Computer Science: A Core Discipline of Liberal Arts and Sciences

Robert E. Beck
Villanova University

Lillian N. Cassel
Villanova University

Richard H. Austing
University of Maryland

1 Introduction

The most consistent aspect of computer science is change. It is not surprising then, to note a significant change in the ability and interest of computer science departments to offer new courses for liberal arts and sciences students. With computer science enrollments finally stabilized and even declining, many departments find that they have the faculty resources to provide a selection of courses for non-computer science majors. At the same time that computer science faculty are free enough to look outside their discipline, a number of studies have expressed concern with the state of liberal arts and sciences education in American colleges and universities. Among the concerns expressed by a workshop sponsored by the American Association of Colleges (AAC) was that

> Scientific and technological developments have so outpaced the understanding of science provided by most college programs that we have become a people unable to comprehend the technology that we invent and unable to bring under control our capacity to violate the natural world [2].

The workshop also developed a list of nine experiences that should be integral to liberal arts and science education. Every academic discipline can contribute to these core areas. We will argue that computer science can be a significant contributor.

In this paper we review briefly the development of service courses within computer science departments, and consider a new approach to serving the needs of the student in liberal arts and sciences. Rather than considering which topics of computer science we feel are most useful to these students, we consider the experiences enunciated by AAC and identify computer science topics that contribute to each. In so doing we demonstrate the central position of computer science in the core of liberal arts and sciences, and indicate how courses could be developed that form a meaningful part of a bachelor's degree and are interesting and challenging for computer science faculty to teach.

We recognize that there is a continuing need for a number of types of courses to be offered as services by computer science departments. Though other types of these courses are also worthy of consideration, this paper deals only with the core material for the liberal arts and sciences bachelor's degree and the role of computer science in meeting the needs of these students.

2 Service Courses

2.1 The Role of Service Courses

For the purposes of our discussion we divide the courses offered by an undergraduate computer science department or program into two parts, major and service. Major courses, which satisfy requirements for students pursuing either a major or a minor in computer science, have been discussed at great length and over a period of at least twenty years in the various curriculum documents and articles published by the ACM and the IEEE Computer Society. The non-major or service courses taught by the department, are, in many cases, addressed to specific groups of students and are designed in cooperation with faculty from other departments or colleges of a university. Of particular concern to us are the service courses addressed to the liberal arts and sciences students.

Service courses for non-majors typically form the core of a liberal arts education and provide the breadth of knowledge that is a fundamental goal of a liberal arts college. These courses perform the functions of introducing the student to the central ideas of the discipline and providing an overview of the issues and problems which scholars in the discipline address. Service courses in the sciences also provide a laboratory experience in which students can follow the paradigm of scientific discovery by posing hypotheses and testing them under controlled conditions. Any service course, whether in the sciences or the humanities, must be designed with the realization that the students will take only one or two courses in the discipline.

These courses should also offer the experiences that are integral to a liberal arts and sciences education. One may not assume that the students have any particular knowledge of the discipline from their previous educational experiences. However, one should be able to assume students have a general sophistication and ability to think critically, to reason logically, and to synthesize ideas. Thus, it is important for service courses to present fundamental ideas and their interactions consistent with the goals of a liberal arts and sciences education.

2.2 Computer Science Service Courses

It is convenient to divide the computer science service courses into two categories. We put all the courses dealing with computational tools into the first category. These include courses about applications programs, such as "Using dBase III", courses describing a particular programming language, such as "Fortran for Engineers", and courses for the novice, such as "Your First Steps with MSDOS". In many cases, service courses in the first category are designed in cooperation with the discipline whose majors take the course. The role of these courses is to give the students a particular skill which they will find useful in the rest of their college and professional careers. The second category of courses contains those for which the assumption about minimal background knowledge is false. These courses may be offered to upper division students and will address specific areas where computing, either in design or in experimentation, has a strong influence on the discipline. Examples of such courses are "Data Communication for Scientists" and "Data Analysis and Evaluation in the Social Sciences". This is a largely untapped area in need of further development.

2.3 History of Computer Science Service Courses

Until 1962, however, all computing courses were service courses because no degree programs existed in computer science. The first service courses in computing were, appropriately, about computer hardware and software. They were much more technical and detailed than courses now are, because they had to be. Details were necessary to program in machine language, to wire boards and eventually to write assembly language code. With the advent of higher level languages (e.g. Fortran, COBOL), programming courses could concentrate more on language syntax and algorithm design and less on specific hardware details.

Curriculum '68 [4] included a section on the need for service courses, minors and continuing education. The report cited the estimate of the Pierce Report [1] that curricula of about 75% of all college undergraduates would be enhanced by computer training. Curriculum '68 suggested that the other 25% of students might also benefit. Curriculum '78 [5] cited the need for service courses in three categories: liberal arts or general university requirements, supporting work for majors in other disciplines, and continuing education.

Programming courses in specific languages have always been a major component of the service course offerings in computing. However, courses on societal impact became prevalent in the late 1970s and early 1980s as computer applications became more visible to the general public. These were followed by courses that featured hands-on experience in applications software as low cost personal computers became readily available. All of these courses were directed primarily to a general audience or to computer users. Some courses appeared that introduced students to the discipline of computer sciences [3]. These courses enabled students to find out about the discipline, often as a means to determine their interests in becoming a major.

It was clear to the developers of Curriculum '68 and Curriculum '78 that departments of computer science could offer important services to the entire academic community. Departments, to the extent that they could, provided service courses, but primarily of the kind that imparted facts and information with some hands-on experience. Very few of these courses required, or even encouraged, students to reason, analyze, understand, or evaluate. We were, and are, performing a service at the levels of knowing about and using computers, but not at the deeper levels that should characterize a person with a Liberal Arts and Sciences (LAS) education.

Departments of computer science need to do better, to provide a service to LAS students that is more in keeping with the goals of LAS. In a later section of this paper, we identify topics and areas within computer science that pertain to the goals. First, however, it is important to specify the kinds of experiences students should gain from a LAS education.

3 Liberal Arts and Sciences Education

In February, 1985, the Association of American Colleges published the report "Integrity in the College Curriculum: A Report to the Academic Community" [2]. The report resulted from a three-year study begun in January, 1982, as the Project on Redefining the Meaning and Purpose of Baccalaureate Degrees.

One of the sections of the report identified the following nine experiences as essential to a coherent undergraduate education, one that would enable graduates to fulfill "their promise as individual humans and their obligations as democratic citizens" [2, p.15]. The cited experiences are

1. inquiry, abstract logical thinking, critical analysis,
2. literacy: writing, reading, speaking, listening,
3. understanding numerical data,
4. historical consciousness,
5. science,
6. values,
7. art,
8. international and multicultural experiences, and
9. study in depth.

These experiences result from a program of study, not necessarily from any specific course. All disciplines combine with study in a major area to provide the experiences. Methods and processes, not just learning facts, provide the necessary experiences. For example, students should experience how scientists arrive at conclusions and what enables scientists to say the conclusions are accurate; students should not just be given the conclusions.

Meanings for many of the experiences are suggested by the titles given, but there is room for several interpretations. We provide a brief summary of the meanings in order to establish a context for contributions of computer science.

Inquiry, abstract logical thinking and critical analysis refer to the thinking process that enables humans to establish facts, combine facts to draw conclusions, to arrive at meaning, and to create ideas.

Literacy pertains to competence in writing, reading, speaking, and listening. One should be adept in these skills through knowledge and practice. Here competence represents a high level of attainment, not an overview or summary as is often implied in the use of the term "literacy".

Understanding numerical data requires the ability to interpret and represent numerical data as well as to recognize the misuse of data.

Historical consciousness involves comprehending present situations as a point in the evolution through time of a complex collection of thoughts, actions, and circumstances.

Science pertains to the recognition of what a scientific discipline is, and what are its processes. It requires understanding that its concepts and conclusions are human acts of intelligence and creativity, that it has inherent limitations, and that it impacts and informs other disciplines.

Values refers to the ability to make informed choices and assume responsibility for them as well as to understand one's own behavior.

Art includes appreciation and experience in both fine arts and performing arts, and knowledge of the language of these arts.

International and multicultural experiences lead to an understanding of how different people live and think and the reasons for these differences.

Study in depth implies a focused search, improving analytical capabilities and leading to more thorough perspective and knowledge of a topic or discipline over time.

4 Meeting the LAS Core Criteria with Computer Science Topics

Because there is renewed interest in providing a solid core for a liberal education we propose a new approach to the design of computer science service courses based on the fundamental elements of a liberal arts and sciences education. Computer science here serves as a vehicle to develop students' ability to analyze, synthesize and evaluate. For each of the AAC experiences we offer a selection of appropriate computer science topics.

Inquiry, abstract logical thinking, and critical analysis. This first criterion is the strongest contribution of computer science in the LAS core. The goals can be met by algorithm development and analysis, program design, implementation and testing, and all the methods of systems analysis and software engineering. The programming topic of data types, the representation of real or imagined items in a form that can be manipulated by computer, is an exercise in abstraction. In another approach, the introduction of abstract machines such as the Turing machine or Petri net contributes to meeting this goal. The idea of computability is a topic that is suitable here and also meets the desirable goal of introducing the limitations of computers as well as their power. Inquiry suggests also the topics of database and expert systems. In fact, there are few topics in computer science that would not contribute directly and significantly to meeting this criteria.

Literacy. Programming language development is a topic that explores the nature of communication including the acceptability of ambiguity and some errors in human communication and the need for precision and correctness in communication with machines. In addition, the necessity of expressing the needs and expectations of users of computer programs can be explored. Searching for a program to meet a particular need would be a valuable exercise in this context. In addition, in direct response to the AAC expectation, students should read the popular literature about computers, their developments and uses. They should read critically, looking for instances where the writer was not well-informed or where inaccurate or misleading information is included. Oral presentation of the results of this critical reading should be included whenever possible. In another approach to developing literacy in LAS students, a course may investigate ways in which word processing impacts the task of writing. The ease of making changes allows fine-tuning that would not otherwise be done. Perhaps it is now more feasible to write by filling in an outline. The advanatages that word processing offers some writers may handicap others by interfering with spontaneity.

Understanding numerical data. Number systems and the representation of decimal values in binary fit in here. The inability to represent simple decimal values such as one-tenth in binary gives rise to discussion of accuracy of information stored in the computer and the significance of roundoff errors. Such topics as the coding and transmission of data, pattern recognition, and image enhancement support this LAS experience. Methods used, reliability, and important application areas are appropriate subtopics. Graphic vs. tabular presentation of data should be illustrated and compared.

Historical consciousness. The development of computers and computer languages belongs in this area. In addition, the role of computer-based technology in human history should be included. The developing information age and changes in the location and nature of work are relevant.

Science. Methods in simulation and modelling are important here, as well as the use of these methods to conduct experiments that could not otherwise be performed. Uses of the scientific method in computer science belong here also, for example, the development and testing of heuristics where exact solutions are not feasible.

Values. Ethics issues relevant to the use and development of computers and computer systems provide a forum for discussion of values. Issues such as hackers, privacy, and the debate regarding SDI research funds provide examples of topic areas.

Art. The computer as a tool in the extension of human creativity should be explored. Some examples include the development of a language or representation system for dance, the growing dependence of theatres on computer assistance, the use of CADD systems in architecture, the question of the artistic merit of computer generated art and music, and the use of computers in film production.

International and multicultural experiences. Questions of transborder data flow arise here. Topics chosen should also include the use of computers in producing world models including limits-to-growth predictions, the importance of modern communication systems including computer-controlled telephone systems, and the growing use of electronic mail, the cooperative research efforts made possible by nearly instantaneous mail and file transfer facilities, the rapid reporting of news events around the world. Another relevant topic is the international competition for technological superiority. Cultural considerations such as the impact of work at home and the potential impact of computers in schools could also be explored here.

Depth of study. The use and misuse of computer science provide a wealth of topics suitable for in-depth study by LAS students. Individuals or small groups should research the impact of computers in a particular application area.

5 An Example

Section 4's simple listing of topics within each of the nine areas of experience illustrates the fact that computer science does contain material that substantially contributes to an LAS education, but does not indicate how suitable courses can be developed. It is possible to conclude from reading the collection of topics in the list that many colleges already offer appropriate courses. However, the common approaches to these courses are to cover many topics at an introductory level. We are suggesting a different approach.

Course design begins by selecting the experiences to be targeted. Several different courses with sets of experiences may be needed. These may be based on the disciplines of the students – some may have a greater need for emphasis on inquiry, abstract logical thinking, and critical analysis than they have for literacy, for example. Once the role of the course in the LAS education has been decided, appropriate computer science topics are selected, based on their relevance to the role chosen for this course. Many worthwhile topics will have to be left out of the course because the design goal is to *serve* the LAS education, not to teach computer science.

The following example illustrates the approach suggested above. We develop a module that emphasizes inquiry, abstract logical thinking, critical analysis, literacy, values and understanding numerical data for students in the social sciences. A project involving the interpretation of demographic data forms the base of the module. Students are taught to collect data and organize a database or are instructed to investigate existing databases to obtain the data they need. Class presentations and discussions focus on what a database is, how it is designed and developed, how user needs affect the result, criteria for selecting query languages and applications software, and issues of security, integrity, and privacy. Students conduct experiments with database software packages and integrated software, then report on the ease of use, clarity of documentation, advantages and disadvantages, what criteria determine when each should be used, ways in which data can be presented, and for a specific case what information can be obtained from given data (including accuracy and limitations). Students also investigate existing laws pertaining to privacy and databases or effectiveness (and limitations) of security measures. Class discussions include why the Data Encryption Standard was developed and how it works, characteristics of database languages and how they differ from or are the same as other programming languages, and case studies that illustrate the process of specification, design, implementation, testing, and verification. Laboratories include learning and practicing database querying to help understand capabilities and limitations of application software so that a comparative analysis can be made. A module of this kind could easily encompass most of a course, or it could be broken into submodules that could be covered in a shorter time. In either case, the selection, presentation, and assignment of material is done with the objective of providing students several of the nine experiences of an LAS education.

6 Conclusions and Recommendations

The nine experiences identified by the Association of American Colleges as fundamental to a liberal arts and sciences education form a framework for the design of service courses in computer science to be offered to liberal arts majors. Computer science can contribute in significant ways to all of the experiences. Consequently it can play a vital role as a discipline in the core of the liberal arts and sciences.

As colleges and universities examine their core requirements for liberal arts and sciences students, they should include computer science as a discipline whose study will meet many of the goals of liberal arts and sciences education. Courses should be constructed from the ideas we have given above to fit with the resources and needs of the individual school. Each of these courses should consist of a few topics, covered with care and depth so that they form a coherent whole from pieces that provide some of the fundamental experiences.

References

[1] *Computers in higher education.* President's Science Advisory Committee The White House. Washington,D.C., February 1967.

[2] *Integrity in the College Curriculum: A Report to the Academic Community.* Assoication of American Colleges, February 1985.

[3] Schneider G. Michael. A proposed redesign of the introductory service course in computer science. *SIGCSE Bulletin*, 18(4):15–21, December 1986.

[4] Curriculum Committee on Computer Science. Curriculum 68, recommendations for academic programs in computer science. *Communications of the ACM*, 11(3):151–197, March 1968.

[5] Curriculum Committee on Computer Science. Curriculum '78, recommendations for the undergraduate program in computer science. *Communications of the ACM*, 22(3):147–166, March 1979.