

Exploring Computer Science – A Freshman Orientation and Exploratory Course

Stephen U. Egarievwe and Vivian J. Fielder

*Center for Internet Based Education and Research
Department of Mathematics and Computer Science
Fisk University, Nashville TN 37208*

segariev@dubois.fisk.edu

Abstract

Computer Science being the youngest discipline in many colleges and universities is still undergoing serious curriculum development, and teaching methods are being improved. “Exploring Computer Science” is a new freshman course intended for both CS majors and non-majors who have strong interest in Computer Science. The course is designed to give students an overview of the discipline – content, career opportunities and some foundational skills. Thus, it serves as both an orientation and a gentle preparatory course. In addition to contributing to the retention of CS majors the course has attracted new majors to the Computer Science Program. One sophomore who indicated her intention to change major from computer science decided not to change after taking this course in spring of 1999. Another freshman chemistry major and sophomore biology major that took the course in spring of 2000 decided to change their majors to computer science. The course promotes students’ enthusiasm for pursuing a course of study in CS.

1. Introduction

Curriculum development and improvement of instructional methods are getting serious attention from computer science educators. Computer science (CS) is a very young discipline, and it will continue to expand as a result of its usefulness in both old and new fields. Its growth is strongly encouraged by the endless human search for new grounds in all aspects of life, and better ways of doing things. Thus curricula in various institutions will continue to experience changes and additions of new courses. Some of the courses will focus on CS theories and concepts, while others will focus on applications. The need to bridge learning levels, adequately prepare students for other courses, retain and attract CS majors, will lead to the development of other new courses.

The shortage of computer skilled workers to the ever-increasing job market in this and related fields is obvious. There is need to increase student enrollment in CS and retain them in the field. Unfortunately, as freshmen leave computer literacy courses and began to take programming courses, some have difficulties in grasping the subject. The result is change of major to other disciplines.

There are several first-year CS courses in many institutions that are designed to provide some form of introduction to certain areas of computer science. The most common are computer literacy, introduction to programming, and introduction to computer science. As observed by Cook [1], these courses offer very little orientation topics and activities. In describing a CS freshman orientation course, Cook stressed the importance of orientation courses which give an overview of the discipline, what the profession is like, and what professionals do; these topics will enable students to make informed decisions about their educational and career goals [1]. At Fisk University we have developed a CS course titled “Exploring Computer Science” which serves as both an orientation and exploratory course that gently introduce students to CS and prepare them for some upper level courses. The development of the course (including content, instructional methods, students’ activities, students and course evaluations, and achievements) is briefly presented in this paper.

2. Motivation and Objectives

The idea of developing this course came from the needs that were identified in teaching freshmen and advising students [2]. These needs led to the following objectives:

1. *Increase students’ understanding of the CS discipline.* To achieve this every student need to be exposed to the various aspects of both theoretical and applied CS. This will include detailed description of the various CS areas, and the courses students need to take from freshman level up to graduation.
2. *Promote student’s enthusiasm for pursuing a course of study in CS.* Many students coming into the CS discipline, as freshmen, are quite enthusiastic about becoming computer scientists. However they begin to experience difficulties as they get into programming and other CS core courses. They soon realize that CS is more than using word processors, spreadsheets, the Internet, and the other everyday applications. This causes the level of enthusiasm to decrease. To combat this problem the new CS course must sustain the enthusiasm of freshmen by involving students in motivating activities and preparing them for what to expect from freshman year to the senior year. Such activities could be designed around students’ interests.
3. *Retain CS majors and attract new majors.* The above two objectives will help in retaining CS majors. To attract new CS majors, the course has to be open to none majors. For this group of students, the course should be able to ‘sell’ CS to them. The course content and activities should be interesting and make CS attractive also to none majors, especially those that have not decided on a major.
4. *Provide orientation for students.* This includes a gentle exposure of students to the various aspects of CS and the overview of what the profession is like, and what professionals do. The course presents the scope of CS in terms of meaning, applications, careers, social and legal issues, and future courses required of CS students.

As stated by Eschen *et. al.*, this course will also fill the need for none majors who are interested in going beyond the literacy course, but are not sure they want to leap into the CS core curriculum [2]. The course will provide students with early successes in a variety of CS activities, and empower them with a vision of their potential roles as computer scientists.

3. Course Organization and Structure

The course is organized to achieve the objectives stated in section 2. The components of the course are briefly described below:

1. *Regular class sessions.* The regular taught class sessions take about 60% of the entire course. The course content for this session is discussed in section 4.
2. *Seminars and workshops.* These cover about 10% of the course. The seminars are given by faculty members, upper level students, outside professionals, and Fisk CS graduates (with jobs in industries or who are in graduate schools). Workshops on CS topics are also organized for students' participation. Students are encouraged to participate in outside CS related workshops.
3. *Laboratory sessions.* The laboratory session is about 10% of the course. The students are introduced to the CS laboratory and are given lab assignments. Some of the lab assignments cover topics in the regular class sessions. Other assignments cover the use of the CS laboratory and introduction to one or more programming languages.
4. *Group projects.* Students are assigned group projects to encourage and introduce them to working in groups. The group work takes about 15% of the course.
5. *Field Trips.* One or more field trips are planned for the students, depending on availability of funds. This should account for about 5% of the course. Before the trip, students are shown videos that describe the industrial sector for the planned visit and the applications of computer science in that area. Students are encouraged to interact with the host industry. Short written reports, describing field trip experiences, are required from the students.

The detailed contents and coverage of these components may vary from year to year depending on the evolution of the CS discipline and students' needs.

4. Course Content Outline

In designing the course content we considered the objectives, time, and resources. We also referred to Cook's paper on CS freshman orientation course [1]. Cook stated that a CS orientation course should provide opportunity to 'sell' the CS discipline, make the students feel part of the CS community, teach some basic technical skills, and give an overview of CS [1]. The course is assigned two credit hours. It is presently allocated two hours per week, including lectures, labs, projects sessions, and the other activities. It might be necessary to have additional time for the lab sessions. The course content outline is summarized as follows:

1. *Introduction to computer science:* Definition of computer science and overview of CS components, applications, professionals, and content of this course.
2. *Algorithm and problem solving:* Problem solving steps in computer science. The definition of an algorithm. Algorithm representation – pseudocodes and flowcharts. Basic algorithm operations – sequential, conditional, and iterative. Brief introduction to the efficiency of algorithm.
3. *A brief introduction to programming in C++:* The C++ programming environment – edit, preprocess, compile, link, load, and execute. Organizing a C++ program –

preprocessor directives and header files, function 'main' and other functions. Input and output statements. Data types, variables and constants, and arithmetic operators. Sequential, conditional and iterative operations – *if*, *if else*, *while*, and *for* statements. 2-dimensional array.

4. *Binary numbers, Boolean logic, and gates*: Basic arithmetic in binary number system. Binary representation of information. Binary storage devices. Boolean logic and gates. Building computer circuits.
5. *Computer system organization*: A brief history of computers. The Von Neumann architecture – memory, input/output, arithmetic and logic unit (ALU), control unit, central processing unit (CPU). An elaborate structure of the computer architecture – cache and main memory, data access and storage (*RAM* and *ROM*).
6. *The virtual machine – Operating systems and assembly language*: The virtual machine and system software. Assemblers and assembly language. Operating systems.
7. *Overview of high-level languages*: C++, Java, HTML, Fortran, Lisp, LabVIEW, etc.
8. *The CS curriculum, computer applications, CS professionals, and the job market*.
9. *Social and legal issues of computers and the Internet*.

The last two topics are presented to students in the form of seminars. Experts in the areas concerned do this. The standard textbook currently used for this course is "An Invitation to Computer Science," by Schneider and Gersting [3].

5. Instructional Methods and Activities

A variety of techniques were used in teaching various parts of this course. In the regular class sessions, the instructor is mainly in control. Students freely ask questions. Students sometimes ask questions that are not related to the class topic. Rather than ignore such questions/students, an answer may be given in the context of computer science and as close as possible to the topic being taught. The idea here is to refrain from any action that may discourage students and make them think they do not belong. During lectures students are asked questions to check how well they understand class materials. When a student gives a wrong answer he/she is asked a simpler question that will give a clue to answering the original question. In summary, we are gentle with the students, and at the same time make sure that our standards and objectives are met. There are both individual and group homework assignments.

Group projects may be in form of programming assignment or simple research on computer applications, CS professionals and jobs, social and legal issues, and the Internet. Some of these projects involve seeking public/expert opinions through the use of survey questionnaires. Students analyze the results and write a short report on their projects. Students are also required to submit reports on any seminars, workshops and field trips they participated in. The group project topics for spring of 2000 centered on the Internet, including education, ethics, crimes, research, shopping, and communication.

The first computer science seminar designed for the students in this course was held on April of 1999. It was advertised to the entire Fisk University campus. Students who were not in any CS class were also impressed by the success of our graduates. It made both CS majors and none majors aware of careers that are available in computer science and technology. The presenters were quite impressive and represented themselves and Fisk University well. The presentation

topics were in the areas of CS curriculum and industrial needs, real job experiences, graduate schools, and CS professionals.

In the spring of 1999 students went on a field trip to Saturn automobile plant in Spring Hill, Tennessee. Before visiting the Saturn plant students watched a video on automobile plants with emphasis on computer applications and robotics. There was an open discussion in class, before and after the field trip. The field trip reports by students cover the video, their interactions with the plant workers (especially robotics professionals), and their overall experience.

6. Students and Course Evaluation

The results of questionnaires and surveys used throughout the course showed improvement of students' understanding of the meaning of computer science, its applications, and related career, social and legal issues. The evaluation of the class taught in spring 1999 showed that the number of students that completely understood the meaning of CS increased from zero to 26% by mid-semester and increased to 60% by the end of the semester. The career goals of students fluctuate as the meaning and scope of CS become clearer to them. The seminar featuring Fisk graduates in April 1999 impacted significantly the students' idea about what computer science is and about what computer scientists do. Initial evaluation of the course this spring shows similar results.

7. Achievements and Conclusion

The new course has contributed to increasing the retention of majors in computer science, and attraction of new majors. One sophomore who indicated her intention to change major from computer science decided not to change after taking this course. Another freshman chemistry major and a sophomore biology major that took the course this spring, decided to change their majors to computer science. The course has encouraged several students to participate in computer related summer jobs, internships, and undergraduate research. In conclusion, this course has served well in providing orientation to the freshmen and giving them the opportunity to explore the CS discipline and related career options.

Acknowledgement

This work was supported by NSF-MII grant #9810673. We acknowledge the contribution of E. M. Eschen for the development of the NSF-MII proposal that resulted in this grant award. We thank the following students (who have graduated) for serving as teaching assistants to this course: Kevin L. Thaxton, Martina Davis, LaTonja Johnson, and Christopher Thorpe.

References

1. C. R. Cook, "A Computer Science Freshman Orientation Course," *SIGSCE Bulletin*, Vol. 28 No. 2, pp. 49-61 (June 1996).
2. E. M. Eschen, V. J. Fielder, and N. H. Mann, "Building Computer Science at Fisk University," *NSF/Special Projects Proposal* (1998).
3. G. M. Schneider and J. L. Gersting, *An Invitation to Computer Science*, 2ed, Brooks/Cole Publishing Company, Pacific Grove, CA (1999).