Creating a Computer Science Canon: a Course of "Classic" Readings in Computer Science

Michael Eisenberg
Department of Computer Science and Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0430
duck@cs.colorado.edu

Abstract

Computer science has a reputation of being a discipline in a perpetual state of accelerated progress—a discipline in which our techniques, our hardware, our software systems, and our literature rarely exhibit a staying power of more than several years. While undeniably exciting, this state of continual intellectual upheaval can leave computer science students (and faculty) with a disturbing sense that there is no essential core of great work within the discipline. This paper describes a readings course entitled "Computer Science: the Canon" whose purpose is to counter this perception by exploring a set of "great works" in computer science. We describe our own (undoubtedly idiosyncratic) reading list used for the course, and discuss several central issues involved in offering such a course within a computer science curriculum.

Categories & Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education, literacy.*

General Terms: Human Factors.

Keywords: Classic readings, canon.

1 Introduction: An Ephemeral Discipline?

Few academic disciplines are swifter-moving, more prone to recurring upheaval (or renewal), than computer science. Indeed, it can often seem that within a period of four or five years at most, a significant proportion of one's favorite subject matter—be it artificial intelligence, computer architecture, or programming language design—has been entirely revamped, with older references becoming obsolete and newer ones mandatory. For most computer science teachers, this is part of the challenge and allure of the subject; there is a true excitement in participating in what seems to be a pattern of near-continual intellectual revolution.

SIGCSE '03, February 19-23, 2003, Reno, Nevada, USA.

At the same time, even teachers of computer science must occasionally pause for breath; and in our reflective moments, it is possible to wonder whether there is anything *not* ephemeral in our field—any body of work to serve as an emotional contrast to a tone of perpetual metamorphosis. One might of course identify certain relatively long-lived core ideas within computer science, but the question here is focused on our discipline's literature. Is there any writing in today's curriculum that could conceivably be worth rereading in a generation's time—or anything written a generation ago that is worth rereading today? Or, to put the matter another way: is there anything that we would deem as an essential core of great literature within computer science—a set of works that every "serious" student of the subject should have read?

This paper describes a course that reflects an (ongoing) attempt to answer this question. The semester-long course—entitled "Computer Science: the Canon"—has been offered on four separate occasions at the University of Colorado (and will almost certainly be offered again within the next two years). The Canon—as it will be referred to in the remainder of this paper—is a "great-readings" course, structured primarily as a discussion seminar, and offered to both undergraduates and graduate-level students. The purpose of the course is to give computer science students a chance to encounter, explore, and discuss works that might arguably be said to have achieved a certain degree of longevity and importance.

In the following section of this paper we present a table summarizing the reading list that has been used for the Canon, and discuss (from the perspective of the teacher) several recurring issues in offering a course of this sort within a department of Computer Science. The third section presents—in all-too-encapsulated form!—a number of reflections on the individual works themselves, and how they have been received by our students. In the fourth and final section, we discuss several plausible ways in which a "Canon-like" course might be integrated within a broader computer science or engineering curriculum.

Table 1: The Computer Science Canon—a (Suggested) Reading List

Author	Title and Year	Publisher or Source
Aiken, Howard	Proposed Automatic Calculating Machine [1937]	Reprinted in Cohen, Welch, and Campbell (eds.), *Makin' Numbers: Howard Aiken and the Computer.* Cambridge: MIT Press, 1999.
Ada Augusta, Countess of Lovelace	Notes on the Analytical Engine [1842; accompanying a paper by L. Menabrea]	Reprinted in P. and E. Morrison (eds.), *Charles Babbage and his Calculating Engines*. New York: Dover, 1961,
Babbage, Charles	Of the Analytical Engine [1864]	*Passages from the Life of a Philosopher*. (Reprinted by Rutgers University Press, M. Campbell-Kelly (ed.), 1994.)
Backus, J.	Can Programming be Liberated from the Von Neumann Style? A Functional Style and Its Algebra of Programs [1978]	*Communications of the ACM*, 21: 8
Berkeley, E.C.	*Giant Brains, or Machines that Think* [1949]	New York: John Wiley.
Boole, George	*An Investigation of the Laws of Thought* [1854]	Reprinted by New York: Dover Press.
Bush, Vannevar	As We May Think [1945]	Reprinted in Nyce, J. and Kahn, P. (eds.) *From Memex to Hypertext*. San Diego: Academic Press, 1991.
Brooks, Frederick	*The Mythical Man-Month* [1975]	Reading, MA: Addison-Wesley.
Clarke, Arthur C.	*2001: A Space Odyssey* [1968]	Reprinted by New York: New American Library.
Dijkstra, E.	Go To Statement Considered Harmful [1968] Solution of a Problem in Concurrent Programming Control [1965]	Reprinted in *Communications of the ACM*, 26:1 (Jan. 1983)
Feynman, R.	There's Plenty of Room at the Bottom [1959]	In A. Hey (ed.), *Feynman and Computation*, Reading, MA: Perseus Books, 1999.
Gardner, M.	*Scientific American* columns on John Conway's game of Life (approx. 1970)	Reprinted in *Wheels, Life, and other Mathematical Amusements.* New York: W. H. Freeman 1985.
Gödel, K.	*On Formally Undecidable Propositions of Principia Mathematica and Related Systems* [1931]	Reprinted by New York: Dover Press.
Karp, R.	Reducibility among Combinatorial Problems [1972]	In R.E. Miller and J. W. Thatcher, eds. *Complexity of Computer Computations*. NY: Plenum Press, pp. 85-104.
Leibniz, G.	On His Calculating Machine [1685]	In Smith, D. E. (ed.), *A Source Book of Mathematics*, New York: Dover Books, 1959.
Licklider, J. C. R.	Man-Computer Symbiosis [1960], and The Computer as a Communication Device [1968]	*IRE Transactions on Human Factors in Electronics*, HFE-1: 4–11, March. *Science and Technology*, April.
McCulloch, W. and Pitts, W.	A logical calculus of the ideas immanent in nervous activity [1943]	Originally in *Bulletin of Mathematical Biophysics*. Reprinted in *Embodiments of Mind*. Cambridge, MA: MIT Press, 1965.
Minsky, M. and Papert, S.	*Perceptrons* [1969]	Cambridge, MA: MIT Press.
Naur, et al.	Report on the Algorithmic Language Algol 60 [1960]	*Communications of the ACM*, 3:5
Papert, S.	*Mindstorms* [1980]	Cambridge, MA: MIT Press.
Pascal, B.	On His Calculating Machine [c. 1650]	In Smith, D. E. (ed.), *A Source Book of Mathematics*, New York: Dover Books, 1959.

Table 1: The Computer Science Canon—a (Suggested) Reading List (continued)		
Author	**Title and Year**	**Publisher or Source**
Shannon, C.	The Mathematical Theory of Communication [1948]	Reprinted in Shannon, C. and Weaver, W., *The Mathematical Theory of Communication*. University of Illinois Press, 1963.
Simon, H.	*The Sciences of the Artificial* [1969/1981]	Cambridge, MA: MIT Press. (2nd edition)
Sutherland, I.	Sketchpad: a Man-Machine Graphical Communication System [1963]	Reprinted in Glinert, E. (ed.) *Visual Programming Environments: Paradigms and Systems*. Los Alamitos, CA: IEEE Computer Science Press, 1990, pp. 198-215.
Tarjan, R.	Depth-first search and linear graph algorithms. [1972]	*SIAM Journal on Computing*, 1(2):146-160, June 1972.
Turing, A.	On Computable Numbers, with an Application to the Entscheidungsproblem [1937]	Originally in *Proceedings of the London Mathematical Society*, v.42. Reprinted in Davis (ed.), *The Undecidable*, New York: Raven Press, 1965.
	Computing Machinery and Intelligence [1950]	Originally in *Mind*. Reprinted in Collins and Smith (eds.) *Readings in Cognitive Science*, San Mateo, CA: Morgan Kaufmann, 1988.
Von Neumann, J.	First Draft of a Report on the EDVAC [1945]	Reprinted (in part) in Randell, ed., *The Origins of Digital Computers: Selected Papers*. New York: Springer-Verlag, 1973.
	The General and Logical Theory of Automata [1951]	Reprinted in J. Newman (ed.), *The World of Mathematics*. New York: Simon and Schuster, 1956.
Wiener, N.	*Cybernetics* [1948/1961]	Cambridge, MA: MIT Press, 1962,

2 The Canon: Reading List and Course Design

Before proceeding to a discussion of the course itself, it seems worthwhile to present the reading list that we have employed. Since no two semesters have used exactly the same set of readings, Table 1 above includes what is actually a "composite" reading list, comprising most of the papers and books that have appeared in at least one iteration of the course (where possible, accompanied by anthologies or relatively recent sources in which these materials may be found).

In any such list, the particular choices will inevitably be idiosyncratic, and must therefore be put forward in a spirit of caution. (The reader has already, no doubt, rejected some of the choices here and has proposed at least a half dozen new ones of his or her own.) Nonetheless, the entries listed in Table 1 seem to be at least a plausible set of "canonical" readings; at least, in the course of a decade's worth of conversation on the subject with colleagues, none of them has been a consistent and obvious target for removal.

With Table 1 serving, then, as a representative set of choices for a great-readings course, we can discuss a variety of recurring issues involved in offering a course of this type.

The Pedagogical Role of Great Readings. It should be emphasized at the outset that, for students, a course of classic readings has a very different purpose than (say) a course organized around either a set of current research papers or a textbook. In particular—and here the comparison with the textbook is most apt—classic readings are only rarely finely-polished works of pedagogical exposition. That is, a researcher or writer in the process of thinking through important ideas is not, as a rule, in the position of future writers, who can expend their energies on devising better explanations of (by then) well-understood ideas. Thus, one does not (for instance) read Turing's 1937 paper to arrive at the easiest explanation of the Universal Turing Machine; one does not read McCulloch and Pitts to experience one's first explanation of neural nets; and one doesn't read Boole to learn Boolean algebra. (By the same token, in the realm of physics, one does not generally read Newton to learn Newtonian mechanics.) The more compelling reason for reading these authors is to see examples of creative minds at work in the course of exploring new intellectual territory—or, perhaps, to better understand works that had particular historical importance. In some lucky cases, this will also afford the student an encounter with marvelous writing (Lovelace, Brooks, and Papert come especially to mind); but in general, computer science students should not expect to gain a cogent or complete introduction to technical material from a classic.

Preparation and Background of Students (or, Who Should Take Such a Course?). As something of a corollary to the previous paragraph, it is likely that a beginning computer science student will not have the preparation or breadth of perspective needed to enjoy a classic-readings course. (A student new to computational complexity, for example, will simply have too much difficulty with Karp's paper to derive much pleasure from it.) In the case of our own course, there is no hard-and-fast prerequisite (except for a general degree of "mathematical sophistication and fearlessness"); but undergraduate students not yet in their senior (or perhaps junior) year of a computer science major are discouraged, though not absolutely forbidden, from taking the course.

Comparison with a History of Computing Course. A course of this sort clearly affords a substantial overlap with the material that might be included in a "history of computing" course. Advanced computer science students interested in the history of

their discipline will no doubt derive tremendous value from viewing our course as an introduction to critical historical documents. At the same time, we do not characterize The Canon as a history-of-computing course (and unfortunately, our department does not offer such a course); the "lens" of great papers is not, in our view, the ideal one through which to view all the larger patterns in computer science history. For example, the development of data tabulation machines (by Hollerith and others) in the late nineteenth and early twentieth centuries represents an important chapter in the history of computing; but this is a chapter that is not (to our knowledge) associated with any particularly provocative or compelling original writing. More generally, many crucial strides in the history of computing were made by means of the introduction of commercial products; and while these products deserve a central place in a discussion of history, they generally have not provided an occasion for great contemporary computer science literature. Our own feeling is that a history-of-computing course and a great-readings course would, in combination, make for an excellent two-semester sequence for advanced students; we'll return to this suggestion at the end of this paper.

Guidelines for Selecting Readings. The readings listed in Table 1 are admittedly somewhat skewed toward a particular view of computer science—a view in which topics such as artificial intelligence and human-computer interaction have a prominent place. To some extent, the readings are also skewed away from those works that could be described as summaries of some piece of software (e.g., language manuals—though the Algol 60 report is an exception); this stems from the (possibly unjustified) intuition that such documents make for relatively slow reading and desultory discussion. Another aspect of the readings shown is their age: for the most part, we have hewed to a rule that Canon papers should be at least 20 years old to be included in the course. The purpose of this rather arbitrary cut-off point is to provide a rough means of ensuring the lasting interest of the work in question. A pleasurable side-effect of this rule has been that, within each iteration of the course, several new works become "eligible for inclusion": the Papert and Simon books, for instance, were not included in the first iteration of the course, but have since come to be added to the reading list.

Inclusiveness and Diversity. As followers of academic debates undoubtedly know, within the humanities there have been impassioned—and in our view, often fruitful—arguments over the selection of, and even the value of, "great works" lists in disciplines such as literature and philosophy. (See, for instance, Denby[6] and Atlas[1] for readable discussions of these debates.) Indeed, the title of our own course is a playful allusion to the disputes over "canonical" reading lists in these disciplines. One dimension of these debates focuses on the lack of diversity exhibited by traditional selections of classics—a pattern of selection that disproportionately (and counterproductively) favors "dead white males."

As it happens, the list of readings in Table 1 is heavily populated by dead white European and American males (though there are also some living white males and a dead white female within the list). Our feelings about this are frankly ambivalent. On the one hand, Table 1 reflects our best professional judgment about the central intellectual documents of computer science; and as such, we do not apologize for its construction. On the other hand, the fact that the list is so demographically skewed reflects (in our view) broad historical inequities in professional and academic opportunity. Our hope is that—if a "Canon-like" course is taught in succeeding generations—the reading list for future versions of the course will come to reflect far greater diversity in gender, ethnic background, and geography.

Related and Background Readings. Our experience to date in offering The Canon suggests that the students have enough reading to do in the course without assigning them still more. As a consequence, we do not generally require the students to read supplementary background or biographical materials. To some degree, this also reflects an intuition that "great works" should be encountered by each individual student without too much outside interpretation or influence as provided by additional readings. (One consistent exception to this rule has been the suggestion that students read the excellent *Gödel's Proof* [10] by Nagel and Newman as a supplement to the paper by Gödel—the original paper itself can be quite daunting to students without a background in mathematical logic.)

As it happens, there are a number of good full-length biographies of "canonical" authors—including books devoted to Turing [9], Von Neumann [2], Babbage [13], Lovelace[15], Bush [16], Gödel [5], Licklider [14], Feynman [7], and Aiken [4]. There are other fine sources of biographical material for McCulloch and Wiener [8], Shannon [12], and Boole[3]; while [11] includes chapters based on interviews with Backus, Brooks, Dijkstra, and Tarjan. Again, to keep the students' reading list manageable, and to focus attention on the works themselves, such background readings are not assigned within the course—but they are extremely useful for the professor! Moreover, these readings may also serve as the basis for extended reading and final reports.

3 Reflections and Brief Comments on Individual Readings

While space does not allow even a cursory discussion here of the works themselves, it is perhaps worthwhile—in the interest of those faculty members who may wish to pursue similar efforts—to summarize our experiences of student reactions to several of the readings. In general, we have noted at least a few consistent patterns among these reactions.

Perennial Favorites: Lovelace, Brooks, Turing. What seems to set these authors apart for the students is the clarity and grace of their writing combined with the importance of their subject matter. The response to Countess Ada's writing is often particularly strong, since students are delighted that so many central themes of twentieth-century computer science were anticipated a century earlier by Lovelace and Babbage in the course of working on Babbage's Analytical Engine.

Perennial Least Favorites: Boole, McCulloch and Pitts. Much of the difficulty (and student disappointment) in reading these works stems from the perceived opacity of the writing (in Boole's case) and notation (in that of McCulloch and Pitts). Typically, the professorial role in discussing these works is not to defend them as unassailable gems of perfection, but rather to highlight their creativity and historical importance.

Difficult (but Positively Regarded) Papers: Gödel, Shannon, Feynman. Even advanced computer science students are often unfamiliar with (or at least less than expert in) mathematical logic, information theory, and the physics of computation. Whether this is a defensible state of affairs is the subject of another discussion; but it does require additional effort for the professor in teaching this material—usually, at least part of the class discussion is given over to lecturing so that the students can appreciate the ideas in these papers.

Pure Fun: Berkeley, Gardner, Clarke. These are readings aimed at a popular audience, and everyone (including the professor) has a grand time with them. The book by Berkeley (a founding member of the ACM) is a special treat, as it provides a portrait of how computers were presented to the general public at a time when very few actual computers were in existence.

4 Conclusions from the Teacher's Perspective

In our own department, The Canon is regarded as an "enrichment" course for advanced students; but faculty members interested in instituting a similar course at other institutions may find an opportunity to integrate such a course either with a broader curriculum in the history of computing (as noted earlier), or with other curricular themes. One intriguing possibility might be to pair a Canon-like course with a survey course expressly focusing on futuristic ideas in computer science (including readings in, e.g., quantum and biological computing, spintronics, interfaces to ubiquitous computing, and so forth).

Still another approach to extending The Canon would be to look beyond the disciplinary boundaries of computer science, and to view our course as an initial (and rather specific) attempt to weave together the traditional formats and concerns of both engineering and liberal arts curricula. Thus one might imagine, for instance, a College of Engineering developing a historical "great readings" course in the broader areas of engineering or technology, aimed at students who wish to situate their study of technology within the larger context of the history of ideas. Such a course would most likely include at least a few of the readings in The Canon, though (as we've imagined it here) this broader course would probably also be aimed at a more general student audience, and might well steer away from including several of the most technically challenging Canon readings.

In any event, as noted earlier, our own course is still very much a work-in-progress; future versions of The Canon are likely to continue the process of experimenting with the reading list, adding a few works here and subtracting others there. This experimentation is part of what has kept the course "fresh" from the faculty standpoint; but in truth, the readings are for the most part so strong that they reward both repeated reading and repeated discussion over multiple semesters.

References

[1] Atlas, J. *Battle of the Books*. [1992] New York: Norton.

[2] Aspray, W. [1990] *John Von Neumann and the Origins of Modern Computing*. Cambridge, MA: MIT Press.

[3] Bell, E. T. [1937] *Men of Mathematics*. New York: Simon and Schuster.

[4] Cohen, I. B. [2000] *Howard Aiken: Portrait of a Computer Pioneer*. Cambridge, MA: MIT Press.

[5] Dawson, J. [1997] *Logical Dilemmas: the Life and Work of Kurt Gödel*. Natick, MA: Peters.

[6] Denby, D. [1996] *Great Books*. New York: Simon and Schuster.

[7] Gleick, J.. [1993] *Genius: the Life and Science of Richard Feynman*. New York: Vintage Books.

[8] Heims, S. J. [1991] *The Cybernetics Group*. Cambridge, MA: MIT Press.

[9] Hodges, A. [2000] *Alan Turing: the Enigma*. New York: Walker and Company.

[10] Nagel, E. and Newman, J. [2002] *Gödel's Proof*. (Revised Edition) New York: NYU Press.

[11] Shasha, D. and Lazere, C. [1995] *Out of Their Minds*. New York: Springer-Verlag.

[12] Sloane, N. and Wyner, A. (eds.) [1993] *Claude Shannon: Collected Papers*. New York: IEEE Press.

[13] Swade, D. [2001] *The Difference Engine*. New York: Viking Press.

[14] Waldrop, M. [2001] *The Dream Machine*. New York: Viking Press.

[15] Woolley, B. [1999] *The Bride of Science*. New York: McGraw-Hill.

[16] Zachary, G. [1997] *Endless Frontier*. New York: Free Press.