# DESIGNING AND RUNNING A PRE-COLLEGE COMPUTING COURSE

*Mihaela Sabin, Bryan Higgs, Vladimir Riabov, and Alberto Moreira*
*Department of Mathematics and Computer Science*
*Rivier College*
*Nashua, NH, 03060*
*(603) 897-8571*
*{msabin, bhiggs, vriabov}@rivier.edu, amoreira@ieee.org*

**ABSTRACT**

Enrollments in undergraduate computer science programs have declined steadily over the past few years. Overall, high school computer science curriculum is limited to either basic usage of Microsoft office tools or some advanced placement programming. Student expectations of undergraduate computer science courses for non-majors as well as the scope of computer science literacy in general are moving targets. All these problems have a negative impact on undergraduate recruitment and on promoting computing sciences courses and programs. In this paper we present our experience with a pre-college course in computing for high-school students offered at our College in the summer of 2004. We have learned promising lessons from running this program. These lessons can guide the curriculum design of computing courses for non-majors and motivate prospective students to pursue computer science programs.

## 1 MOTIVATION

The decline in undergraduate computer science enrollment [1] is a worrisome phenomenon that puts at risk especially small computer science programs in liberal arts colleges. Federal funding for basic research in physical and computing sciences does not show positive signs and will likely decrease by 1 to 2 percent this coming year [2]. Computer science courses in high schools seem to have a second-class status among the other science courses: biology, physics, and chemistry, and rarely count towards satisfying the science requirements in high school curricula [3, 4]. High school computing courses are sharply polarized between two curricular approaches: (1) basic training in using Microsoft office applications, and (2) strong emphasis on programming in the computer science AP exams. Computer literacy and computing science courses for non-majors in colleges tend to follow a similar bipolar pattern, and are either intellectually unattractive or unwelcoming to those who perceive programming as a "solitary activity practiced by so-called 'geeks'" [2].

All these factors raise obstacles in encouraging students to pursue computer science majors, minor in computer science, or simply choose computer science electives in their program of studies. Our own experience with recruiting, teaching, and advising computer science majors and minors at our College confirms these observations. The problem is compounded by another phenomenon we observed on our campus: students rarely switch to computer science, while the opposite is more common. We are convinced that the early start in studying computer science belongs in high school for those who might associate their professional future with computer science. We also believe that the computing science discipline has a more compelling "story" to tell high school students than AP programming or Microsoft Excel applications.

The rest of the paper is organized as follows. In section 2 we present the planning and development process that led to the actual program. The core of the paper is section 3, in which we describe what running the program entailed. We present the logistics of coordinating with other offices on campus; the student application and acceptance process, orientation, and daily schedule; the course and its curricular modules; the field trips and the social activities. The paper concludes with Lessons Learned, Conclusions, and Acknowledgments sections.

## 2 THE VISION

Strongly motivated by these challenges, at the beginning of the 2003-2004 academic year, the Office of Undergraduate Admissions and the Department of Mathematics and Computer Science at our College decided to develop a pre-college summer course in computer science for high school sophomores and juniors. The course was scheduled for the summer of 2004. The goals of the course were to:

- Increase the visibility of our computer science program within the community,
- Improve the program's recruitment,
- Pilot a hands-on course that presents various knowledge areas of computing, not only programming, per se,
- Assess high school students' interests, engagement with, and response to a course whose content and format differ from what is typically taught in high schools.

The pre-college summer course was configured to run for six days, with six and a half hours of class time per day, which were divided in two sessions, in the morning and the afternoon. The course was designed for a maximum of twenty residential students. The class size was limited by the number of seats in the computer lab. We allowed for commuting students to enroll in the program too. The academic and extra-curricular program activities were the same for residential and commuting students, except for the night lab study session scheduled after the departure of commuting students, typically around 8 PM.

The residential nature of the course was meant to give students a real taste of the student life on campus: living in a college dorm, dining in the cafeteria, working in the lab or library, participating in social activities, and interacting extensively with peers in and outside class. We planned to have two undergraduate student resident assistants who would supervise students outside class, live in the dormitories, and accompany students on all the social activities.

The course had two interrelated major goals:

- Introduce students to fundamental concepts in computing and some of its relevant applications [5], and
- Have students take a very active role in their learning by experimenting with computing techniques and tools [6].

The course material was intended to expose students to web authoring and programming, databases, networking, computer organization, computer graphics, and Java programming. There were four of us in the department (three full-time and one adjunct senior lecturer) who expressed interest in participating in this project. From the very beginning of writing up the proposal, we took a very hands-on approach. We decided that the lab was central to implementing this approach and to teaching students to engage with the subject matter and collaborate with peers. We also planned to supplement the lab activities with three field trips that would bring student experiential learning closer to the real world.

To counterbalance the compressed time format of the course we designed a web-based student portfolio requirement in the course. The purpose of the portfolio was to have students create a learning resource that “tells the story” of what each student would actually do in the course. The portfolio not only would showcase the complete individual work of each student, but would also “go home” with the students at the end of the course. We planned a student presentation session of selected elements from student portfolios in the last day of the course. We also thought of having students save their electronic portfolios on memory USB “flash drives” at the end of the portfolio presentation session. Table 1 shows the overall configuration of the designed course modules.

<table>
<tr><th></th><th>Day 1</th><th>Day 2</th><th>Day 3</th><th>Day 4</th><th>Day 5</th><th>Day 6</th></tr>
<tr><td rowspan="2">9:00 AM to 12:30 PM</td><td><u>Class 1</u><br>Web authoring</td><td rowspan="2"><u>Class 4</u><br>Databases</td><td rowspan="2"><u>Field Trip 2:</u><br>Robotics</td><td><u>Class 7</u><br>Java GUI</td><td rowspan="2"><u>Class 10</u><br>Computer architecture</td><td rowspan="2"><u>Class 12</u><br>Portfolio completion</td></tr>
<tr><td><u>Class 2</u><br>Databases</td><td><u>Class 8</u><br>Networking</td></tr>
<tr><td rowspan="2">2:00 PM to 5:00 PM</td><td><u>Field Trip 1:</u><br>Search engine optimization</td><td rowspan="2"><u>Class 5</u><br>Web authoring</td><td rowspan="2"><u>Class 6</u><br>Java GUI</td><td><u>Field Trip 3:</u><br>College IT Department</td><td rowspan="2"><u>Class 11</u><br>Computer graphics</td><td rowspan="2"><u>Class 13</u><br>Class presentation</td></tr>
<tr><td><u>Class 3</u><br>Databases</td><td><u>Class 9</u><br>Networking</td></tr>
</table>

**Table 1: Overview of the course schedule**

The course proposal was approved in the spring of 2004. The Offices of the Vice President of Academic Affairs and Undergraduate Admissions provided all the necessary funds to cover the incurred costs with program advertisement, mailings, staff salaries, extra-curricular activities, field trips, teaching supplies, etc.

## 3 THE EXPERIENCE

### 3.1 The Logistics

The course was scheduled in the last of week of June, 2004, from June 26 to July 1. The Undergraduate Admissions Office ran several ads in the local newspapers, and mailed brochures and registration applications to potential students in the local area. The department advertised the course on the department web site. The Information Technology Office set up an online registration. Other offices on campus, Conference Services, Student Development, Dining Services, College Relations, Registrar’s, and Public Safety, worked collaboratively with us to make the course happen and run successfully. The undergraduate student resident assistants were two of our majors: a computer science senior and a mathematics alumna who had just graduated in May 2004.

## 3.2 The Students

Twenty-nine students submitted registration applications, twenty-two students were accepted in the program, and fifteen students actually enrolled (68% turnout) - thirteen of which were residential. The screening process took into account the student academic standing and guidance counselor or teacher recommendations. The gender composition was eleven males and four females. The geographical break down was nine students from New Hampshire and five students from Massachusetts. Nine students were from public schools and four students were from private schools.

Students arrived on campus the evening prior to the beginning of the course. The events that evening consisted of check-in, orientation with parents, dinner (for students, faculty, and student assistants), and checking out the computer lab and student computer accounts. Table 2 shows the daily schedule of the overall program.

| Time | Activity |
|---|---|
| 7:30 AM – 8:45 AM | Breakfast and announcements |
| 9:00 AM – 12:30 PM | Class session |
| 12:30 PM – 2:00 PM | Lunch break |
| 2:00 PM – 5:00 PM | Class session |
| 5:30 PM – 6:30 PM | Dinner break |
| 6:30 PM – 11:30 PM | Lab study session and social activities |

**Table 2: Daily Schedule**

At the end of the program, we invited the parents to join the entire group of students, faculty, and staff for dinner and a graduation ceremony. Students made brief presentations of selected portfolio assignments and showed justifiably great pride in their achievements.

## 3.3 The Course

### 3.3.1 Curricular Modules

Web Authoring and Databases

Because our students needed to start creating and filling in their portfolios from the start of the course, we chose Web Authoring as the first topic. Furthermore, because the discussion of web authoring led naturally to the use of databases on the web, we combined the topics of Web Authoring and Databases. There were five sessions, over the course of the first two days, each session being about 2 hours in duration.

First, the instructor introduced the students to the concepts of the World-Wide Web. In particular, the instructor described what happens when they "point" their browser at a web site, and introduced the basic concepts of the HTTP request/response protocol. The instructor then showed the students how to create a web page, using Microsoft FrontPage. HTML tags were mentioned, but not described in detail. The approach was a very hands-on, with a short demonstration, followed by the students reinforcing what they learned by actually creating web pages on their own PCs.

Once the students were comfortable with using FrontPage, the instructor gave them an in-class assignment to create their own portfolio web site. The instructor supplied a portfolio template that the students used as a starting point to create their own. They customized their portfolios to suit their own preferences, using Microsoft FrontPage themes. This satisfied the students' need to produce their own individual look-and-feel for the web site, while still remaining within the portfolio creation rules. As anticipated, some rather imaginative themes were produced.

The instructor progressed from the Web Authoring topic on to a simple introduction to relational databases. First, he gave a motivational presentation about why databases were important, and how they differed from mere data files. Building on this, he described the concepts of data types, tables, columns, rows, and relationships, and their use in a relational database. Finally, he showed the students how to create a simple Pizza database with four tables, using Microsoft Access.

Combining the concepts of the Web and databases, the instructor described how one might use a database from a web site. He used Microsoft FrontPage to upload ("publish") the Pizza database to a web site.

The students were given another in-class assignment, where they created their own Pizza database, given rather precise instructions, and then populated their database with some suitable data. The students continued on with the assignment, and published their Pizza database to their web portfolio, using FrontPage.

The instructor described how one could create "active" web pages, which can present data obtained from a database on the web server machine. He demonstrated how FrontPage can be used to conduct a conversation with a user, and from that information generate an Active Server Page (Microsoft ASP). The students completed a final Web Authoring assignment to create two such active web pages: one to display results from the database, and a second to allow them to insert data into the database.

Networking

Two lectures on networking technologies were presented to students. The first Networking class covered network basics [7], OSI and TCP/IP reference models, local and wide area networks, history of the Internet, broadcast topologies, routing network interconnections, TCP/IP protocols and services [8], connection-oriented and connectionless networks. Also covered were wireless networks, wireless sensor networks, storage area networks [9], X.25, Frame Relay, Asynchronous Transfer Mode, tunneling, and virtual private networks.

The second Networking class covered analog and digital data transmission methods [7], data communication media, ASCII code standards, data encoding, multiplexing, circuit and packet switching principles [8], TCP/IP protocol architecture, IPv4 vs. IPv6, application protocols, and distributed systems.

There were two homework assignments for this class. The first assignment required students to describe installation procedures and plot a diagram for a virtual private connection from their home computer to the College network. The second assignment asked the students to encode a bit-stream that represents the student’s last name written in the ASCII Coding Standard and encoded by using the Manchester Digital Signal Format.

Java GUI Programming

This module was taught in two segments: a three-hour afternoon session and one and a half hour session in the next morning. The topics of the first segment covered GUI applets, event-driven programming, and a commercial IDE – JBuilder. Students were expected to become familiar with basic JBuilder usage and the edit-compile-debug-test cycle; understand fundamental Java programming concepts such as class, object, GUI components, and event; use step-wise refinement to write simple applet programs. The second segment reinforced the programming concepts and practices in the first segment, introduced a few new GUI components, and emphasized code reading, analysis, and documentation. In each segment the instructor guided the students through the steps of writing a simple GUI applet.

Students developed two lab projects in a top-down fashion. The first project was a “photo flipper” applet. It alternated the showing and hiding of an image through the use of two buttons. The second project was a “to do list” editor. The editor functions were to build a list of items in a text area based on text field input information, and to reset the entire list with a clear button. The projects were based on exploratory elements that students solved in a highly participatory and collaborative setting. They asked questions, experimented in groups, used analogies, filled in stubs, or corrected answers offered by peers or themselves.

The class assignments were built on the lab projects. The first assignment was a “my top pic(k)s” applet that expanded the photo flipper into a mini album of favorite pictures selected with buttons. The second assignment asked for code documentation and analysis of the second lab project. For both assignments students produced demo web pages of their applets with descriptions of what they have learned.

Computer Organization

This class was designed to give the students maximum practical exposure by focusing on directed personal exploration of real machines. In preparation for the class, we rounded up a number of obsolete but working personal computers and lined them up on top of long tables. Students stood at the edge of the tables and had access to those machines. In addition, we had some live, working systems, where students could quickly do some exploration and experimentation.

The instructor opened one machine and showed the students the main components: motherboard, processor, memory, buses, video subsystem, network card, USB devices, and so on. This was followed by a period of exploration – students could open their machines, reach inside, identify the individual components, and manipulate them at their will and discretion. There were plenty of machines to allow access by all students, so, after a busy period of individual exploration, students asked lots of questions, not only about the technical aspects of the computers themselves, but also about career options related to the subject under scrutiny, that is, the understanding of how a computer is put together and of how its individual parts interact with each other. A lively dialog about the relevance of the subject knowledge followed, with the instructor addressing future likelihood of work in the area, for example, systems administration and computer networking. Most of the students were college bound, so, the subject of student desk help was of particular interest.

We also bought a number of 32Mb USB memory “flash drives”, and the instructor distributed them to students, and then showed them how to install and use the equipment on a real computer. This allowed for a quick exploration of Plug and Play concepts and on how to add equipment to real computers. Students also used those live machines to run the System Manager and examine the system configuration: what system programs were installed, how much memory, how much disk space, and so on.

The response of the students was very positive. Both the exploratory phase and the ensuing debate were lively and stimulating. There were many requests for additional information and much interaction between students and instructor. This was further helped by the informality of the classroom setting.

The students were given a lab assignment that involved using the Windows "Manage" program to display details of the hardware available on the machine, plus determining the amount of memory on the machine from two different sources, and finally, determining the assignment of the machine's interrupt lines.

Computer Graphics

The class introduced students to the main concepts of computer graphics through the use and simple updating of a sample OpenGL computer program. Concepts of coordinates, color, shading, geometry, and animation were examined by applying simple and localized changes to the sample program, and by observing the results in each student's live system. The instructor made available an OpenGL program that drew a rotating 2-dimensional triangle on a window, and applied little perturbations to the triangle to highlight specific graphics concepts.

Students learned how to use Visual C++ and Microsoft Visual Studio to build and run the program. Once every student had the program running on his or her machine, the instructor successively applied little changes to the code and ran it again. This was used to introduce concepts such as x and y coordinates, how red, green and blue combine to make up colors, and how a triangle can be smooth shaded or flat shaded. The students were given a lab assignment that consisted of downloading an existing OpenGL C source program, compiling and running it, and then changing the color of two triangles displayed by the program.

The emphasis was on basic concepts, rather than on specific computer science issues, and the instructor spent some time talking about the difference between computer artists and computer graphics specialists, and on how graphics-intensive programs such as computer games are created and put into production.

This class drew from our experience with the department's Computer Graphics course, where our majors spend the semester learning 3D computer graphics through live, hands-on sessions in the computer lab. The process can be described as "perturb and observe": the instructor brings a sample computer graphics program that is going to be the subject of the class; students first build and run it unmodified; learning then consists of applying directed small perturbations to the source program, rebuilding and rerunning the program, and learning by observing the results. The student can easily experiment on the material and immediately see the effects of that experimentation and gauge the relationship between theory and practice.

### 3.3.2 Field Trips

Search Engine Optimization

Our first field trip was to a small company run by a recent graduate alumnus of the College. The company specializes in optimizing web searches to ensure that its customers' web sites are high on the list of "hits" for searches against well-chosen keywords.

The owner of the company explored search engines from three perspectives. First, he looked at search engines from the search engines perspective. What are search engines? How do they work to deliver relevant answers to search queries? Next he examined search engines from the searcher's perspective. How do users search for information? How can we improve our chances of finding more and better information more quickly? And lastly, he looked at search engines from the perspective of the website owners who want more people to find and visit their sites. How do you go about getting your site found faster and by more users? Can you buy your way to the top of the engines?

To illustrate his points, he actually went to the Google web site, and had the students set up a paid advertising campaign on Google's web site. Students entered in suitable phrases to try to attract web search "hits" for our summer program web site. The company owner offered a small prize for the student who was most successful in attracting hits by the end of the week. The students enthusiastically entered their slogans, and enjoyed the challenge and competition in a real-world web application. The prize was duly awarded during the presentations on the last day of the course, in front of the students, faculty, and parents.

Robotics Center

The second field trip was to a local Robotics Center, where each team of three students built a scooter robot and three sensor additions (light sensors for orientation, rotation, and stop), using assembly line methods. The team members handled the robot programming, troubleshooting, and operation optimization. One team reported that one of the scooter motors was severely underpowered due to the fact that it was too tight. To fix the problem, a team member adjusted the power of a more powerful motor so that the scooter could travel in a straight line. The others provided insight as to how they could cause the robot to stay on a black tracking line by using repeat commands, and to switch the conditions to make the robot to turn right or left.

Through trial and error, another team managed to get the scooter robot to follow a dark line all the way to the opposite end of the table (the "Arctic Table Challenge" exercise). They also "taught" the robot how to push the target device into the two-point zone. Unfortunately, while trying to optimize the program, the team leader accidentally wrote over the working version, and the team could not salvage it. In the field-trip report, he wrote: "There is a lesson to be learned there, and the lesson is 'Do not ever save over an existing program…especially if it works.'"

The College IT Services Center

During the third field trip to the college IT Services Center, students met with the IT Services director, who demonstrated the state-of-the-art networking equipment and management software. He also described the college Internet Service Provider, data communication media on campus, educational software installed in the College Computer Labs, and how the students can access these resources remotely from their home computers via a Virtual Private Network (VPN).

After this field trip, each student submitted an individual report that contained both practical information about the on-campus networking facilities, hardware and software, and the detailed instructions of how to establish a VPN-tunnel link between their home computers and college computing resources.

### 3.4 The Social Activities

While this course was a fairly intensive and time-compressed course, we anticipated the need for some evening entertainment activities, especially at the start of the course, to help cement relationships and build a team spirit.  So, we arranged for a set of suitable activities.  These included a rather wacky local youth theatrical production, which we attended on the Saturday evening.  One of our students was "pulled" onto the stage, and the resulting interaction was much appreciated by his fellow students, and by those faculty members who were also present. A local “dinner + movie” attraction was attended on the Sunday night.  Despite a late night, the students were all up and active early the next morning.  Other activities included a trip to a local ice-cream place, and an on-campus movie night.

## 4 LESSONS LEARNED

More high school students attracted to and better prepared for computer science is every computer science educator’s dream. Designing and running a pre-college summer course is not as simple as one might initially think. The enthusiasm of giving the sponsoring department a sharper visibility with the objective of attracting more students is not sufficient. We learned that a pre-college summer program requires clear goals, careful planning, a motivated team of dedicated teachers, supportive offices, high-quality campus services, appropriate pedagogy, and finely tuned receptiveness to student needs and expectations. We believe that we were successful in meeting these standards. We also know that there is room for improvement.

The driving force of the entire enterprise was the clear set of goals articulated from the very beginning. Complementing the more obvious goal of promoting our undergraduate programs, we were very interested in piloting a hands-on, lab-based course in which students can showcase their learning in an electronic portfolio. We anticipated that high school students eager to invest their summer time into being part of a college experience in the computing discipline are the most valuable critics of an innovative and demanding program. The students set the bar high and we were ready to meet their expectations. Our next endeavor is to revamp the computer science course for non-majors following the model of the summer course. We have the proof that it works and we plan to prepare a proposal.

The process of putting in place a summer program can be daunting when it is a totally new program on campus, there is no external funding, and the timeframe is tight. We were encouraged to learn that “over 30,000 high school students, many quite savvy, attend these programs every summer” [10]. However, many of the schools that traditionally offer these programs are prestigious research universities that also recruit talented and ambitious high school students through pre-college programs. The number of summer programs, especially in sciences, offered by small liberal arts colleges is very limited. We are interested in exploring avenues for external funding, and in partnering with other departments at small colleges who have run, or are interested in running, summer programs.

The key to the success of our summer course was the teaching team. The team was formed early on, and it consisted of talented and dedicated teachers, with sheer energy and enthusiasm to make the program happen. The advantage of working together on an interdisciplinary course was the constant exchange of teaching techniques, assessment methods, lab activities and tools, course materials, etc. We informed ourselves of what works and does not in our own areas of study, and strengthened our professional ties.

Student evaluations were positive, with good marks for the quality of teaching, the campus experience, and overall organization. The majority of students concluded that they enjoyed the course and learned a lot. The students were, in general, positive about the degree of integration of the field trips in the academic program. Some concerns were raised about the need for more and detailed directions in preparing the homework assignments. Students correlated this aspect with the limited time for homework and the fast pace of the course. Although they found the six-day format with morning and afternoon classes very compressed, they indicated that their preference is for no more than a two week long program. We think that the course will improve if some class time is set aside to have the class go over the assignment requirements in more detail and give opportunities to clarify any question at that time. The course assistants were available in the lab when the students were working on their assignments, and the instructors could be reached via email. However, due to time constraints, this type of conversation was limited, and, at times, additional clarifications gave the impression that the assignment statements had changed. We learned that sometimes our set of assumptions about what and how we asked students to proceed with their assignments did not match their set of assumptions.

Finally, the success of the computer organization module and the student positive evaluation of the Robotics Center field trip provided us with first-hand evidence about the importance of a specialized computer science lab [11]. We plan to use this evidence in preparing our proposal to the administration for setting up such a lab.

## 5 CONCLUSIONS

We have described the goals, logistics, design, implementation, and experience of running a six day Summer Institute in Computer Science for high school students. The experience was in general a very positive one, while at the same time providing useful lessons learned. We believe that this approach can be effectively applied to future courses of a similar nature, and believe that the model can be extended to other disciplines beyond the computing sciences.

## 6 ACKNOWLEDGEMENTS

Many offices on campus, colleagues, staff, and administrators played a direct and crucial role in the organization and success of this program. We recognize that, in a college of our size, different constituencies often work more closely and productively together than might be the case in larger institutions. This is to acknowledge that we “lived” this advantage and we are very grateful to all those involved in making the program a success.

## 7 REFERENCES

[1] Zweben, S. and Aspray, W. 2002-2003 Taulbee Survey - Undergraduate Enrollments Drop; Department Growth Expectations Moderate. In *Computing Research News 16(1)* (May 2004), 5-19.

[2] Foley, J. Old Challenges, New Opportunities. In *Computing Research News 16(4)* (September 2004).

[3] Morris, J.H. and Lee, P. The Incredibly Shrinking Pipeline Is Not Just for Women Anymore. In *Computing Research News 16(1)* (May 2004), 20.

[4] Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., Verno, A. Implementation challenges for K-12 computer science curriculum. In *Proceedings of the 35$^{th}$ SIGCSE Symposium* (Norfolk, VA, March 2004).

[5] Zimmerman, B. Content and Laboratories of a Computing Science Course for Non-Majors in the 21$^{st}$ Century. In *Journal of Computing Sciences in Colleges 19(5)* (May 2004).

[6] Whittington, K.J. Infusing Active Learning into Introductory Programming Courses. In *Journal of Computing Sciences in Colleges 19(5)* (May 2004).

[7] Tanenbaum, A. S., *Computer Networks, 4th ed.*, Prentice Hall, 2003.

[8] Stallings W., *Data and Computer Communications*, 7$^{th}$ edition, Prentice Hall, 2003.

[9] Riabov, V. *Storage Area Networks*, The Internet Encyclopedia, Vol. 3, Wiley & Sons, 2004, pp. 329-339.

[10] Hydrisko, R. (editor). Early College Programs. The Nautilus Publishing Company (2002).

[11] Wooley, Bruce A. Utilizing a Computing Lab to Improve Retention and Recruiting of Computer Science and Computer Information Science Students. In *Proceedings of the 18$^{th}$ Annual CCSC Eastern Conference* (Bloomsburg, PA, October 2002).