

# The Internet as a Motivating Theme in a Math/Computer Core Course for Nonmajors

Chaya Gurwitz

Department of Computer and Information Science

2900 Bedford Avenue

Brooklyn, NY 11210

gurwitz@sci.brooklyn.cuny.edu

## Abstract

The challenge of a computer literacy course is to present general principles of computer science, while at the same time engaging students who are nonmajors and demonstrating the relevance of the subject. In this paper we describe our experience with a course that uses the Internet as a motivating theme to introduce students to fundamental concepts of computer science. The students found the course interesting and were visibly stimulated by the material. In comparison with previous semesters, the students left the course with more practical skills as well as a greater appreciation of the interrelations between mathematics and computer science.

## Introduction

Most Computer Science departments offer an introductory course geared toward students who are not planning to major in computer science. This course is generally known as a "computer literacy" course, although there is no consensus on what "computer literacy" encompasses ([4],[6]). In recent years the trend has been to emphasize principles and concepts rather than to concentrate on specific languages or applications. Many principles-based courses have been suggested, all of which are designed to be appropriate for students of diverse background (for example, [1], [2], [3], [5], [11], [13]). These courses aim to present concepts and long-lasting skills, and at the same time stimulate and interest students who are not planning to major in Computer Science.

Our situation is unique, in that our introductory course for nonmajors is a combined Math/Computer Science course. Core Studies 5 is a 3-hour, 3-credit course entitled *Introduction to Mathematical Reasoning and Computer Programming*. It is offered jointly by the Department of Mathematics and the Department of Computer and Information Science and is aimed at non-science majors. This course is required of all undergraduate students, with

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

SIGSCE 98 Atlanta GA USA

Copyright 1998 0-89791-994-7/98/2..\$5.00

the exception of those who take more advanced courses in both mathematics *and* computer science. Core Studies 5 is part of Brooklyn College's nationally recognized Core Curriculum [7]. Core courses are designed for nonspecialists and are suitable for nonmajors, but each is meant to introduce material of fundamental and lasting significance.

The unique challenge of this course is to present fundamental principles of both mathematics and computer science, at a level suitable for nonspecialists, and yet to achieve an integrated and interesting course. In the past, the course ([2],[3]) combined mathematics and computer science by focusing on propositional logic, digital circuit design and programming. In recent years the course has become outdated since logic is now included in the New York State high school curriculum. Additionally, students felt disappointed that the course did not teach them what they felt would be useful in the "real world". Furthermore, although the faculty understood the importance of logic in computer science (à la [15]) the students felt that there was a split between the "math", i.e. logic, and the "CS", i.e. programming, portions of the course.

During the Spring 1997 semester, we introduced an experimental version of Core Studies 5, which was taught in three sections of the course. In designing the course, we were motivated by a desire to revitalize the course and to update the subject matter taught in the course. We looked to focus on recent advances in Computer Science and to include topics that are not currently part of the New York State high school curriculum. Furthermore, the topics were chosen to show connections between Mathematics and Computer Science. Above all, we hoped to provide an interesting and engaging introduction to both Mathematics and Computer Science, and to communicate the vitality and relevance of these fields to nonmajors.

## The Internet as a Motivating Theme

We decided to capitalize on the high visibility of the Internet, by using the study of the Internet to introduce key computer science concepts. Familiarity with and understanding the nature of the Internet will be beneficial

to all students, regardless of whether or not they pursue a major in Computer Science. We felt that we could use the Web to introduce network concepts and teach HTML as a gentle introduction to programming. Furthermore, we felt that topics such as digital communication, parity checks and cryptography would provide natural ties to number theory.

The course was divided into three parts, each comprising roughly one-third of the semester. The topics - the Internet, number theory, and computer programming - were integrated by emphasizing the interrelations between them.

The first component of the course was the Internet. In this section of the course we discussed the development and nature of the Internet, as well as introduced students to the use of email and the World Wide Web. Among the topics presented were analog and digital communication, modems, binary coding, parity checking, networks, communication protocols and security issues. The students learned to use email, Web browsers, and search engines. We taught the students HTML syntax and had them design their own home pages, which were posted on the Core 5 Web site.

During the second part of the course we focused on number theory, and its relevance to the topics that had previously been presented. We introduced historical number systems and number systems using non-decimal number bases, and taught the students to do conversions between number bases. We emphasized the importance of binary and hexadecimal number systems in computer systems. We explored identification number and bar code schemes, concentrating on error detection methods. We discussed ASCII and binary codes, parity checks, data compression methods, and their applications to storing data in computers. We presented prime numbers and modular arithmetic, and related those concepts to public key cryptography and security over the Internet.

The final component of the course was Pascal programming. In this section of the course the students learned how to write simple programs. They were able to write programs to verify check digits for various coding schemes, implementing algorithms presented in the previous part of the course. Among the concepts presented were variables, assignment statements, for and while loops, Boolean conditions, and if-then-else statements.

### Course Text

Since the course covered several different topics, it was impossible to find one - or even two - textbooks for the class. Instead, we used a Course Pak prepared by the copy center on campus. The Course Pak service allowed us to

custom-design a text for the class by selecting chapters from several textbooks and incorporating our own notes. We included handouts developed by the College's Academic Computer Center as references for learning to use Windows, Notepad and HTML. To present material about the nature and development of the Internet, we used several chapters from [9]. Selections from the math textbooks [8] and [14] were used in the number theory portion of the course. Chapters from the Pascal text [12] as well as our own handouts were used during the third part of the course. Tailoring a text in this manner involves extra work on the part of the instructor, but results in a text that can be relied on throughout the semester.

### Course Web Site

The course was supported by an extensive Web site. The Web site provided access to general course information, such as the syllabus, course requirements, and office hours. Homeworks and exam reviews were posted online. Taking advantage of the wealth of material available on the Web, we compiled a bibliography of online resources, which the students could access with a click of the mouse. These supplementary materials provided information beyond that included in textbooks. (The references range from an Internet timeline, to a description of the 1996 discovery of the 35th known Mersenne prime, to a help page for Turbo Pascal.)

Aside from serving as a repository of course information, the Web site was also used to provide interactive course tools and self-quizzes. The tools include explanations of conversions between number bases, implementations of check digit schemes, use of identification codes to access on-line data bases, and simulations of Pascal programs. Students were able to use these tools as a supplement to classroom instruction.

The files on the Core 5 Web site can be accessed at any time from any Web browser. The URL for the course Web site is

*<http://www.acc6.its.brooklyn.cuny.edu/~gurwitz/core5new.html>*

### Means of Evaluation

The students were polled by means of a detailed survey, which was distributed at the final. A feedback form was provided on the Core 5 Web site to allow students the opportunity to submit anonymous comments.

### Results: The Good ...

Both students and the faculty teaching the course judged it to be a success. The students found the material interesting and relevant. The faculty enjoyed teaching motivated students.

- Student response to the course is extremely encouraging. An impressive 91% responded that they felt that they had gained something useful from the course. We were gratified that 80% found that the course was better than they had expected. Student comments were overwhelmingly positive. One student remarked, "If I could, I would take it again."
- The responses to the survey indicate that the course succeeded in showing interrelations between Mathematics and Computer Science.
- Students responded that they had learned new things about the Internet (81%), and about computer programming (91%) that they didn't know before.
- The students felt that they had been taught worthwhile skills, and that the material they had been exposed to was relevant to the "real world".
- All of the student respondents had used the Core 5 Web site to some degree. Almost all the students responded that they had used the Web site for accessing homeworks and exam reviews. Between one-half to two-thirds of the students noted using the Web site for notes and practice tools. Those that had used the notes and tools found them helpful.
- Most students (78%) did not feel that there was any topic of the course that they already knew from high school. Although several students responded that they had learned about prime numbers in high school, we feel that our presentation of prime numbers, with their application to cryptography and network security, provided a new perspective.
- As instructors, we found the students enthusiastic and motivated. Students participated actively in animated class discussions. Homeworks were completed and turned in on time. Many students went beyond the requirements of the course, explored supplementary material on their own, and submitted extra credit assignments.
- Introducing HTML early in the course provided a foundation for teaching programming. Students were familiar with concepts of creating, editing and saving files. They understood the difference between source and output. They had been disciplined to conform to the requirements of HTML syntax. For the most part, the students learned HTML eagerly, motivated by the goal of producing their own home pages. This pleasant introduction prepared students for the more rigorous requirements of writing complete computer programs.
- More students successfully learned programming in these sections than in any previous term. This appears to be a direct result of their more constant focus on the computer during the semester. They were able to become comfortable with using the computer prior to attempting to write programs. In addition, their success with HTML gave them practice in meeting the exacting standards of a compiler, necessary to programming successfully.
- We used computer classrooms periodically over the course of the semester to provide hands-on instruction. At the beginning of the semester we used these facilities to introduce the students to Windows, Notepad and Netscape. At the end of the semester we used the computer classrooms to teach Pascal. At other times, when the computers were not needed for hands-on instruction, we found it more convenient to lecture in a traditional classroom. The ability to alternate between classrooms as needed was an invaluable resource.

#### ... the Bad and the Ugly

While we judge the course to be a success, there were several shortcomings:

- The Mathematics component of the course was not as strong as the Computer Science component. The majority of the respondents to the survey (54%) felt that they had not learned something about mathematics that they didn't know before. This is somewhat inconsistent, since 78% responded that there was no topic that they already had learned in high school, and many cited math topics as being the hardest part of the course.
- Student email access was not convenient. At the beginning of the semester, the students were directed to one lab for email and to another lab for Web access. This situation was later remedied so that email could be accessed from both computer labs. Students who accessed the Web from the library or from home could not retrieve their email.
- Intermittent Web access was very frustrating. The week we assigned a large homework requiring searching the Web, there was no access all week. The day we chose to demonstrate Netscape in the classroom, the server went down.
- Although we appreciated using computer classrooms for hands-on instruction, we found it difficult to lecture in those rooms. The students could not see the

boards and found the television screens, meant for projection, uncomfortable to view.

### Changes for the Future

There are several areas in which the course could be improved:

- The students need more and better access to the Internet - both for Web browsing and for email. They must be able to access a browser and email from the same location. The College's computer center has made plans to address these problems for the Fall 1997 semester. The challenge of meeting ever-increasing student need for Internet access is likely to remain an issue here and on many other campuses.
- We alternated between lecturing in a traditional classroom, using chalk and blackboard, and using a full computer classroom in which each student sat at his own workstation. The traditional classroom is convenient for standard lectures, but not suited for demonstrating the use of Windows or Web browsers. Computer classrooms are wonderful for online instruction, but not convenient for lecture purposes. We propose to upgrade the traditional classrooms by installing a workstation, networked to the campus LAN and running Web browsing software, as well as an overhead projector with a pull-down screen. With this setup, we could include short online demonstrations while remaining in the classroom and retaining use of the blackboard.
- We are considering teaching the programming section of the course using JavaScript in place of Pascal. JavaScript is a programming language used to embed programs in Web pages [10]. This would have the advantage of integrating the entire course. The programming portion of the course would directly build on the Internet section of the course. Since the course tools for the number theory portion of the course were coded in JavaScript, students would see a clear connection between the Web, programming, and number theory.

### References

- [1] Allen, J. Thomas, Hayden Porter, T. Ray Nanney, and Ken Abernethy, "Reexamining the Introductory Computer Science Course in Liberal Arts Institutions", *Proceedings of the Twenty-first SIGCSE Technical Symposium on Computer Science Education*, February, 1990.
- [2] Arnow, David. M., "The Iliad and the WHILE Loop", *Proceedings of the Twenty-second SIGCSE Technical Symposium on Computer Science Education*, March, 1991.
- [3] Arnow, David M., "Teaching Programming to Liberal Arts Students: Using Loop Invariants", *Proceedings of the Twenty-fifth SIGCSE Technical Symposium on Computer Science Education*, March, 1994.
- [4] Arnow, David M., Moderator, and Owen Astrachan, James Kiper, Robert Workman, and Paula Whitlock, "Themes and Tapestries: A Diversity of Approaches to Computer Science for Liberal Arts Students", (Panel), *Proceedings of the Twenty-fifth SIGCSE Technical Symposium on Computer Science Education*, March, 1994.
- [5] Bierman, Alan W., "An Overview Course in Academic Computer Science: A New Approach for teaching Nonmajors", *Proceedings of the Twenty-first SIGCSE Technical Symposium on Computer Science Education*, February, 1990.
- [6] Budd, Barry, Moderator, and J. Glenn Brookshear, Rick Decker, Frances Gustavson, Mildred Lintner and Greg Scragg, "Revitalizing the Computer Science Course for Non-majors", (Panel) *Proceedings of the Twenty-sixth SIGCSE Technical Symposium on Computer Science Education*, March, 1995.
- [7] Cheney, Lynne V., *50 Hours - A Core Curriculum for College Students*. National Endowment for the Humanities, October 1989.
- [8] COMAP, *For All Practical Purposes: Introduction to Contemporary Mathematics*, W.H. Freeman, 1994.
- [9] Comer, Douglas E., *The Internet: Everything you need to know about computer networking and how the Internet works*, Prentice-Hall, 1994.
- [10] Flanagan, David, *JavaScript: The Definitive Guide*, 2nd edition, O'Reilly and Associates, 1997.
- [11] Goldweber, Michael, John Barr, and Chuck Leska, "A New Perspective on Teaching Computer Literacy", *Proceedings of the Twenty-fifth SIGCSE Technical Symposium on Computer Science Education*, March, 1994.
- [12] Jones, Jacqueline and Keith Harrow, *Problem Solving Using Turbo Pascal*, Prentice-Hall, 1986.
- [13] King, L.A. Smith and John Barr, "Computer Science for the Artist", *Proceedings of the Twenty-eighth*

*SIGCSE Technical Symposium on Computer Science Education*, February, 1997.

[14] Miller, Charles D. Vern E. Heeren and E. John Hornsby, *Mathematical Ideas*, Harper Collins, 1994.

[15] Myers, J. Paul, Jr., "The Central Role of Mathematical Logic in Computer Science", *Proceedings of the Twenty-first SIGCSE Technical Symposium on Computer Science Education*, February, 1990.