# Student-Centered Learning
# in the
# Introductory Computer Course

**Thomas Gibbons**
**CS/CIS Department**
**College of St. Scholastica**
**Duluth, MN**
tgibbons@css.edu

## Abstract

The introductory computer course poses a number of challenges, including students with varied experience, technology that changes faster than textbooks, and students with different expectations. This paper looks at three student-centered learning techniques and how they can be used to overcome these obstacles. Classroom experiences with the strengths and weaknesses of these techniques are covered.

## History of our Introductory Computer Course

For many years the CS/CIS Department at the College of St. Scholastica has taught an introductory computer course open to all students. While this course was an elective course for many students, it was a required course for students in a number of majors including our CS/CIS major.

Originally the course introduced students to general computer concepts. This included an overview of computer hardware and software. Students also learned the basics of spreadsheet and database software. The course faced a number of challenges, the largest of which was that students were often bored by the computer concepts covered. Those students with a strong interest in computers had covered these concepts in high school courses. Those students with limited interest in computers were likewise often not interested in these computer concepts. Neither the CS/CIS department nor the other departments requiring the course were pleased with the outcomes.

About six years ago, the course was broken into two separate quarter courses which both concentrated on office applications and problem solving. One course concentrated on spreadsheets, while the other focused on desktop databases. Both courses also spent some time on other computer literacy applications and provided a brief introduction to programming. For a time these two courses produced management majors with significant data analysis experience with spreadsheets, and medical records majors experience with databases. For many years, we continued to favor problem solving with applications over computer literacy.

Over time we noticed that some CS/CIS juniors did not know what “TCP/IP” or “CD-R” were. A number of our students seemed to have blind spots in their education from the lack of a computer literacy course. As our university moved from quarters to semesters in the summer of 1999, the CS/CIS Department was given a chance to combine the two introductory computer courses and revise the topics taught in them.

At about the same time, the College of St. Scholastica was awarded a Bush Foundation grant for creating a student-centered learning environment at the college. Part of the funding from this grant went towards redesigning courses in a more student-centered way. The introductory computer course was one of the pilot courses that received funding from the Bush foundation.

## Challenges Facing a Computer Literacy Course

A wide variety of students take our introductory computer course. The students’ prior computer course work varies from none to over three years of high school courses. The reason students have for taking our introductory course varied from being an interesting elective to a requirement of the major. Similarly, the students’ interest in the course varied from devotion to apathy. The result was a classroom where there could be a Nursing major (who thought computers were a fun tool) sitting next to a Management major (who thought computers were stupid). Sitting near these two could be a computer major who thought she knew everything on the syllabus by the first day of class.

The speed at which technology changes only adds to the challenges faced in this course. While a student may have an extensive computer background in high school, a significant percentage of that material will be out of date by the time the student is in college and new concepts and methods will need to be appended to the material from high school. Also, much of the material covered in the college course will be out of date by the time the student graduates. This makes it difficult to convince the student who does not see the usefulness of the course that the course is teaching deep underlying principles and methods, which will outlive the current technology.

## Student-Centered Learning

The goals of student-centered learning primarily focus on making students more active participants in the learning process. This includes allowing students to learn at their own pace and considering the different learning styles of the students. Learning should become more individualized. This generally involves relying less on lecture as a learning tool as the instructor becomes more of a mentor or facilitator. The question is, what methods can be used as a replacement for lectures? Student-centered learning often utilizes open-ended research and student collaboration.

## Methods Used

The introductory computer course at the College of St. Scholastica uses three different methods grounded in student-centered learning. First, students choose between a basic and an advanced version of the course. Second, students master material by working in teams to solve problems. Third, students select topics to research and then present to their peers the information they have found.

### Advanced Option

While it is not possible to offer different versions of the introductory computer course to fit the exact educational needs of each student, it is possible to offer a couple different options. At the College of St. Scholastica, we have chosen to offer two different versions of the course, a basic and an advanced version. The basic version is a four-credit course that covers computer concepts, problem solving with spreadsheets and databases, web page authoring, and PowerPoint presentation. The advanced version is also a four-credit course, which covers the same material plus a short introduction to programming using Visual Basic. To allow time for this additional material, the advanced course moves at a more rapid pace than the basic course. While computer majors and minors are required to take the advanced course, other students can choose either the basic or the advanced course.

Students who originally take the basic course and find later that they would like to complete the computer minor or major can take a one-credit introduction to programming course. Thus the basic courses and the one credit course are equivalent to the advanced course.

One of the goals of this approach is to lessen the range of student experience and expectation in these courses. Students taking the advanced course are advised that it moves at a rapid pace and prior computer experience will help them in completing the course. The advanced course attracts students with a strong interest in computers and high expectations for the course. On the other hand, many of the students in the basic course have limited computer experience. By dividing the students into these two basic groups, the hope is that an instructor can better address the needs of the students.

## Team Problem Solving

Today, spreadsheets and databases continue to be important applications used in diverse careers. Spreadsheets and databases like Microsoft Excel and Microsoft Access are commonly taught in high school. Yet high school courses often only cover the basics of these applications and do not develop problem solving skills using these applications.

Rather than lecturing extensively on these applications, the approach used in the introductory computer course is to mix short lectures with team problem solving assignments. The lectures highlight advanced features of the applications, but they do not cover all the facets of the application. In team assignments, students are expected to learn additional features of the application on their own using the textbook, online help, and trial-and-error.

These weekly team problem solving assignments are done in class. Students choose their own teams of two or three people. They are given significant problems to solve. The instructor refrains from providing too much assistance. When team members struggle on a problem, they often develop creative methods of solving the problem using the application, and develop their abilities to learn independently. Problems tend to be open ended without one clear answer. Besides turning in a solution, each team much also provide a write-up describing the difficulties the team encountered and how these difficulties were resolved.

Before this method was implemented, the expectation was that mixed teams of advanced and beginning students would form. Beginning students would then learn new techniques from advanced students. While mixed teams are chosen at times, experience has shown that students prefer working with other students of similar ability. Advanced students often prefer the company of advanced students because they feel they can solve the problem faster and do not have one team member “coasting” and not doing a fair share of the work. Similarly, students lacking in computer experience are more comfortable working together. They are much more comfortable learning together with their peers as they struggle with the problem. It is also interesting to note that the amount of past computer experience is not a good predictor of the sophistication of the teams’ solutions to the problems.

In line with the principles of student-centered learning, team problem solving allows students to choose the method for learning the material. A student might use trial-and-error to discover the correct technique or the student might turn to her/his textbook or the examples given in the online help. Other students might be more comfortable waiting for a teammate to discover a particular technique and simply noting the solution found in the same way that the student takes notes from a lecture.

Samples of problem-solving assignments can be found on the web site for this course, http://www.css.edu/users/tgibbons/cis1005 (click on "Labs" from the class home page).

**In-depth Self-study**

Most introductory computer concepts textbooks have a chapter on computer hardware. The challenge facing educators is how to cover this material in the classroom. Often the text has outdated material—even if the text was only published in the previous year. Also, some of the students have covered much of the material in pre-college courses. Seeing the input-CPU-storage-output diagram yet again will not provide a quality learning event for these students. The same holds true for most of the other chapters in today's computer concepts textbooks.

Since student-centered learning gives students a part in deciding what material they want to learn, students and the instructor can work together to decide what they need to learn. The approach taken in the introductory computer course is to have the instructor select a set of topics related to the current chapter and let each student choose one of these topics to study in-depth. The student relies extensively on the internet to research that topic. The student creates a web page with information about the topic to share with the rest of the class. To help the students start their research, the instructor provides web links to three or four resources for each topic. Students also review the in-depth reports their classmates have written.

With the students doing their in-depths and reviewing their peer's work, lectures on the material are limited to 15 minutes. The instructor also creates a list of all the terms and concepts from the textbook and the in-depth reports, which students are responsible for knowing at exam time. If there is a term or concept not covered in class, the students are expected to learn it on their own.

This approach allows the students some control over what they want to learn. They also choose, to some extent, what the rest of the class learns and is tested on. This in general produces positive results, but it also means that the instructor is no longer in complete control of what is covered in the course. It is possible for the instructor to select a topic which he/she thinks is important only to have none of the students select that topic for in-depth work.

This approach also raises the question of the quality of education. In a perfect world, students all do quality research and create well-organized web pages for presenting the information to the class. In reality, some students do superficial research and create confusing presentations. This causes two distinct problems. First, the students who do the superficial research will not be presented with the material at the same depth that a lecture might. Yet when you give students more control over their learning, some students will choose to do less rather than more. That is their choice and so not as significant as the second problem, which is that students interested in learning may be disappointed by their peers' work. What argument can be made to a student who reminds you of the significant tuition he/she is paying to learn material presented in a confusing manner from a fellow student? Still, no learning method is perfect; the best we can do is to strive for quality. Many students have reminded me how the lectures of some trained professionals seem to lack the same quality as the poorly done student presentations.

The final concern with this method deals with preparation time. Few instructors have extra time to spend teaching a course. At first glance, student-centered learning may seem to lessen course preparation time. If the instructor only lectures for 15 minutes of a 60 minute class, then lecture preparation time is much less. Unfortunately, it often takes more time to prepare for student-centered activities than it does to prepare for a lecture. Some of the preparation time could be reduced by not providing students with a list of topics from each chapter and initial research links for these topics. In the future, web-enabled textbooks may do this adequately. It always comes down to a choice as to the best use of the instructor's time in delivering a course.

Samples of in-depth topics and web pages created by students can be found on the web site for this course, http://www.css.edu/users/tgibbons/cis1005.

## Conclusion

The introductory computer course poses a number of challenges, including students with varied experience and different expectations. This paper looked at three student-centered learning methods and how they can be used to overcome these obstacles. The first method was to offer a basic and an advanced version of the course and let students choose what version they wanted. The second method was the use of team problem-solving assignments. The third method had each student share research on a topic the student and instructor choose together.

While student-centered learning is not without its difficulties, it does seem to address many of the challenges facing the introductory computer course.

## References

1. Buckhartd, R. M. (1994). *The Inquiry Process: Student-Centered Learning*. Iowa: Perfection Learning Corp.

2. Knowlton, D. S. (2000). A Theoretical Framework for the Online Classroom: A Defence and Delineation of a Student-Centered Pedagogy. *Principles of Effective Teaching in the Online Classroom*, *84*. Calif: Jossey-Bass

## Acknowledgements

I would like to acknowledge the Bush Foundation for funding of the "Creating a Student-Centered Learning Environment at The College of St. Scholastica" program which included funding for this course development.