Teaching the non-science major: EEl0l – The most popular course at Yale

Roman Kuc
Department of Electrical Engineering
Yale University, New Haven, CT 06520-8284

EE 101 - The Digital Information Age, a course for non-science majors, is the largest course at Yale with an enrollment of more than 500 students. The goal of the course is to describe how common-place information systems work and why they work that way by illustrating clever engineering solutions to technical problems. The course considers the following topics: information sources, logic gates, computer hardware and software, measuring information using entropy, information coding and encryption, information transmission and information manipulation. EElOl includes a hardware and software project. For the hardware project each student implements a bean counter that counts a student-specific number of beans. The real success of the course is the software project that involves writing a personal World Wide Web page and developing a Web page for a Yale-affiliated organization. Having taken the course, students feel that they have an appreciation for the digital information artifacts they encounter on a daily basis. The joys and tribulations of teaching EElOl are discussed.

Introduction

The problems with teaching science and technology to the non-science major are well known [l, 2, 3,4]. The main problem is dealing with the wide spectrum of the student's experience in math and the sciences. A secondary problem is what to have the students do that is meaningful, instructive and satisfying. The solution to both problems that has found acceptance at Yale is to present material that the students find interesting and relevant, thus providing the motivation for expending the effort to learn the material.

EElOl is a course for non-science majors, as well as for freshmen are considering EE as a major. In addition to teaching students about electrical engineering, the student is invited to be an engineer for one semester: To think quantitatively, to design a simple digital system that does something useful and to develop pages on the World Wide Web.

The course attempts to teach technology in the least stressful manner to allow the *poets*, who would not normally have access to this material, to take the course. Among the difficulties with teaching a 100-level course are that there are no prerequisites and the course itself is not usually a prerequisite for follow-on courses. In a student's time and effort, it competes with courses in the major. Teaching such a course introduces a challenge to make the course accessible to the liberal arts major, while still making it interesting for the science major.

This paper describes the successes and shortcomings of the first two years of teaching this course. In the next section we give an overview of the course material. We then present the impressions of the students as provided on course evaluations, followed by the impressions of the instructor. The paper concludes with a short discussion on whether an EElOl-type course is appropriate for your school.

Overview of Course

The course consists of two 75-minute lectures weekly, optional recitation sessions and one laboratory session during the term. The lectures present the theory with demonstrations to illustrate the basic ideas behind many commonly-encountered information processing systems. The engineering approach to solving problems is illustrated by considering case studies from problem definition to final design. In a few cases, alternate designs are considered and the tradeoffs are discussed. The mathematical and physical principles are developed as needed to show how they are applied in practice.

Hardware and software projects provide a hands-on feature of the course. For the hardware project, each student implements a bean counter to count a student-specific number of beans. The software project involves writing a personal World Wide Web page and developing a Web page for a Yale-affiliated organization.

The course materials include a scientific calculator, a 300-page manuscript being prepared for publication and a 20-page pamphlet describing a step-by-step procedure on how to write a simple Web page on the Yale computer system.

The enrollment was equally distributed over the Freshman to Senior classes. Two graduate students and two high school students were also enrolled. Students came from various backgrounds, but predominantly from the liberal arts. A few computer science and physics majors who were interested in learning about applications also enrolled.

Lectures

The following topics are covered in the lectures, with each topic corresponding to a chapter in the manuscript:

1. Digital information sources. Information% defined as a quantity that is needed by a system to complete a task. The mechanical switch, familiar to all students, is treated as a source of binary information. An open pair of contacts represents a logical 1 and a closed pair a logical 0. Push-button, mercury and magnetic reed switches are discussed and used in applications including lighting a lamp when the car door opens, inflating an air bag in an accident and detecting motion.

Optical detectors are described as producing a switch closure when the detected light intensity exceeds a threshold. The inverse square law is presented to illustrate the limitations and as one method of locating a cellular telephone within a network of antennas. Beam-interrupt sensors are used to count objects. The same ideas are applied to optical communication channels for binary data transmission. Reflectance sensors are applied to bar code readers and camera autofocus systems.

2. Analog-to-digital conversion. Analog-to-digital conversion describes how analog information is transformed into a sequence of numbers. The binary number system is introduced. Sampling and quantization effects are illustrated using audio, image and video signals. The Nyquist criterion is explained and aliasing effects are demonstrated by undersampling a selection of popular tunes. Digital-to-analog conversion illustrates how analog information can be generated from a sequence of numbers, using children’s talking books as examples.

3. Digital logic. Combinatorial and sequential logic gates are introduced and used as building blocks to implement simple logic circuits. The basic combinatorial AND, OR and NOT gates and truth tables are used to design simple logic circuits, such as a circuit to recognize a particular binary pattern and the exclusive-OR (ExOR) gate. Set-reset and toggle flip-flops and a clock are the sequential logic gates that are considered. The timing diagram is used to illustrate their behavior. The logic gates are then combined to implement an elementary random-access memory and a modulo-N counter.

4. Computers. An overview of computer hardware and software is presented. The architecture of a typical desk-top computer is described and illustrated by disassembling a PC. Computer software is briefly described in terms of machine code instructions, operating systems, computer languages and application programs. Multi-processor computers are introduced as a solution to searching multiple data bases. The Smartcard is examined as an example of a elementary computer.

5. Measuring information. Basic probability theory is used to model information sources. The relative frequency approach is used to determine the probability of a particular symbol produced by a source. The source entropy equation for a source producing M symbols, $X_1, X_2, \ldots, X_M$ is given by

$$\mathcal{H} = -\sum_{i=1}^{M} P[X_i] \log, P[X_i] \text{ bits/symbol}$$

The entropy is used to determine the information content of a data file and the degree to which it can be compressed. The effective probabilities of symbols in a data file are computed by using relative frequencies. A pseudo-random number generator algorithm illustrates how randomness is generated by computer using modulo-N counting.

6. Information coding. Constant-length codes, variable-length codes and encryption are described. ASCII, credit card and bar codes illustrate constant length codes. Error detection and correction techniques used in postal and Universal Product Code (UPC)

bar codes are described. The Huffman coding procedure illustrates variable length coding with application to facsimile machine coding and data compression. Encryption is modeled as an ExOR operation on data with a pseudo-random sequence. Transmitting the encryption key securely using modulo-N arithmetic is illustrated.

7. Information transmission. Source data rate and channel capacity are defined and illustrated with examples. The bandwidth and signal-to-noise ratio are shown to determine channel capacity. One-way channels (IR remote control) and two-way channels (modems) are discussed and demonstrated. Simple computer network architectures are described. Data packet formats used in the Internet and the proposed asynchronous transfer mode (ATM) format are discussed.

8. Information manipulation. The computational complexity of solving a problem by computer is described. Cracking a computer password, game trees and the traveling salesman problem illustrate difficult problems. The idea of a suboptimal solution is introduced using Kruskal's method for obtaining a solution to the traveling salesman problem. Virtual reality techniques involving perspective, lighting and changing viewing perspective are discussed. Geometric principles are explained and applied to the making of Toy Story.

9. The course concludes with a discussion about the obvious advantages and disadvantages (loss of personal privacy) that can occur with unbridled expansion of information technology and the need for an informed citizenry.

Lectures were conducted in a 270-seat auditorium using overhead transparencies of the figures and examples in the text. An ethernet connection allows access to the Internet and a projection system displays the monitor image for examining Web pages. To accommodate the 500 students, two consecutive 75-minute sessions were held. Demonstrations of the principles were done on a lab bench in the front of the class using a camera and monitor system. Each lecture started by spending a few minutes viewing Web pages that had some novel features . Course-related announcements were posted on the EEl0l Web page (http: / / www.yale.edu / eel0l).

Students were assigned to one of ten recitation sections. These sessions met in the laboratory where the hardware project was also implemented. If students had questions about the lecture material they asked the teaching assistant (TA) during this time. The progress on the Web page project was monitored during the recitation section. In addition, an optional weekly recitation section was held to assist students with their Web page projects.

Having 500 students write Web pages required the full resources of the Yale Computing Services. Each student has a personal account. on the Yale computer system. The main uses of these accounts are email, browsing the Web, examining course syllabi and word processing. Yale College is organized into 12 residential colleges, with each college having PC and Macintosh computer facilities and a Computer Assistant (CA) to help students with basic computer questions. There are four additional computer clusters containing image scanning and printing facilities located around campus.

Projects

The course requires each student to do hardware and software projects.

For the hardware project, students were asked to design a bean counter using an infra-red detector, a chain of toggle flip-flops and whatever additional logic elements that were needed. Each student was assigned a number of beans to count between **17 and 56.** In preparation for the project, the student is asked to design two binary-counting systems on paper. The first system uses the conventional approach described in class that recognizes the full binary pattern representing their number. The second is a simplified version that takes advantage of the counting sequence and uses only the “1”s in the pattern. The student is asked to implement the simplified system for the project in the lab.

To implement the bean counter, students used logic modules. Each module is simple enough that its function is easily grasped. The input responds to a switch closure and the output is a switch closure provided by a reed relay. Circuits within the module were implemented using 3M circuit strips. Each module was powered with 4 rechargeable AA batteries. Light-emitting diodes were used to indicate power-on and logic levels. Each module is packaged in a clear plastic container, normally used for VHS video tape storage. Following block diagram convention, input terminals are located on the left side and output terminals on the right side. Interconnections between modules are made with standard audio cables.

To accommodate 500 students, 40 lab sections were distributed over the term in four lo-section sessions each lasting two weeks. The sessions started after the material necessary to complete the project was covered in class. Students could work individually or in pairs, although an individual report was required from each student.

For the software project, students were required to write a personal Web page and a page for a Yale-affiliated organization. Instructions on writing simple HTML code were described in a pamphlet. Most lectures started by examining random Web pages and their source codes to illustrate different methods of displaying information in an effective manner.

The personal Web page was to contain four components at a minimum:

1. an email tool,
2. a link to a file in the student’s directory,
3. a link to another Web page, and
4. an image in JPEG format.

Even with their varied backgrounds, all students were able to compose a satisfying Web page. Many found this to be an appealing project to show off their creative skills by implementing advanced techniques, such as novel background patterns, blinking displays, a counter indicating the number of hits, audio tracts and moving pictures.

The purposes of the organizational Web page were two-fold. The first was to give the student some experience with applying their new-found experience in a client/server context. The second was to bring university.entities that were not yet on the Web an opportunity to be exposed to it. Students were encouraged to approach their favorite organization (department, course, sport, fraternity/sorority, singing group, etc) and offer to write a Web page. A list of Yale organizations interested in having a Web page was also posted. The default organization was the EElOl course. Page development was typically done on the student's personal computer account. If the client was happy with the result, the code is transferred from the student account to the organization account for permanent accessibility.

Grading

The course grade was based on three exams taken during the term, the hardware project and the two pages on the Web. There was no curve: an A is earned with an average from 90 to 100, a B from 80 to 89, a C from 70 to 79, a D from 60 to 69 and an F for below 59. In fact, students in EElOl preferred not to have a curve, as they were afraid that a few gifted students or science majors would spoil the grade for the rest.

Three exams were given during the semester, with an optional final exam. If a student did badly in an exam, he/she could take it again during the finals period and the higher score would be used in computing the grade. This approach follows this instructor's view that a student should be able to work toward achieving a desired grade. Some exam questions were straight-forward, in order to determine if the student had mastered the basic ideas and math skills, while others probed their understanding of the concepts.

In the first two years of the course, the exams consisted of 9 questions and the students were graded on the best six answers. Nine TA's were each assigned a problem to grade, TA 10 tallied the best 6 marks and sorted the exams and TA 11 entered the scores in a computer and sent the grade to each student using email. This was completed in the evening of the exam over pizza and soda.

Impressions

To obtain student reaction to the course, a course evaluation form **was** appended to the last exam. About 90% of the students responded in some fashion, while about half provided comprehensive remarks. The impressions of the students can be summarized by the following.

1. About three-quarters of the class found the course worthwhile, feeling empowered with having gained some understanding about how information systems work. Several stated that it was the most worthwhile course they had taken at Yale

   With such a large class size, there will inevitably be students that are dissatisfied: Some will find the material overwhelming, others trivial. Being for many a first exposure to technology, there is material ,that all students found new and interesting.

2. The personal Web page project was an almost unanimous hit with the students. They appreciated the usefulness and timeliness of what they had learned in class.

3. The class met for two 75-minute sessions each week. This duration turned out to be too long to maintain the class attention, especially for the liberal arts majors. It was initially thought that examining the Web for ten minutes at the start of each class and having demonstrations would fill in the time, but it did not turn out that way.

4. Lecturing from overheads that were taken directly from the figures and examples in the book proved to be a failure: It lacked illustrations of the thought process that an engineer undergoes in examining a problem and reaching a design solution. It would have been much better to start with an empty black board and develop the problem and solution in class. Also, the unexpected result was that as many as two/thirds of the students stopped coming to some lectures, responding that the lectures did not help them in preparing for the tests.

5. The exam format was appreciated. The three exams allowed the student to study the material in reasonable chunks. Not having a formal final allowed students to devote more time to study for their other courses. Being graded on the best 6 out of the 9 questions allowed students to probe more deeply into an area that was personally interesting. The novel possibility of taking an exam covering the same material over again was appreciated. In most cases, it proved to be a means of having the students go back and study the material that they did not grasp during their initial exposure.

   Unfortunately, these considerations were abused by more than half the class, especially those taking the course for the credit/D/fail option. As soon at that critical C- level was reached, many students stopped coming to class. Their absence was certainly noticed by the more conscientious students and was a source of confusion.

   The students felt that the exams were too quantitative, testing mathematical skills rather than conceptual skills. With a class size of 500 it is difficult to grade exam questions that have other than numerical values. The search for more suitable questions continues.

After reviewing the course evaluation forms it became clear that there was a mechanism by which the students can pass the course without too much effort. By merely studying the examples and questions in the book, students could achieve a C- grade on the exam, or one that is sufficient for the Pass in the credit/D/fail option.

The impressions of the instructor can be summarized by the following:

1. The Web page project was an overwhelming success. Rather than merely observing the Web, this project allowed the student to be a participant. The open-ended nature of the project allowed students to work at any desired depth and experiment with newly-discovered features. Some students also developed pages for friends. All enjoyed the opportunity to have friends and family located in different parts of the world view their creations.

2. An EE instructor must resign himself to the fact that liberal arts and humanities students are different than engineering students, but not necessarily better or worst, just different. Unlike engineering students who faithfully attend every class, come on time and wait until the end of class to leave, liberal arts students tend to come and leave sporadically during the class. This instructor found this to be quite disconcerting, especially when a student leaves just before an important point is to be made. (Maybe the point was important only for the instructor?)

3. In a service course such as EE101, a detailed syllabus describing the topics covered in each lecture is important. In a standard engineering course, the syllabus is a list of topics that are to be covered and provides an overview. Non-science students use the syllabus to select the lectures that they need or want to attend.

4. Each student could elect to take the course for a letter grade or a Pass/D/Fail grade. About one-third of the students selected the latter. The students that cause the most frustration fall in three categories:

   (a) those that aim at a C- grade in order to achieve a Pass by expending the minimal amount of effort,

   (b) the students who, having attained the necessary number of points to achieve their desired grade, stop coming to class, and

   (c) those students who feel they have enough background to pass the course without attending the lectures. These include students who have had advanced math and science in high school. Unfortunately, their exam scores indicated that, in many cases, their background was insufficient.

Is EE101 for your school?

The question asked most often by publishers considering the manuscript for the course is: Is ***the course a success because of the material or the personality of the instructor?*** I feel it is a combination of the two, but the material covered is certainly timely and relevant. Teaching such a course will elicit different reactions at different universities depending on the student body. It will take a few teaching cycles to adapt the course to any particular setting. But, once it is established, it should become a popular course that meets the needs of the students who live in an increasingly technological age, and one that is satisfying to teach.

Acknowledgments

The equipment for lecture demonstrations was provided by a grant from the Paul Moore Memorial Fund at Yale.

References

[1] F.S. Barnes. Engineering education under attack. IEEE ***Transactions on Education, 37:1-2,*** 1994.

[2] J.R. Lohmann. Myths, facts and the future of U.S. engineering and science education. ***Engineering Education,*** pages 365-371, April 1991.

[3] S.B. Sample. Engineering education and the liberal arts tradition. ***IEEE Transactions on Education, 31:54-57,*** 1988.

[4] Morris A. Shamos. ***The Myth of Scientific Literacy.*** Rutgers University Press, New Brunswick, New Jersey, 1984.

ROMAN KUC received the BSEE in 1968 from the Illinois Institute of Technology, Chicago, and the PhD degree in Electrical Engineering in 1977 from Columbia University, New York, NY. From 1968 to 1975 he was a member of technical staff at Bell Laboratories engaged in the design of audio instrumentation and in the development of efficient speech coding techniques. In 1979, he joined the Department of Electrical Engineering, Yale University, where as the Director of the Intelligent Sensors Laboratory he is pursuing research in the solution of inverse problems and in sensors for robotics and biomedical applications.