The Design Of A History of Computing Course With a Unique Perspective

Thomas J. Cortina*
Computer Science Department
Carnegie Mellon University
tcortina@cs.cmu.edu

Richard McKenna
Department of Computer Science
Stony Brook University (SUNY)
richard@cs.sunysb.edu

ABSTRACT

In this paper, we describe the design and implementation of a new history of computing course that includes personal and historical perspectives from faculty members to supplement the course material. Despite decreasing enrollments in our computer science courses, this new course has achieved significantly large enrollments and a wide audience due to this unique faculty perspective in addition to the approval of this course as a general education requirement that addresses the implications of science and technology on society.

Categories and Subject Descriptors

K.2 [**Computing Milieux**]: History of Computing – *hardware, people, software, systems, theory.*

K.3.2 [**Computing Milieux**]: Computers and Education – *computer science education, curriculum*.

General Terms

Human Factors, Legal Aspects.

Keywords

History of computing, Computer science education.

1. INTRODUCTION

Enrollment declines in computer science have captured the public's attention in recent years, and as a response to this trend, we explored offering new courses in computer science that would appeal to a wider audience. One such course developed in 2003 at Stony Brook University was CSE 301 History Of Computing. This course covers the fundamental milestones in computing from a number of perspectives, including software, hardware, computational theory and their legal and social effects on society.

Some computer history courses have already been described in recent literature. [4,8] Additionally, a task group has published a paper outlining the importance of computing history in a computer science curriculum and a set of sample outlines to guide faculty who wish to design and teach such a course. [7]

We designed a course that not only serves our computer science majors but also is accessible to students outside the major who are interested in how computers and computation evolved to what we have today. We built in a special feature to this course that has made it a very popular offering by asking computer science faculty at various levels of professorships to give seminars about their own experiences starting out in the field. These personal stories, along with films that document the massive changes in automated computing, make the course very interesting for not just computer science majors, but to a wide variety of students across the university.

This paper will describe the efforts made in putting such a course together. Then we will discuss how the faculty in the department played a key role in making this course a success. Afterwards, we'll summarize some key data about the course including enrollment information and student feedback. We will then close with additional resources and future directions to allow the reader to implement a course such as this at his or her own university.

2. BUILDING SUPPORT

Recent articles in the media have focused on the dramatic decline in enrollment in computer science programs across the United States. [5,10] At Stony Brook University, we have seen the number of our CS majors drop by approximately one-third in the span of just two years. This trend is echoed in a majority of major universities across the country. This drop caused us to examine our CS offerings closely, and we found that the addition of interdisciplinary courses could increase the participation in our courses and open some new areas of computer science to our own students. One such area that was missing in courses at our university was a course in the history of computing.

Traditionally, the history of computing course would be taught by the history department, but we designed the course so that we would be able to teach the course ourselves, giving it a personal perspective by adding in stories and recollections from current professors at various stages in their careers. However, to open this course to the widest number of students in a dense schedule of electives, we worked hard to design the course so that it would satisfy a general education requirement for the university. This made the course more than just a free elective now. Instead, students could use it to satisfy graduation requirements for the

*This paper describes work done by this author as a lecturer at Stony Brook University prior to June 2004.

SIGCSE'06, March 1-5, 2006, Houston, Texas, USA.
Copyright 2006 ACM 1-59593-259-3/06/0003…$5.00.

university, and surveys indicate that this has indeed helped attract students to this new course.

2.1 Course Objectives and Design Methodology

Our history of computing course has four important objectives:

- To study the historical, political, and social events that led to the development of modern computational devices.
- To examine the lives of some of the most influential inventors, thinkers, and entrepreneurs of the computer revolution.
- To understand how computer hardware and software evolved based on social and economic forces in the 20th century.
- To consider current social, legal and ethical issues in computing and determine the historical factors that led to these dilemmas.

One key aspect of a history course on computing is that there is a rich history for discussion before the development of ENIAC and the development of modern electronic computers. A critical decision was to start the discussion with the early uses of computation, going as far back as the abacus and the invention of number systems, specifically focusing on how the binary number system developed.

On the other hand, we conclude the course with a discussion of modern legal and social issues, discussing the reasons for a slew of new laws passed in the 1990s to deal with numerous issues magnified by the popularity of the Internet including privacy laws, child pornography, spam, and music file sharing. Although modern topics might not be considered "historical", we focus on tying these recent topics to earlier events that led to their importance today.

2.2 Course Topics

The course is organized based on time, starting from the very early days of computing to modern times. The course material centers on the key figures of computing, the major milestones in computer hardware development, the development of software components (assemblers, compilers, operating systems, etc.) and programming languages, and the social and legal effects of computers and software.

The lecture topics week by week (in order) are:

- The origins of computing: devices and numbering systems
- Computing in the 1800s
- Electromechanical computing and theoretical models in the early 1900s
- World War II and the advent of modern computing
- The dawn of commercial computing in the 1950s
- History of software and programming languages through 1980
- Mainframe and mini-computers: IBM & DEC in the 1960s
- The integrated circuit and other advances
- The microprocessor and the personal computer
- The GUI and the rise of Microsoft
- Advances in computer memory, computer languages and workstations
- The Internet and the World Wide Web
- Major legal and social issues of computing

Two books have been used as required texts for the course are by Campbell-Kelly and Aspray [2] and Ceruzzi [3]. The former book has a wider range of topics, starting in the 1800's with Babbage up to the early 1990's.[1] The latter book focuses on computing with more detail from World War II on to the later 1990's. Additional books are recommended for students who want to learn more about specific periods of time in computing history [1,6].

2.3 Student Assessment

A new course such as this requires students to do a significant amount of reading, so students are required to take a quiz at the start of each week to test their understanding of the assigned readings. Initially, students are very lax in their reading, but soon they find that they need to read each chapter more carefully to pick out the important points about computing history and leave out the irrelevant details just as one would do in any other history course.

In addition to quizzes on assigned readings and exams, students are required to write two term papers. The first paper involves a topic focused on the early history of computing up to World War II (e.g. Konrad Zuse's role in developing computers in Germany). The second paper involves a topic focused on more recent events (e.g. the battle between Microsoft and Apple, and others, over the GUI).

We faced several challenges in this part of the course as computer science teachers. Since a majority of our students were from computer science or other technical disciplines (see Section 4), we had to deal with poor student writing abilities. Fortunately, there is a university-level writing center available to help students improve their term papers, so we sent a number of our students to this center for assistance. Additionally, we discovered that we had to spend some significant class time teaching students how to do research for a term paper, since most of the students were quick to use Google to look everything up and simply pull text from the Internet. The library provided tutorials for us to explain to students how to perform research on the Internet, how to evaluate sources and determine credibility, and how to document such sources in the references.

Finally, an additional feature we incorporated into the course was a section on the midterm and final exams where students were given photos of famous people or things in computing history,

[1] A second edition of this text has been published in 2004 by Westview Press.

and students were asked to identify these. Since our course includes a lot of visual content including historical photographs in the readings along with videos and pictures in class slides, we felt that the exams should have some component that addresses this aspect of the coruse. At first, students were stunned that they would have to perform such an exercise, yet students (and we) were surprised at their good results when given the exams.

2.4 General Education criteria

In order to bring this course to as wide an audience as possible, a conscious effort was made to design the course in such a way that it would fulfill a general education requirement for the university. By doing so, students looked at this course as a required course rather than a free elective.

Our university has a Diversified Education Curriculum that specifies that students must take 1 or 2 courses in each of 11 areas to satisfy university requirements. One such D.E.C. category is “Implications of Science and Technology”, which states that

> “[These] courses are designed to help students understand the social and global implications of science and technology and to examine examples of the impact of science, culture, and society on one another.” [9]

Since this course is being offered by the computer science department in the College of Engineering and Applied Sciences and the D.E.C. designation is approved by officials in College of Arts & Sciences, some additional hurdles had to be overcome to get the course approved for general education credit.

By including an emphasis on the human aspect of computing (i.e. the inventors, company leaders and visionaries) and the inclusion of two term papers that require students to do significant research on topics that address the effect of computing progress on society (e.g. computing development during times of war, new laws and social change during the age of the Internet), officials in the College of Arts & Sciences approved the designation in November 2003 before registration began. This D.E.C. approval clearly helped boost the enrollments for the course (see Section 4).

3. ADDING ADDITIONAL PERSPECTIVES

The course includes an event once a week to break up the monotony of a traditional lecture format. The events were either films or documentaries about a specific period of time in the history of computing, or visits from faculty of the computer science department at various stages in their careers to give unique perspectives of the history of computing from when they first got involved with computers.

3.1 History from the eyes of faculty

In order for this course to succeed, we felt that we needed to present a unique perspective on the history of computing that we, as younger lecturers, couldn’t necessarily provide. To accomplish this, we asked members of the computer science faculty at various stages of their careers to come to visit the class for one hour to speak about the world of computing when they first got involved in academia or research.

By asking the Full Professors to speak earlier in the semester, Associate Professors to speak during the middle of the semester, and Assistant Professors to speak near the latter part of the semester, we are able to tie the presentations to the course material to the lecture material of the course. In each case, we discussed the particular topic and then had a faculty visitor who came shortly afterwards who gave more insight into the topic from his or her own perspective.

Here is a sample of some of the faculty lecture topics we used during the first three offerings of this course:

- Studying at MIT While Project Whirlwind Whirled By
- The U.S. Space Program and its Contribution to Computing
- Why Unix Should Have, but Didn’t, Win
- Using the WISC and Recollections of Alan Turing
- The History of Symbols in Computing

Our department has a large faculty with a wide variety of interests and ages, so providing a perspective from various eras in modern computing is feasible. For small departments, an alternative would be to invite members of industry to come to speak to the class at various times during the semester to coincide with particular portions of the course syllabus.

3.2 History from the eyes of the camera

In addition to the faculty who come to speak, the course includes a video presentation every other week or so that highlights some topic of the history of computing. The films are timed such that they coincide generally with the topics currently being discussed in class.

Here is a list of films used for most of the course offerings to date (listed alphabetically):

- "Biography: Bill Gates" (A&E Video)
- "Computer Pioneers" (ACM)
- "Creation of the Computer" (A&E Video)
- "Minerva's Machine: Women in Computing" (ACM)
- "Modern Marvels: Computers" (A&E Video)
- "Modern Marvels: The Internet: Behind The Web" (A&E Video)
- "Nerds 2.0.1: A Brief History of the Internet" (PBS Video)
- "Pioneer Computers" (ACM)
- "Triumph of the Nerds" (PBS Video)

This visual content provides yet another perspective on the issues students are reading about in their texts. A number of students have commented on how their views on the early pioneers (Atanasoff, Hopper, etc.) changed when they saw them speaking on film. They are especially charmed by Admiral Grace Hopper's story of the first computer bug which takes on a special meaning when she tells it herself (rather than an instructor telling the story).

4. DATA & FEEDBACK

Response to this new course was immediate and very positive. This section will outline the enrollment, major distribution, grade distribution and student feedback from our first offerings of the course at Stony Brook University.

4.1 Enrollment

The course was added to the university schedule in late Fall 2003, with no information in the undergraduate bulletin, yet students signed up in large numbers for this new course. Table 1 shows the enrollment in the course for the first four offerings. Note that this course was offered in a time of severely dwindling enrollments for computer science, so the overall interest in this course is very encouraging. Despite the large amount of reading and the term paper requirements, students still recommend this course to their peers semester after semester, so we feel we have a course that opens our discipline to a wide range of students and allows more students to visit the computer science department who normally might not come

Table 1. Number of students enrolled Spring 2004 – Fall 2005 (4 offerings)

Semester	Enrollment
Spring 2004	51
Fall 2004	36
Spring 2005	58
Fall 2005	44

One issue we had to deal with initially when this course was offered was that computer science students thought that this course could be used to satisfy a technical CS elective since this course had a 300-level number assigned to it. Even though our advertisements clearly indicated that this course was not technical in the traditional CS manner, some students signed up with this in mind. The numbers above, however, do not include these misinformed students, since they withdrew before the end of the registration period. Therefore, the numbers above indicate final enrollment of students after this issue was resolved, so we still had significant class sizes despite the restriction on CS majors.

4.2 Major Distribution

Although this course is taught in the computer science department, this course reaches a number of students outside of the computer science program, as shown in Table 2. Computer science students make up the largest group of students taking the class, but a wide variety of students from other majors have found this course interesting and relevant. The wide range of majors in a computer science course also has an effect on the computer science majors themselves, since most of their courses are limited to computer science and computer engineering majors. Our majors see that there is a great interest in their own field by students from distant majors, and we consider this beneficial.

4.3 Grade Distribution

The amount of course work is significant in this course, so the popularity of the course is not considered to be (solely) a function of the authors who have taught this course. Table 3 shows the grade distribution for the first three semesters, indicating that the performance of the students is quite good despite the fact that a number of students reported the workload of the course to be more than the average upper-level course. Also, we have a very low withdrawal rate from this class compared to the typical computer science courses we offer.

Table 2. Distribution of majors taking new course Spring 2004-Fall 2005 (4 offerings)

Major	Percentage
Computer Science and Information Systems	34.8%
Mathematics and Applied Math	13.4%
Social Sciences (Political, Psychology, Anthropology)	10.7%
Undeclared, Multidisciplinary Studies	9.6%
Engineering (Computer, Electrical, Mechanical, Biomedical)	8.6%
Business, Economics, Technological System Mgmt.	7.0%
Natural Sciences (Biochemistry, Biology, Chemistry, Earth/Oceanic, Physics)	6.4%
Humanities (English, Linguistics, Italian)	3.7%
Other majors	1.1%

Table 3. Distribution of majors taking new course Spring 2004-Spring 2005 (3 offerings)

Grade	Percentage
A's (A, A-)	34%
B's (B+, B, B-)	33%
C's (C+, C, C-)	14%
D's (D+, D)	5%
F's	8%
W (withdrawal)	6%

4.4 Some Student Feedback

At the end of each semester, students are given a standard university survey form to rate each course they take. On this form, there is a write-in question that asks: "Aside from the quality of the instructor, what is particularly good or bad about the course?" A sampling of student feedback is given below.

> "As a computer science major, I feel it was essential to learn its history and this course did it."
>
> "It was good to meet guest lecturer and watching video on Friday was fun."
>
> "Good for CS students as they learn history of their field."
>
> "Very general which allows students from different majors learn and enjoy computer history."

> "I liked that it was open to all majors because it was a fun way to satisfy a DEC that I needed."
>
> "The papers are too much to write."
>
> "Too much reading. There are 2 books, but one of them can be eliminated. The two books are too similar."

We did not list student feedback that was simplistic in nature (e.g. "It was good.") or that commented on teacher ability. In general, the feedback has been positive, although we have used the negative feedback to examine future possibilities for new texts and writing assignments.

5. SUMMARY & FUTURE DIRECTIONS

Offering a course like History of Computing in our computer science department has brought a new dimension to our course offerings and has been received favorably by students and administrators alike. By offering this course in our own department rather than the history department, we are able to add a personal perspective to the course from faculty who were around as particular events unfolded in computing history.

One student issue that we are addressing is finding a single textbook that covers the full range of historical periods that we discuss. Currently, the two texts used overlap in time, so students become frustrated when they have to read about some events twice. Additionally, students have complained about the writing assignments, yet we have found that this is one of their weakest abilities, so we will continue to include this in the course along with appropriate writing assistance from the university's writing center.

We would like to include industry leaders as future speakers for this course in addition to the faculty we have on campus. We feel this will give an added perspective to some of the more recent historical events that students currently study.

Readers who are interested in developing and running a course such as the one described here can contact the authors for more information and sample course material.

6. REFERENCES

[1] Akera, A. and Nebeker F. (eds.) *From 0 to 1: An Authoritative History of Modern Computing*, New York, NY: Oxford University Press, 2002.

[2] Campbell-Kelly, M. and Aspray, W. *Computer: A History of the Information Machine*, New York, NY: BasicBooks Press, 1996.

[3] Ceruzzi, P.E. *A History of Modern Computing, 2nd edition*, Cambridge, MA: The MIT Press, 2003.

[4] Eisenberg, M. Creating a computer science canon: a course of “classic” readings in computer science. In *Proceedings of the 34th SIGCSE technical symposium on computer science education*, pages 336-340, 2003.

[5] Foster, Andrea L. Student interest in computer science plummets. *The Chronicle of Higher Education*, Vol. 51, Issue 38 (May 27,2005), Page A31.

[6] Ifrah, G. *The Universal History of Computing: From The Abacus to the Quantum Computer*, New York, NY: John Wiley & Sons, 2001.

[7] Impagliazzo, J., Campbell-Kelly, M., Davies, G., Lee, J.A.N. and Williams, M.R. History in the computing curriculum IFIP TC3 and TC9 joint task group, *IEEE Annals of the History of Computing*, vol. 21, no. 1, pages 4-16, January-March 1999.

[8] Rupf, J.A. Teaching the history of computing (painlessly). In *Journal of Computing Sciences in Colleges: Proceedings of the 18th CCSC Southeastern Conference*, pages 212-218, December 2004.

[9] Stony Brook University 2005-2007 Undergraduate Bulletin, http://www.stonybrook.edu/ugrdbulletin/

[10] Vegso, J. Interest in CS as a major drops among incoming freshmen. *Computing Research News*, Vol. 17, No. 3, Page 6-1.