

# Improving Student Motivation in a Computing Course for Non-Majors

Stan Kurkovsky

Department of Computer Science  
Central Connecticut State University  
New Britain, CT, USA  
Email: kurkovskysta@ccsu.edu

**Abstract** - *This paper focuses on introductory computer science courses offered to students not majoring in computer science. In particular, the issues of student motivation and engagement in the course material are the most crucial and they should not be overlooked in the process of course design. This paper presents the results of a study of student perceptions of computer science and their reactions to “Introduction to Internet Programming and Applications” course in which they were enrolled. 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 discusses the main highlights of the design of this course.*

**Keywords:** Computer Science education, courses for non-majors, Internet

## 1. Introduction

Computer science is often perceived by students as being significantly more difficult than other academic disciplines. Students majoring in computer science make a conscious choice about taking computer science courses of varying degree of difficulty, while non-majors often take these courses because they are required to do so by the curricula of their programs of study. Furthermore, some students view computer science as “nerdy” and “not a cool course to take” as some of them 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 computer science courses for non-majors can be generally categorized as computer literacy courses that do not teach any programming and “CS 0” courses that give a broad overview of computer science as an academic discipline and usually include some

programming. Our primary interests concern the latter courses that go beyond the basic computer literacy skills. 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 titled “Being Fluent with Information Technology” [2], some researchers believe that teaching programming skills to non-computer science majors makes no significant impact on their chances of becoming sophisticated computer users [6, 7]. Many others believe that regular computer programming techniques [4, 5, 8] or highly tailored programming (such as media computation [3]) should be taught as the main subject in an introductory computer science course for non majors. Furthermore, given the nature of such a course, it becomes increasingly important to address the needs of students with different educational backgrounds [1, 3, 8].

In this paper we discuss an introductory computer science course titled “Introduction to Internet Programming and Applications” offered at Central Connecticut State University that is specifically designed to introduce non-computer science majors to the basic concepts of computer programming through the study of the Internet, many Internet-enabled applications and their role and impact on today’s society. Through this course we address the issue of increasing motivation among non-computer science 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 by connecting the course topics with their past experiences and interests. This paper is organized as follows. Section 2 details the design of the “Introduction to Internet Programming and Applications” course. Section 3 describes the study that we conducted to validate the design of our course, in which we learned about the students’ perception of computer science and how their attitudes can be changed by carefully selecting the topics covered by an

introductory computer science course. Section 4 concludes the paper with a summary.

## 2. Course Design

The immediate objective of designing the course “Introduction to Internet Programming and Applications” were to introduce non-computer science majors into 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 of students towards computing, motivate them to take an active participation in the course activities and perhaps encourage them to look further into the possibility of pursuing more in-depth studies in the field of computing.

Despite the fact that the vast majority of today’s college students have experts skills in web surfing, very few of them actually realize the big picture of the Internet architecture and are not aware of many existing Internet applications other than the web and file sharing.

Throughout the entire course, we tried to find real-world situations, examples and analogies, to which the students can relate. More complex technical concepts are introduced through these analogies and students find this approach very useful, which is substantiated by our study detailed later in this paper. While covering seemingly simple concepts, we also try to expose students to related, but more advanced computer science concepts, such as algorithms, software development lifecycle, network architecture, computer security and many others.

Specific areas covered in our course are briefly discussed below. The topics presented here are studied in the course in the same sequence although many of them are significantly intertwined; most topics are also frequently revisited throughout the course to add more detail and depth of coverage as students learn more material necessary to understand these more intricate concepts.

### 2.1. Internet: Yesterday, Today and Tomorrow

We introduce a historic perspective of the development of the Internet starting from the research at ARPA in the early 1960’s aimed at developing a fault tolerant network that could survive a nuclear attack. Little by little we introduce the 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. The vast majority of students have hands-on experience with dialup and either DSL or cable. Students appreciate technical details of how these technologies operate because this gives them a further insight into something that they have already experienced.

### 2.2. 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 web clients (browsers) and thus students are gently introduced to the client/server architecture which is mentioned many more times throughout the course. Usually, students are fascinated to learn about the browser wars (the competition between Microsoft Internet Explorer and Netscape Navigator that took place in late 1990’s) – many of our current students were actively using computers and surfing the web at the time and therefore they can very well relate to this topic that puts their personal experiences in a bigger picture. Furthermore, now they can witness the second phase of the browser wars when Mozilla Firefox browser poses a serious threat to the domination of Microsoft Internet Explorer in the browser market. 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.

### 2.3. Communication and the Internet

Typically, this subject means email, but students usually are very familiar with the everyday use of email for communication. However, very few of them 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. In such a course, it is necessary to give a preview of some essential technical concepts; other concepts, however, such as packet encapsulation and fragmentation, need to be omitted for the sake of simplicity. We also talk about email headers and the role that they play in email delivery. While students may know about spam and viruses, they often get hands-on experience in seeing 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 from such web sites as Yahoo or Hotmail. Students then send email messages from one account to the other and analyze the resulting email headers. By doing so they can better understand the concept of IP addressing and routing because they can match these concepts to what they see in the headers of email messages created during this lab activity and IP addresses of their computers. Many textbooks covering this topic

also include a discussion about instant messaging, chat rooms and usenet newsgroups; however, we find it sufficient to just highlight the functional differences between these different technologies, but only focus on email for the purposes of an in-depth technical discussion.

## **2.4. Web Information Resources**

Again, 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 in particular at the structure of hyperlink tags and URLs, since they are essential for understanding the process of how a web search engine's indexing robot traverses the web space. Students usually find it fascinating that web search engines may be used to discover information that normally may be very difficult to find, such as some public records. Among the lab activities typically offered to the students on this subject is performing a web search on the same terms using different search engines, for example, Google and AltaVista. In some cases, the results are the same, while in others they are drastically different. We use this to explain the difference in indexing and ranking algorithms between these two different search engines. Not only students get a better understanding of the inner workings of these two popular web sites, we use also this opportunity to introduce the notion of algorithm.

## **2.5. File Sharing and Media on the Internet**

Despite the fact that most students are very knowledgeable about the concept of file sharing, 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 the legal and ethical aspects of using copyrighted content, such as songs and movies. Not all students are familiar with all peer-to-peer file sharing applications, but our experience shows that most of them had heard of tried using the original Napster and most of them are currently using one of the current file sharing systems, such as KaZaa (or other FastTrack network clients), BitTorrent or DirectConnect. To illustrate the legality issues in this topic, 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 in the US, Australia and elsewhere 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 may be using in their coursework, for example while writing term papers and using online sources as references.

## **2.6. 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. While it is impossible to provide a deep coverage of this topic in such a course, students must acquire the necessary skills and knowledge in computer and network security that is sufficient at least for their everyday computer use. 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. We discuss different issues of malicious software, such as viruses, worms, spyware and the importance of using an up-to-date antivirus. 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 email messages 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 development lifecycle. Most of the 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.

## **2.7. HTML and Web Design**

This and the following section of our course have a very special place in our course. Programming plays an integral role in computer science. Although HTML authoring is very far from writing program code, it is a step in the right direction. To many students with no technical background HTML authoring takes them much closer to programming because they can see how a small number of simple tags make a significant difference in the appearance of a web page. 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 paradigm and of many aspects of communication between web servers and client browsers. Many students are immediately interested in designing their home pages (which begins one of the assignment sequences offered in this course) 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 detailed in this paper. The author was truly impressed by the creativity combined with technical mastery exhibited by the works of several arts and humanities students who had no previous exposure to HTML or web design.

## 2.8. JavaScript and Basics of Programming

The course concludes with a study of elements of computer programming with JavaScript. JavaScript is not specifically designed to be the first programming language (like Pascal) and it may not have a simple and compact syntax (like QuickBasic); however, learning the basics of JavaScript programming presents a very easy and natural transition for the students who just learned how to design web pages with HTML. We do not attempt to cover all topics of coding with JavaScript. However, we discuss the most essential common programming aspects, 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-computer science majors. Usually, given the typical student body of our Introduction to Internet Programming and Applications course we try to steer away from any examples based on mathematics. However, students in our course are usually capable of writing small JavaScript programs that calculate GPA or total purchases on a shopping list.

## 3. Studying Student Motivation and Attitudes Towards Computer Science

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 the process of this study, which was partly influenced by the work of Forte and Guzdial [3], we offered a number of surveys to the enrolled students aimed at eliciting their perceptions and opinions about this course. Not all data

collected as a result of this survey is presented in this paper.

This course is a part of the general education requirements in most undergraduate programs offered at our school. In particular, students often have a choice to take this course or a mathematics course to satisfy their mathematics area skill requirements. However, in some of our academic programs students do not have such a choice. This explains a very high enrollment of humanities majors and a very low enrollment among students majoring in engineering and science, as illustrated in Figure 1. Many students, especially humanities and arts majors dread the idea of taking a mathematics course. For most of such students an introductory course in computer science appears as a safe haven and thus they make a decision to take the course that is in the focus of this paper. However, this means that these students are taking this course not because they want to learn more, but because they have to take it instead of the dreaded math or because they just need yet another course to satisfy their degree requirements. Many students in this group are much less motivated or interested in the course content than those who take this course intentionally, as illustrated further in this paper.

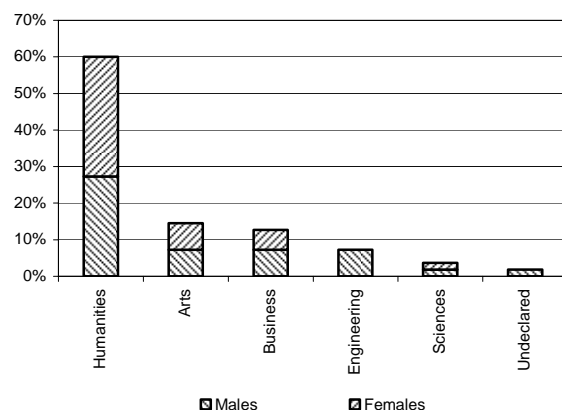
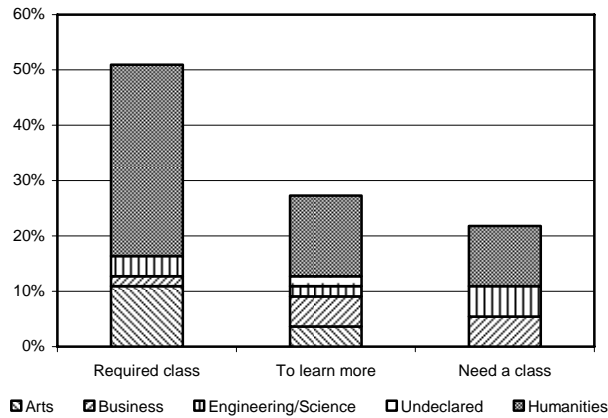


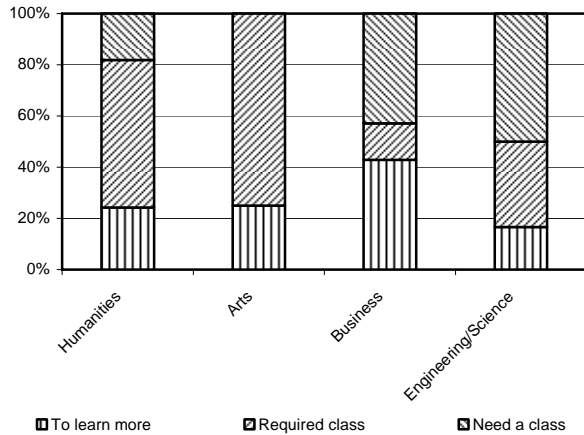
Figure 1. Course composition by student major.

In general, students majoring in arts and humanities say that they take this class because it is required by their program, but when they have a choice between a mathematics course and an introductory computer science course, they would gladly prefer the latter. This trend is demonstrated in Figure 2. Figure 2 a) shows the distribution of student majors broken down into the categories describing why students take this class. Over a half of all students enrolled in the Introduction to Internet Programming and Applications do so because they are required to take this or a mathematics course. Approximately 27 percent of all students enrolled into this course intentionally because they wanted to learn more about computers, while almost 22 percent of the

students enrolled because they simply needed some class to take. Figure 2 b) illustrates how the students' reasons for taking this course vary among the 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 in this course because they want to learn more about computers or because they just needed another class. It is worth noticing that we did not expect that a sizeable number of students majoring in humanities (approximately 35% of this major) enrolled in this course with an intention to learn more about computing.



a)

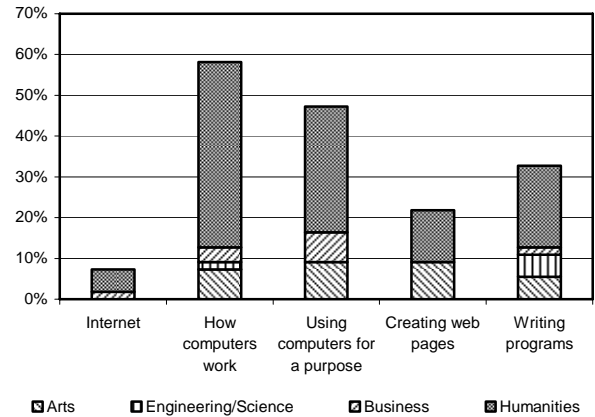


b)

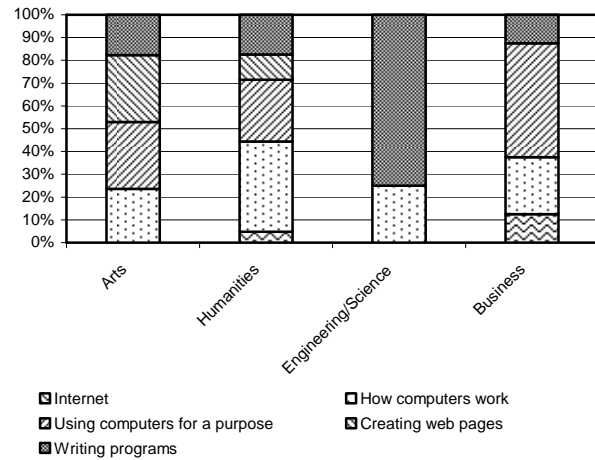
**Figure 2. Reasons why students enroll in course.**

One of the goals of our study was to learn about the student's perception of Computer Science. Better understanding the beliefs of students would help 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 a) represents the students' beliefs about what computer science is. Note that students were allowed to select several answers to this question so that

the percentages could sum up to over 100. It is not surprising that many students believe that computer science 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.



a)



b)

**Figure 3. What is Computer Science to you?**

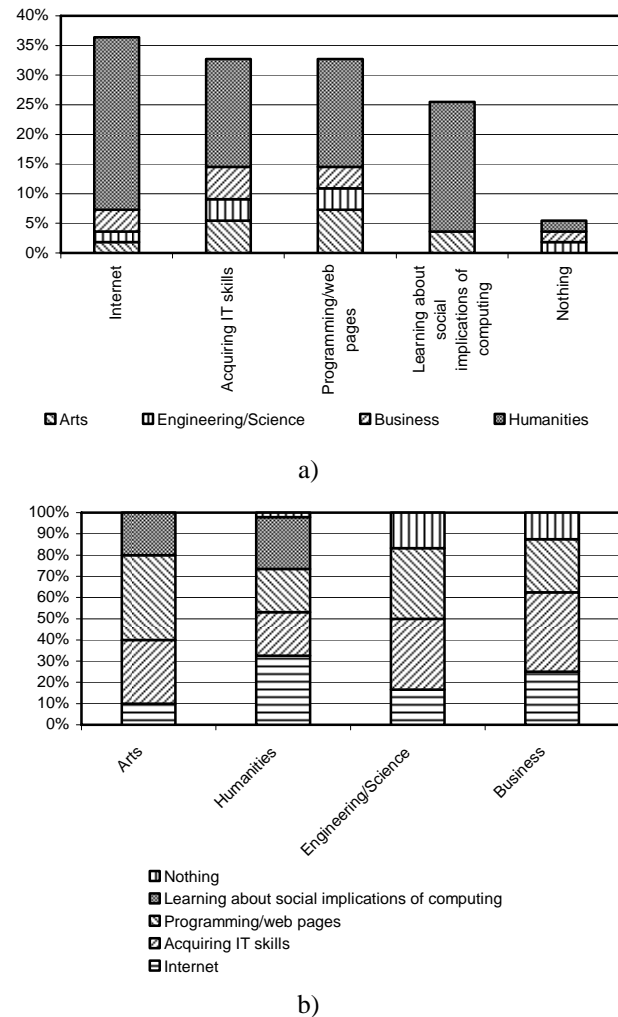
Figure 3 b) shows the distribution of the students' beliefs about the nature of computer science based on their major. Not surprising, engineering and science students were interested either in programming or in understanding how computers work. Business, arts and humanities students presented a more varying range of opinions indicating that most of them have no clear idea of what computer science truly is or what to expect from this course. For example, "writing programs" was the least popular answer among the humanities students when they were asked what is computer science to them.

Among business, arts and humanities students, over 60 percent believe that computer science 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 computer science. Our course “Introduction to Internet Programming and Applications” may be the only course they ever take in which students will have a chance to learn about this difference.

Helping students understand the true nature of computer science is one of the objectives of our course. 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 computer science 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 the recent technological advances, at least at the level of coverage of popular news sources on the Internet. As discussed in Section 2, a significant amount of time in our course is dedicated to studying the inner workings of the Internet and many network applications. We try to attract non-technically inclined students by teaching them how to design web pages, while others are attracted more by writing simple programs in JavaScript that run on these web pages. As we discuss many new developments and the impact of the Internet technology on our lives, it is impossible not to cover the 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 the student responses about what they like best about computer science 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 a) shows that among the 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 b) 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 b). 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. Summary

Our study indicates that it is extremely important to address many misconceptions common among non-majors about what computer science truly is. Educating student population about the nature of this subject area might help fight declining enrollments in computer science programs, although this alone will not be enough to bring the enrollments in computer science 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 Proceedings of 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 Proceedings of 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 Proceedings of SIGSCE 2001, pp. 184-188.