us think, and not just copy from the board. (3)
- Explanation of the different data structures.
- Talking about new container classes.
- Anytime you crack a joke.
- Writing the programs.
- When you almost went in on how to erase elements from a height balanced tree.
- Learning about the highly exciting `insert()` function of the `set` class.
- Group work, talking about new areas like `sets`, `multisets`.
- “Boy this is exciting[!]” . . . need I say more?
- When I finally understood the `list`.
- No response (1)

At what moment were you least involved (bored, disconnected, . . .)?

- Some of the lectures are very uninteresting.
- During the first day of the `list` explanation – next time was much better. (3)
- During the explanation of things that had already been covered in another class, such as strings, vectors, but *NOT* the `list` class.
- When we were going over something in the book that I had just read.
- Don’t know the stuff well enough to really get bored of it.
- Watching you write code on the board.
- Never really, there have been a few periods of semi-boredom, but never a “moment of most boredom.”
- Big oh time complexity.
- When I am in Physics class.
- While taking this survey.
- Big O notation and when we go over the same stuff constantly for one class period.
- I forget. . . I might have been asleep at the time.
- Budd’s `list` implementation. (3)
- No response (1)

What was the most helpful action taken by anyone?

- The most helpful thing for me is sharing ideas with other students.
- When you re-explained the `list`.

- Me deciding not to use Budd's code.
- Giving us handouts of the notes as opposed to just reading them to us.
- All of the code being put up on the board, so we have some idea of what it looks like.
- When someone helped me debug my `list` class.
- A friend, we shall refer to him as Mr. X, helped me to comprehend the template class thing.
- Giving an extra week to complete the `list` class. (2)
- Sending in e-mail and getting the answer set to everyone.
- The person who put the Simpsons on twice a day.
- I enjoy your sense of humor, it makes class more interesting.
- Really wasn't anything that anyone has done.
- When you taught us German and Russian.
- You correcting Budd's `list`.
- None (1)
- No response (3)

What was the most confusing action taken by anyone?

- Some lectures are hard to follow.
- Me 'originally' deciding to use Budd's code.
- Budd's implementation of the `list` class. (6)
- Again, trying to answer this question.
- I'm confused.
- By you: Referring everyone to possibly use "Budd's Code," as it is now known, instead of just starting from scratch.
- I don't know.
- There's nothing in particular except when I got confused when going over Budd's `list` class. (3)
- No response (3)

What most surprised you?

- How well I am doing in this class.
- How much there was to the `list` class and how little to interface classes.
- You actually listened to our comments from the last one of these. Most teachers don't.
- When I learned my true name is "void." I think I might legally change it to "VoidMan" and become a superhero or villain depending on what latitude I am on at the time. Or nothing.
- When my `list` class compiled. (2)

- How much fun this class is.
- When I woke up and realized that I really was in class in just my underwear. [*That must have been the day my twin brother did the lecture.*]
- That the `set` and `multiset` were implemented as height balanced trees instead of binary trees.
- That Budd's `list` implementation was so pitiful. (5)
- When I received the email saying our lab was not due.
- That you might be able to make a stack of a stack of a stack. . .
- No response (3)

What would be the first thing you would do differently if you were teaching the class?

- Get rid of Budd's book or at least his code. (3)
- Get people a little more involved in the discussion in class.
- Go over more examples.
- The `list` lab has been a mess, and while teaching us something, it has also been extremely frustrating. I personally would not use that lab again.
- Pay the students' tuition.
- We should have to write code for a function as a class every once in awhile.
- Have the students understand how the `list` class works, but not to waste time having them write an inferior version to that given in the STL.
- Keep giving handouts on lecture notes.
- Nothing. I think the class is being taught just fine. (3)
- I would go less over how to code the structures and more on just what methods are being used.
- I could only hope to be half as good a teacher as the illustrious Dr. Taylor.
- I think you're on the right track. You get the class to laugh and admit your mistakes.
- I would probably fall asleep. I give you a lot of credit for showing so much enthusiasm about the class.
- Save the trouble of writing on the board by giving slews of handouts.
- No response (2)

Additional comments:

- This class has been as enjoyable as my previous programming classes, and I think I have a better grasp of the material now.
- In case you haven't figured it out already, I'm not a big fan of the book. I learn better by 'doing' anyway.
- The class has been going quite well, and you are doing a great job.
- I think it is good to learn about different structures and how to use them, but I believe it would be more useful to continue

writing advanced structures instead of using the existing ones.

- Skip chapter 9 of the book. Teach entirely from notes. Your jokes and enthusiasm are what keep me awake. Keep them up! You're doing great.

9 CS-285b Feedback in Week 8

At what moment were you most involved (excited, enthusiastic, ...)?

- Working on the list class, there was a lot of interaction with other students.
- I was most involved when going through the set class.
- When we were given pages of new code we could try out.
- The beginning of lab while you explain the lab.
- When I get help on the area in the lab where I get stuck. The minute I get around the problem, my mind starts flowing with ideas again.
- When we were talking about the hash tables. (3)
- Getting my program to work properly in lab. (3)
- When working on labs. (2)
- Hashing and sets. Always thought I had to make my own tree class.
- When we broke into small groups to solve a problem.
- No comment (1)

At what moment were you least involved (bored, disconnected, ...)?

- Rewriting the spellcheck program for lab 5.
- I was least involved when going through the hash table.
- Sets. Could have done without. Handy things, but not really worth an entire week.
- After I left the lab.
- Lecture.
- When we cover new topics that I don't quite understand yet. (3)
- Some of the lectures during the lab session.
- When I couldn't get my program for lab 5 to work. (2)
- Taking these types of quizzes.

- I'm always interested and involved in CS285.
- I forget... long time ago.
- End of the period... just antsy to leave I guess.
- No comment (4)

What was the most helpful action taken by anyone?

- Assisting with the list class when I was stuck getting one final function working.
- Going through examples of each new class.
- When a friend helped me figure out a really strange error.
- Dr. Taylor taking the time to work with me through some tough C++ periods.
- When we reviewed Big O notation in class.
- When a classmate taught me a few concepts that I didn't know/understand well.
- Talking to someone about the way they approached the problem, and why.
- When the requirements for lab 4 were changed.
- Asking questions. (2)
- You telling me what's wrong with my program.
- When *censored* wrote his/her function on the board, and when *censored* and I worked together.
- No comment. (3)

What was the most confusing action taken by anyone?

- None. (4)
- Why Prof. Taylor didn't give us more of a direction to take with Lab 5. I wasted a lot of time trying different approaches and nothing worked.
- Budd's code in lab 4. (2)
- I still don't understand why the e-mail report format is such a necessity.
- Asking the wrong questions.
- No comment. (7)

What most surprised you?

- How hash tables work.
- How bad Budd's implementations are.
- That Prof. Taylor seems to know C++ pretty well after all.

- Using a set instead of skipping straight to hash tables.
- We really haven't learned all that many new things. We already knew the list, vector, array, stack, and queue.
- The big O notation at first seemed to do little, but it has helped.
- The difficult time I had with this class.
- That the list class lab's requirements were made easier.
- No big surprises, really.
- The test, but maybe it might be mainly because of the method I used to study.
- Having a quiz the class after a test.
- Nothing.
- How hard the labs are.
- The set thing.
- For having just learned C++ in September, you know it pretty well.
- No comment. (2)

What would be the first thing you would do differently if you were teaching the class?

- Give us 2 weeks to finish the second spellcheck lab.
- Nothing.
- Let the students have a full week to do the lab. Many of us do the bulk of the programming on the weekend and if it's due on Monday, we have no chance to ask questions. (2)
- Make the lab mandatory for the whole two hours. I'd get more done.
- Don't know, I don't believe I'm the type who could be a professor.
- Find a better textbook. (2)
- I think the reason why people are afraid to respond in class is because the way you ask the question. Instead of having the student *always* respond in sentences (which opens the room for mistakes), try making it more so that you seem to provide the sentence, and the student just fills in the blank. This way with you providing the majority of the sentence, it increases understanding, and when the student fills in the important *blank*, all the students know where the focus of your lecture is.
- Explain use of "includes," but that probably should have been done in CS182/3.
- Discuss how to do labs in class instead of just asking how the lab is going.
- Give candied apples every Friday? Kidding, Nothing. Great class.
- Nothing.
- No comment. (3)

10 CS-183 Feedback in Week 5

At what moment were you most involved (excited, enthusiastic, . . .) in class or lab?

- I was most involved during the group function writing in class. Small groups of 2 or 3 are effective learning tools.
- When a program finally worked or when I actually understood the material at that one exact time.
- The second day of lecture, because I understood everything that was discussed completely.
- When we started working with classes.
- In lab, implementing the classes (lab 2). (4)
- In class, learning the class structure. (3)
- I really enjoy the labs in general. It is exciting to be given a project to work on. A specific moment would be hard to define.
- I was the most excited when I got my first lab to work! And I'm still excited because I understand what [I] am doing in lab (at least the last two labs).
- Probably the moments in lab at about 9:30 when I've finally figured out how to do the lab and I'm finally waking up. It's just kind of nice to know where you're going and how you can solve the problem to get the correct answer.
- The moment that I finally understood how to use `const` properly in my programs.
- In class and lab. And I feel great when I leave the class with understanding all the materials which covered in the same day.
- The last lab (bbplayer) was one of the most fun. It is probably the only lab that helped me to understand what was talked about in class better and not just use what I know.

At what moment were you least involved (bored, disconnected, . . .) in class or lab?

- Reviewing material from last quarter in CS-182. (4)
- I was least involved when we talked about the theory of abstract data types. The whole lab on writing a class was losing me at first.
- When the material is difficult to understand and in the lab when nothing works, and [I] then become frustrated.
- The first lab I was really confused and thought this was going to be a really long quarter.
- I was disconnected this past Monday not because of class, but *extreme* lack of sleep.
- The first lab. I had problems with the vectors.
- The one day all semester that we used the book – and I forget it.
- For the most part, I have enjoyed coming to class and have not found it boring.
- I'm disconnected when I get lost in class sometimes. But when I go over the same material at home it's much better.
- When we first started talking about classes, I was just lost. Our book isn't very informative, so I wait for the class to

explain the book and then I look at the book if I need help. So, some of the overload functions and operators had me lost. The lab helped to clear that up by forcing me to understand.

- I'm not bored or disconnected, but sometimes you cover material so fast, therefore, I get lost a little bit. (It might be that the problem is me not you.)
- Towards the end of our discussion of classes the discussion seemed a little repetitive.
- No comment. (1)

What was the most helpful action taken by anyone in class or lab?

- When I was told what an enumerated class actually did.
- People asking questions and going through programs step by step on the board. (2)
- Explaining the theory of classes simpler, i.e., mutators, facilitators. . .
- When a student asks me if I understand it and offers to explain it to me. (2)
- Introducing and explaining a lot of the pre-defined functions we never learned about in CS-182, and the helpful lectures on review material from CS-182.
- The review sessions that were held on strings and vectors. (2)
- Taking time during lab to explain answers to questions in great detail is very helpful.
- When I consulted with [a student] about the design of my program (s)he gave me new insight which made for more elegant code.
- Just people helping to explain aspects of the lab.
- When you talked about rvalues and lvalues. That cleared up a lot of confusion.
- In the lab. What I got it from the class. I will use it in the lab like (class object).
- Making us work on our programs in lab and being there to help.
- No comment. (1)

What was the most confusing action taken by anyone in class or lab?

- Some of the lab directions (Dr. Sebern's) on the web.
- When everyone talks about C in class, I feel like they know it more in depth and I'm lost.
- Just the material.
- Asking questions which I had no clue what they were talking about.
- When I asked another student a question about the first lab, they tried to explain how they did it and confused me even more.
- When [a student] told me to use the *this command.
- Nothing comes to mind. (3)

- The fact that some people seem to know instantly how to do most of the labs makes you feel like you missed something.
- When I wasn't allowed to take the quiz and got a zero for being 10 minutes late on someone else's fault.
- I think the class object makes me confused because some of these materials don't make sense to me.
- When we were working on the less than function and someone said that we should switch the < to >.
- You told me to use the *this in a function that wasn't supposed to have it.
- No comment. (3)

What most surprised you?

- All of the different options that classes can have.
- How much we already learned this quarter.
- CS-183 is going to get a lot harder.
- Turing in labs through email. That's the first teacher who ever did that.
- That I'm actually being able to understand what we are learning in class whereas last semester I always seemed lost.
- Your willingness to help us. My teacher for CS-182 told us to read the book instead of trying to explain things to us.
- A 76 on the last lab for an output file error.
- Classes – they are useful and flexible.
- That we could overload basic functions like the =, > and < signs.
- What surprised me was the broad range of knowledge brought from CS-182 by different people – I thought all of the sections were to teach the same thing.
- How much C++ can be forgotten so quickly.
- I was surprised (and happy) that you actually write stuff on the board and then go over each thing carefully. That helped a lot. Better idea than transparencies.
- That this class was a lot harder than CS-182. In CS-182 we had everything handed out on a platter. Most of the programs also were easier because we dealt with easier programs that just had more parts.
- That we only talked about lists for a day.
- I feel the last two weeks ago I improved in C++ with Dr. Taylor.
- No comment. (1)

What would be the first thing you would do differently if you were teaching the class?

- No more quizzes in lab. (2)
- Put daily material into a smaller lab in addition to the big 2-week lab. Maybe have to write a very small function using notes. Similiar to quizzes, I guess.
- I think Taylor is doing a good job. Go over material for the lab again if necessary.

- Have a review session at the very beginning.
- I like the class the way it is with examples, explanations, and the labs having to do with material we learn in class.
- I would probably check to see how much students have accomplished during the labs to make sure they aren't getting too far behind.
- Have much larger programs to write and make us work in groups.
- The design of this class is pretty solid. I can't think of anything to change.
- It's all cool. I like the way the class is taught. I like that the instructor is available for help in lab, class, and office.
- Get a different book! Sure we'll get one or two examples on a subject but they'll never cover all the little extra details you need to actually get it to work.
- When covering new information, I would also talk about, "This would be useful if I ever had to do..." (i.e., incorporate design possibilities/usefulness more.)
- Because everybody agree about C++ is a difficult materials. I think that more exercise we do in the class the more understanding we get.
- Nothing. (3)
- No comment. (1)

Additional comments:

- The mandatory lab time is a nice change from last quarter.
- The class has helped me a lot. I was thinking of changing majors because I didn't know what I was doing at all. The help sessions were a very good thing and let me understand strings and vectors a lot better.
- I am enjoying CS-183 so much more than CS-182, and I feel that I am understanding the material a lot more too.
- If students are having such serious problems with the course material that they take up 1/4 to 1/3 of the professor's time during a lab, I feel that they should get tutoring outside of the class.
- I hope your tests are nice. I'm scared ? that open book that they might be really hard. I would think it generous if you gave us a little prep talk on Wednesday for what's on the test for Friday.
- Your personality helps to make the class more fun.
- No comment. (10)

11 CS-183 Feedback in Week 10

The table below is a summary of student responses to the following:

At the beginning of this course, a number of course objectives were identified. Please give your assessment of your own success in meeting each of these objectives and of the value of the course in helping you to meet them. Please use a scale of 1 (not successful at all) to 5 (very successful) for each rating. Note that this assessment is intended to be a process improvement measure, and will **not** be used in assigning a course grade.

Course objectives (CS-183)

	Assessment of your success in meeting this goal		Assessment of course as assisting you in this goal	
	μ	σ	μ	σ
On successful completion of this course, the student will:				
understand the rationale for object-oriented design and programming	4.23	0.70	4.46	0.63
understand data abstraction and abstract data types	4.46	0.75	4.54	0.63
be familiar with UML class notation and use cases	3.62	1.15	3.77	1.37
understand and be able to apply pointers and dynamic objects	4.08	0.92	4.38	0.74
understand and be able to apply STL iterators	4.31	0.72	4.54	0.50
be familiar with composition, inheritance, polymorphism, and C++ templates	4.46	0.63	4.54	0.50
be able to design and implement simple C++ classes and class libraries	4.62	0.62	4.85	0.36
be able to design small software systems using classification and use cases	3.92	0.83	4.00	0.78
be able to document the design and implementation of small software systems	4.69	0.61	4.31	0.91
be able to work in a team to create a small software system	4.46	0.63	4.00	1.24

Additional student comments in response to:

Please add any comments or suggestions you have regarding the course objectives, your experience in this course, and the role of the course in helping you to meet the objectives. Thank you.

- There was a lot of information we covered on pointers rather quickly, and we didn't implement them in lab. The rest was good, though.
- I didn't understand polymorphism that well. I think we could have spent another day on it.
- This course covered all of the objectives and gave me a better understanding of programming in general. I came out of CS-182 without a lot of understanding, but this course has clarified a lot of that.
- Learning to create GUI apps with MFC should be added to the objectives. It is a very useful skill, learning its use was my most valued experience.
- The objectives were discussed well, but how to use things in actual programming got confusing.
- Labs take up a lot of time. Any way we could make labs less time consuming?
- This class was well taught and I found it enjoyable.
- Something that suggests different methods for a team to work together would be beneficial for the group project.
- I look forward to having you for a professor in the future.

12 CS-280 Feedback in Week 6

At what moment were you most involved (excited, enthusiastic, ...) in class or lab?

- During the times when we get the class laughing.
- When I saw that I actually came close to passing the test.
- During the "Knight Rider" lab. I love flashing lights. (5)
- During a good, descriptive example.
- When the LED lit up in lab, and the first time the briefcase was set up.
- When you explained how to do the exam problem, and what you did made sense.
- Lab 1... when I got help from you. I felt I actually understood.
- Test.
- 4:50 on Friday when we got out of class or 4:54 on a previous day when we *finally* got out of class.
- First day! Based on our first conversation this class seemed to be quite interesting.
- In doing my exam, as the bit number addition was a challenge and it was just that we had less time during the exam.
- The first week of classes and the first day of lab.

- Doing the labs. (3)
- When we broke up into groups to work on actual code.
- When I changed the time delay to 50ms between LEDs in the Knight Rider lab.

At what moment were you least involved (bored, disconnected, . . .) in class or lab?

- During days when we study the same material the whole class period.
- When we discussed the actual solution to the test on Friday.
- The test.
- When trying to follow an example without completely understanding how one element worked.
- The first day during the review of number systems.
- Watching you write code that we had in front of you.
- After you explained the exam problem and I realized that what I did the second time wasn't much better than the first.
- About fourth week. . . I was lost and totally disconnected.
- Disassembling machine code.
- Second week of class. The book is very boring. The example in class and book are not as difficult as the test.
- In performing the first lab as it was all spoon fed.
- When I don't understand a concept in class.
- During lecture, before there were actual coded examples shown. Since then, lecture hasn't been that bad.
- Typing the first lab was boring, and disassembling the code for labs is boring.
- Class – because things aren't always explained and I sometimes confused or have too many questions.
- I was least involved during discussion of assembler directives in class.
- Taking this survey. (2)
- No comment. (2)

What was the most helpful action taken by anyone in class or lab?

- Xxxx keeping us entertained with his problems.
- I get a bit of help from a friend in class but not much!
- Explanation of the test. (2)
- I can't think of one.
- Not yet.
- When someone in class explained what I couldn't understand. Also when the prof. helped me and my partner on our first lab.
- You giving us the opportunity to redo the first exam. (2)

- Allowing 75% of the points back instead of 50% on the test. (6)
- Convincing teacher for more points on make-up test.
- The most helpful action taken by anyone was asking for more example code in lecture.
- No comment. (2)

What was the most confusing action taken by anyone in class or lab?

- None.
- Describing how to use the keypad and ports. I know more about it, but I still don't quite understand the port usage.
- There were a couple of descriptions or answers you gave that were somewhat hard to follow.
- The three explanations I received about accessing the keypad.
- I have no recollection.
- I'm generally confused by the whole class, not just one person's comment or action.
- When doing the group thing.
- Taking the test.
- Dr. Taylor expecting us to be able to do this two week lab without more help.
- The explanation for the op codes. It seemed we were getting half of the information we needed.
- The explanation for the keypad. Still confusing. It would be nice if explained again in a better way.
- Asking questions that are not related to the subject discussed in class.
- The explanation of how each line of code effects the CCR. Mainly the half-carry, mask interrupt, stop enable, and... I forget the other one. The last four are okay.
- How the keypad works is quite confusing.
- Explanation of the keypad was confusing at first.
- The most confusing action (if you can call it an action) was the discussion of whether or not to use a #. I still don't completely understand when to use a #.
- I don't know. This question always confuses me.
- No comment. (3)

What most surprised you?

- Almost passing the test. (I thought I did pretty badly.)
- How you actually use the keypad.
- The difficulty of the first test. (11)
- Nothing surprises me, I am one with my surroundings.
- The test. Even after explaining the problem, I still don't fully understand!

- Nothing yet.
- The test was a big surprise to me. The last part wasn't clear enough for me (directions).
- Being able to understand some of the material.
- When I walked in on my roommate and saw him constructing full scale models of Disney world.
- No comment. (1)

What would be the first thing you would do differently if you were teaching the class?

- More handouts, descriptive handouts. More full program examples, whether they are simple or not doesn't matter, but the way they provide visual examples for everything helps.
- Make the test a little easier.
- Slightly easier test, and more labs based loosely on action/adventure shows of the 80's, such as "Knight Rider."
- Get a different book. The different syntax can be confusing.
- Do more of the examples in the same fashion you took to explain the test question. (4)
- Nothing.
- Explain in more detail how the data is stored in the registers. I know what the opcodes are supposed to do, but I don't know how.
- After talking with students from other classes we are also learning what they are. Just a little better explanation and helpful tips of programming will help in lab and future.
- I would try to give extra credit if necessary. Like for example, short programs or part of a program, but only if the student wants to do it.
- Taken the class prior to teaching the class.
- Estimate the students' abilities better when writing tests, and think about time constraints more.
- Just go through all main opcodes and relate them to what is done in C++.
- Explain more the procedures of a program using pictures and diagrams. Also, if you think something is important enough to be on a test, please stress it.
- Have more short hand-in assignments and fewer tests/quizzes.
- I think giving back 75% of the points on the test was a bit too much.
- I would have given more explanation of the input/output ports before we talked about them and were expected to use them.

Additional comments:

- I like the free style with which you are teaching. You don't discourage questions, comments, suggestions and aren't afraid to joke around a little.

- I wish you could do more visual displays as in to see what actually is going on behind the screen. I believe this would help me and my classmates understand the commands and why those directions must be done.
- Liked that you went over the test and quiz in class.
- The example of checking for a pressed key was good.
- More time should be spent on the different addressing mode syntax.
- The test really wasn't all that long. It could have easily been done in less than an hour outside of class. I think the problem was that we weren't experienced enough with assembly to write an entire program from scratch in a timed situation. Also, the condition code register problem was a bit time consuming.
- The most confusing thing that happened this quarter was when we introduced portb. When utilizing port b in a source file, examples showed it as a symbol set to 1004. At the time, we hadn't talked about, or used symbols for a long time. So, the first use of a symbol was for the definition of portb. This was confusing because you didn't mention that it was simply a symbol called "portb" which referred to the address for port b. I had even forgotten about symbols entirely by the time we used it. My initial reaction was that we were defining where port b was located for the assembler, though it seemed unnecessary.
- No comment. (13)

13 CS-280 Feedback in Week 10

The table below is a summary of student responses to the following:

At the beginning of this course, a number of course objectives were identified. Please give your assessment of your own success in meeting each of these objectives and of the value of the course in helping you to meet them. Please use a scale of 1 (not successful at all) to 5 (very successful) for each rating. Note that this assessment is intended to be a process improvement measure, and will **not** be used in assigning a course grade.

Course objectives (CS-280)

	Assessment of your success in meeting this goal		Assessment of course as assisting you in this goal	
	μ	σ	μ	σ
On successful completion of this course, the student will:				
Understand the proper uses for assembly language programming	3.94	0.77	3.88	0.89
Utilize a modular approach to assembly language programming with code reuse	2.75	1.00	2.69	0.79
Be able to write easy to understand, yet concise assembly language programs	3.63	0.89	3.75	0.86
Be able to use embedded systems development tools	3.44	0.73	3.75	0.77
Be able to use various addressing modes and understand how memory addressing is accomplished	4.13	0.96	4.06	1.06
Be able to understand and use hardware interrupts	3.56	0.89	3.75	1.00
Be able to integrate assembly language subroutines into a high level language program	3.31	0.87	3.38	1.02

Additional student comments in response to:

Please add any comments or suggestions you have regarding the course objectives, your experience in this course, and the role of the course in helping you to meet the objectives. Thank you.

- For modular approach, it should be made a compulsion to write subroutines in different files so that we get habituated to that. Need more assembly/C integration experience.
- Need more reference materials – learner’s book, help files, web postings.
- I learned far more from painful hours of figuring out how to do stuff in lab than I could ever learn in class.
- I learned far more in lab than in lecture. The course textbook was *worthless* and written in a different language than we were using. The tests should have been take-home.
- Examples in class didn’t use a modular approach (at least not that I can remember). Linker files were omitted in most examples, which made it hard to tell which approach was being taken. Personally I fell this (as well as most others) class would be easier to understand/follow in PowerPoint format, as more time could be spent understanding concepts and less time taking notes.
- A better book would be beneficial.
- Have tech support fully test all protoboards and briefcases because there are lots of problems. More computers in 307 will help at the beginning of the quarter. *A new book!*
- Some concepts in the course seemed only half explained. . . was a little confusing.
- To me, the material covered in lecture wasn’t presented in an easy to follow way. The topics didn’t seem to connect and in general was very difficult to understand. One point that was very lacking, I feel, is the objective of “writing easy to understand, yet concise programs.” It didn’t seem that we fully covered design techniques for coming up with good, concise code. Especially, when it comes to the tests. To come up with concise code quickly was an area in which I struggled.