MAKING COMPUTING ATTRACTIVE FOR NON-MAJORS: A COURSE DESIGN*

Stan Kurkovsky
Department of Computer Science
Central Connecticut State University
1615 Stanley Street, New Britain, CT 06050

ABSTRACT

This paper presents an introductory computer science course "Introduction to Internet Programming and Applications" offered to non-computer science majors. The main objectives of this course are to introduce students to the field of computing, keep them engaged in the course and perhaps spark their interest in the field of computing. This paper also presents the results of a study of student perceptions of computer science and their reactions to this course.

1. INTRODUCTION

Computer science (CS) is often perceived by students as being significantly more difficult than other academic disciplines. Students majoring in CS make a conscious choice about taking CS courses of varying degree of difficulty, while non-majors often take these courses because they are required to do so by their programs of study. Some students view CS as "nerdy" and "not a cool course to take" as mentioned in the study detailed in this paper. Many non-majors do not understand that "computer science is no more about computers than astronomy is about telescopes," as Dijkstra once said.

Introductory CS courses for non-majors are generally divided into computer literacy courses that do not teach any programming and "CS 0" courses that give a broad overview of CS as an academic discipline and usually include some programming. Our primary interest concerns the latter. There is an ongoing debate whether programming should be taught in such courses. Despite the recommendations expressed in the 1999 National Academy of Sciences report "Being Fluent with Information Technology" [2], some believe that teaching programming to non-CS majors makes no significant impact on their chances of becoming sophisticated computer users [6,7]. Many others believe

---

that programming techniques [4,5,8] or highly tailored programming (such as media computation [3]) should be taught as the main subject in an introductory CS course for non majors. Furthermore, given the nature of such courses, it becomes increasingly important to address the needs of students with different educational backgrounds [1,3,8].

In this paper we discuss an introductory CS course titled "Introduction to Internet Programming and Applications" offered at Central Connecticut State University designed to introduce non-CS majors to the basic concepts of computer programming through study of the Internet, Internet-enabled applications and their impact on today's society. Through this course we address the issue of increasing motivation among non-CS students to learn more about computing and information technology. We also present our ideas that helped us encourage many students to be more engaged in the course material. This paper is organized as follows. Section 2 details the design of our course. Section 3 describes the study conducted to validate the design of our course that focused on the students' perception of CS and how their attitudes can be changed by carefully selecting the course topics. Section 4 concludes the paper with a summary.

2. COURSE DESIGN

The immediate objective of designing "Introduction to Internet Programming and Applications" was to introduce non-CS majors to the field of computing by showing them how their everyday activities online and offline are influenced by computers and information technology. A more subtle objective of this course is to combat the indifference towards computing, motivate students to actively participate in the course activities and perhaps encourage them to look further into the possibility of more in-depth studies of computing. While covering simple course topics, we also try to expose students to related but more advanced CS concepts, such as algorithms, software lifecycle, network architecture, computer security and others. Specific areas covered in our course are briefly discussed below.

Internet: Yesterday, Today and Tomorrow

We introduce a historic perspective of the development of the Internet starting from ARPA research in the early 1960's. We gently introduce concept of packets and packet-switching networks and contrast them to circuit-switching networks. Students always like the analogy between packet-switching networks and the process of sending a postcard in the mail and its routing across different mail facilities. We also talk about bandwidth and different network access technologies currently used by ISPs, such as dialup, DSL and cable.

World Wide Web

For many students, the web *is* the Internet and many of them do not realize that there is far more to the Internet than a simple web surfing. We discuss the interaction between web servers and browsers; thus students are gently introduced to client/server architecture. Usually, students are fascinated to learn about browser wars; furthermore, now they can witness the second phase of browser wars between Firefox and Internet Explorer. During the discussion about web browsers students get their first preview of HTML, its most commonly used tags and their use in constructing a web page.

Communication and the Internet

Most students are very familiar with everyday use of email, but very few are aware of how email messages are actually delivered. We refer back to the concept of packet-switching networks to illustrate the process of email delivery. We also talk about email headers and their role in email delivery. Students can experience anti-spam and anti-virus systems at work when they examine email headers created by these systems. Students are offered a lab, in which they sign up for several free email accounts and send email messages from one account to the other to analyze the resulting email headers. By doing so, students can better understand the concept of IP addressing and routing.

Web Information Resources

Most students feel very comfortable with using at least one web search engine, such as Google or Yahoo Search. However, very few of them know how to use advanced features of these search engines. Discussion of these features naturally leads to the explanation of the general functionality of a web search engine, its structure and the process of content indexing. Students get a further look at HTML and the structure of URLs, as they are essential the indexing performed by web search robots. Lab activities offered to the students include performing a web search on the same terms using different search engines. In some cases, the search results are drastically different. We use this to explain the difference in indexing and ranking algorithms between different search engines, which provides an opportunity to introduce the notion of algorithm.

File Sharing and Media on the Internet

Most students are very knowledgeable about the concept of file sharing, but not all of them fully understand its ethical and legal implications. Some students wrote in an essay on this subject that while it's illegal to download copyrighted material, they are not inclined to "give any money to the already filthy rich recording artists and actors". It may be difficult to change the students' opinions on this subject outright, but they must be exposed to legal and ethical aspects of using copyrighted content. Most students are currently using one of the current file sharing systems. To illustrate the issues of legality, we usually discuss the history of Napster, its rise and decline, and how eventually its trademark became a subscription-based music service. Students also learn about the recent court orders to force the manufacturers of file sharing systems to filter copyrighted material off their networks. While on the subject of copyright, we always remind students about the importance of giving a proper credit to all sources, printed or online, that they use in their work, e.g. while writing term papers and using online sources as references.

Internet Security

It is very important to introduce students to the fundamental concepts of computer and network security and our course may be the only place in the college curriculum where students could get a proper exposure to this subject. We start with such basic issues as importance of using hard to guess passwords and locking or logging out of their computers while they are not using them. It is important that students understand the consequences of having one's computer being hijacked and being used for denial of service attacks, sending spam or using it to further propagate malicious software. We also discuss the importance of having the most recent security updates for the operating system, web browsers and other applications. This leads to the issue of software evolution and the concept of software lifecycle. Most students in this course will not become

computer scientists, but all of them, throughout their college years and later in their career, will likely be working with computers and software in their day-to-day activities, and therefore it is very important to expose them to the problem of changing and evolving software applications and systems.

HTML and web design

Programming plays an integral role in CS. HTML authoring takes students with no technical background closer to programming. Most students in our course enjoy the transparency with which they can transform a plain piece of text information into an appealing web page. They can immediately see a connection between the results of their work and the content of many web sites that they visited. Working with HTML documents, image files, local and external URLs helps students gain a deeper understanding of the client/server principle and aspects of communication between web servers and browsers. Many students are immediately interested in the assignment sequence to design their home pages that eventually grows into a multi-page personal web site created with dynamic HTML. Many creatively inclined students are especially interested in the topic of web design as illustrated by the results of our study. The author was truly impressed by the creativity combined with technical mastery exhibited by several arts and humanities students with no previous exposure to HTML or web design.

JavaScript and basics of programming

The course concludes with a study of elements of computer programming with JavaScript. We discuss the most essential programming elements, such as variables, data types, arrays, conditions, loops and functions. We are constantly experimenting with different programming assignments and projects that would appeal to non-CS majors. Usually, given the typical student body of this course, we try to steer away from any math-related examples. However, students in our course are usually capable of writing small JavaScript programs that calculate a GPA or total purchases on a shopping list.

3. STUDYING STUDENT MOTIVATION AND ATTITUDES TOWARDS CS

We conducted a study in three sections of our "Introduction to Internet Programming and Applications" course with a total enrollment of over 60 students. In this study, partly influenced by the work of Forte and Guzdial [3], we offered a number of surveys to elicit student perceptions and opinions about this course. Not all data collected as a result of this survey is presented in this paper.

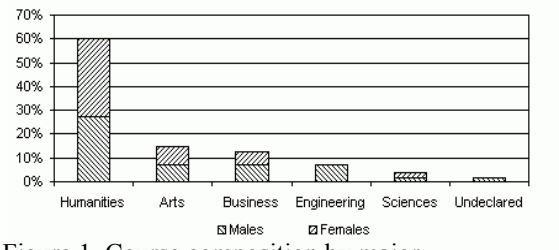
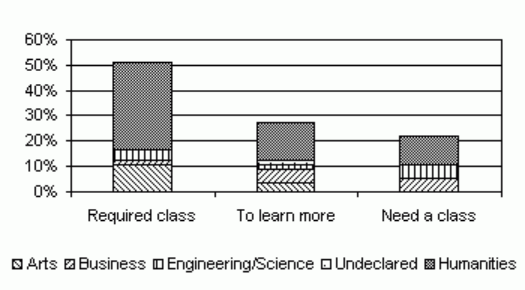


Figure 1. Course composition by major.

This course is a part of the general education requirements in most undergraduate programs offered at our school; students often have to choose between this or a math course to satisfy their mathematics area requirements. However, in some programs students do not have such a choice. This explains a very high enrollment of humanities majors and a very low enrollment among engineering and science majors. Generally, arts and humanities majors take this class because it is required by their programs. This trend

is demonstrated in Figure 2 (left) showing that over a half of all enrolled students were required to take this or a math course. Approximately 27 percent of all students enrolled in this course intentionally because they wanted to learn more about computers, while almost 22 percent of the students simply needed some class to take. Figure 2 (right) illustrates how the students' reasons for taking this course vary among different majors. The majority of arts and humanities students enroll because they are required to do so, in contrast to business, engineering and science majors who enroll because they want to learn more about computers or because they just needed another class. We did not expect a sizeable number of humanities majors (approximately 35% of this major) to enroll in this course with an intention to learn more about computing.



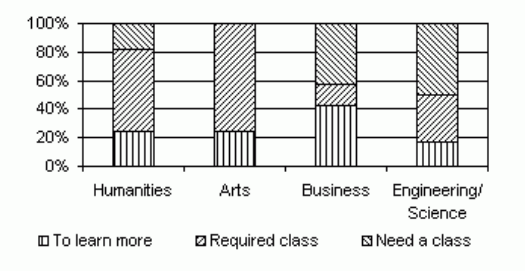


Figure 2. Reasons why students enroll in course.

One of the goals of our study was to learn about the student perceptions of CS. Better understanding the views of students helps us address any possible misconceptions and make sure that our course presents students with a well-rounded treatment of the subject accessible at the level of the audience. Figure 3 (left) represents the students' beliefs about what CS is. Students were allowed to select several answers to this question, thus percentages may not sum up to 100. It is not surprising that many students believe that CS is about using computers in their everyday life and surfing the internet. At the same time, over a half of all students were interested in learning about the inner workings of computer systems, be it hardware or software. An even greater percentage was interested in learning about programming and authoring web pages.

Figure 3 (right) shows the distribution of the students' beliefs about CS based on their major. Not surprising, engineering and science students were interested either in programming or in understanding how computers work. Business, arts and humanities students presented a more varying range of opinions indicating that most of them have no clear idea of what CS is or what they expect from this course. For example, "writing programs" was the least popular answer among humanities students when they were asked what is CS to them. Among business, arts and humanities students, over 60 percent believe that CS is about surfing the web, finding things on the internet and using computers for day to day tasks. As can be seen from this data, it is of paramount importance to help students understand that there is a vast difference between computer literacy and CS. Our course may be the only course they will ever take where students will have a chance to learn about this difference.

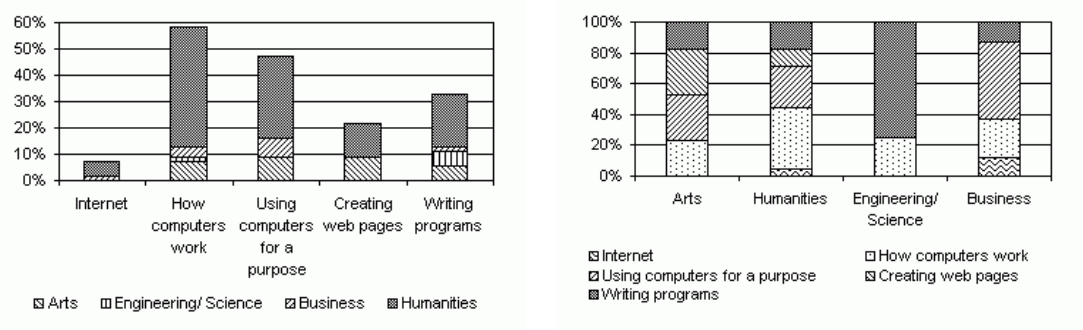


Figure 3. What is Computer Science to you?

Keeping in mind that this course is targeted at non-majors, most of whom lack motivation and/or proper background, we strive to keep the students interested by constantly illustrating the connections and relationships between many areas and concepts of CS and experiences that students may have already had or will have with computers and technology. We have to take into account that most of students today are immersed in the Internet, they use mobile technology in their everyday life and most of them are well aware of recent technological advances. As we discuss many new developments and the impact of the Internet technology on our lives, it is impossible not to cover various implications this technology has on our society; such discussions are usually well received by all students, but students in humanities (and social studies in particular) enjoy them the most. Figure 4 illustrates what students like best about CS as it was presented in this course. Here, the percentages may not add up to 100 since the students were allowed to select more than one answer. Figure 4 (left) shows that among different aspects of course content, each of them appealed to 25 to 36 percent of the students. It is slightly disappointing that 5 percent of the students indicated that they did not like anything about the course. Data shown in Figure 4 (right) indicates that these are science, engineering or business majors. Furthermore, written comments solicited from students during the survey indicate that some of these students have already had a similar course in the past and most of material of this course was not new to them.

As we expected, different aspects of the course appealed to different groups of students, as shown in Figure 4 (right). For example, learning about social implications of computing appealed to arts and humanities students, but not to science, engineering or business majors. Also, programming, designing web pages and acquiring information technology skills appealed less to students majoring in humanities than to students majoring in other areas.

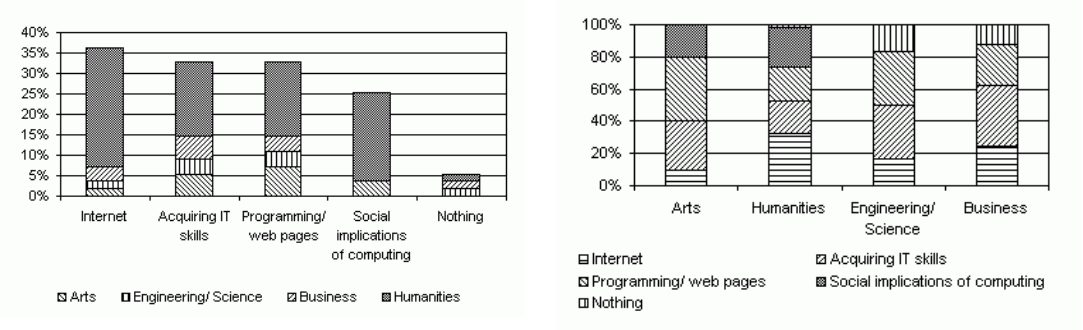


Figure 4. What do you like best about Computer Science in this course?

4. CONCLUSION

Our study indicates that it is extremely important to address many misconceptions common among non-majors about what CS truly is. Educating student population about the nature of this subject area might help fight declining enrollments in CS programs, although this alone will not be enough to bring the enrollments in CS programs to pre-2000 levels. Based on the results of our study, we believe that our course "Introduction to Internet Programming and Applications" makes a step in the right direction by engaging students in the course material and by addressing the needs of students who might feel a lack of motivation.

5. REFERENCES

[1] S. Beyer et al. "Gender Differences in Computer Science Students," in Proc. SIGSCE 2003, pp. 49-53.

[2] Committee on Information Technology Literacy, "Being fluent with information technology," National Academy of Sciences, Washington, DC, 1999.

[3] A. Forte, M. Guzdial. "Motivation and Nonmajors in Computer Science: Identifying Discrete Audiences for Introductory Courses," IEEE Transactions On Education, Vol. 48, No. 2, pp. 248-253.

[4] N. Herrmann. "Redesigning Introductory Computer Programming Using Multi-level Online Modules for a Mixed Audience," in Proc. 34th ACM Special Interest Group Computer Science Education (SIGCSE) Tech. Symp. Computer Science Education, 2003, pp. 196–200.

[5] D. Joyce, "The computer as a problem solving tool: a unifying view for a nonmajors course," in Proc. 29th SIGCSE Tech. Symp. Computer Science Education, 1998, pp. 63-67.

[6] N. Kock, R. Aiken, C. Sandas, "Using complex IT in specific domains: Developing and assessing a course for nonmajors," IEEE Trans. Educ., vol. 45, no. 1, pp. 50–56.

[7] M. Urban-Lurain, D.Weinshank, "Mastery Model Learning for a Large Non-Major Course," in Proc. SIGCSE 1999, pp. 15-154.

[8] B. Wilson, S. Shrock, "Contributing to success in an introductory computer science course: A study of twelve factors," in Proc. SIGSCE 2001, pp. 184–188.