

Computer Science in a Liberal Arts Context

JOSH TENENBERG

University of Washington, Tacoma

and

ROBERT McCARTNEY

University of Connecticut

This issue is devoted to the curriculum guidelines from the Liberal Arts Computer Science Consortium. These guidelines provide a coherent and important model for computing education within a liberal arts context, giving primacy to critical reason, rigorous methods, and student engagement in the research process. In this regard, they are at the same time both conservative, by giving low priority to the latest technologies, and radical by resisting the managerial impulse that at its worst commodifies and devalues higher education.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Curriculum, computer science education*

General Terms: Documentation, Standardization

Additional Key Words and Phrases: Computing curriculum guidelines, LACS, liberal arts

ACM Reference Format:

Tenenberg, J. and McCartney, R. 2007. Computer science in a liberal arts context. *ACM J. Educ. Resour. Comput.* 7, 2, Article 1 (June 2007), 3 pages. DOI = 10.1145/1240200.1240201 <http://doi.acm.org/10.1145/1240200.1240201>

We are pleased to publish “A 2007 Model Curriculum for a Liberal Arts Degree in Computer Science” by members of the Liberal Arts Computer Science Consortium (LACS). This article represents the combined effort of many of the leading teacher/scholars in computing education at liberal arts universities. It updates earlier model curricula published by this group in 1996 [Walker and Schneider 1996] and 1986 [Gibbs and Tucker 1986] and reflects changes to computing and computing education over the last decade.

Authors' addresses: Josh Tenenberg, Institute of Technology, University of Washington, Tacoma, 1900 Commerce St., Tacoma, WA 98402; email: jtenenbg@u.washington.edu; Robert McCartney, Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269-2155; email: robert@engr.uconn.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2007 ACM 1531-4278/2007/06-ART1 \$5.00 DOI 10.1145/1240200.1240201 <http://doi.acm.org/10.1145/1240200.1240201>

We believe that it is important for this curriculum model to have wide dissemination. Curriculum guidelines embed deep knowledge about the structure of the discipline. They embed as well philosophies about what should be taught and what should be left out, what are the basic entities that comprise the discipline, and what are the dominating themes and problems. In this sense, curriculum guidelines serve as much to define a disciplinary paradigm as to reflect a paradigm as it currently exists. Given the effort involved and the expertise represented, these guidelines would be difficult, if not impossible for any individual or department to carry out. The authors present these guidelines not as a strict canon that must be literally adhered to, but as a basis on which others can adapt to suit their local settings. We also are supportive of an *alternative* model of computing education from that of the ACM and IEEE-CS [The Joint Task Force on Computing Curricula 2001]; in an increasing global and pluralistic world, having a number of thoughtful and coherent models for computing education will be vital to maintain the vitality of the discipline.

There are two characteristics of these guidelines worthy of special note. The first is the fitness of these guidelines to the constraints under which Computer Science is most often (though not universally) taught in a liberal arts university: small class sizes, a small number of full-time faculty to teach the entire curriculum, a limited number of courses within the major, and research as something that students and faculty jointly undertake as members of a community of scholars.

The other key characteristic is the way the guidelines embed important aspects of the liberal arts tradition. Such values give primacy to critical reason and rational discourse, to formal and fundamental principles that have stood (and presumably will continue to stand) the test of time. The guidelines were generated over a 5-year period, are the product of many conversations and critical review, and do not radically rewrite previous versions. In this regard, they have a conservative character: one does not see here a celebration of the latest technologies or a knee-jerk response to industrial imperatives.

However, as Jackson Lears reminds us [Lears 2003], the liberal arts tradition also has a radical character, and one that is increasingly important in a world in which students are often viewed as “product” to be moved through an educational “pipeline”:

The attempt to turn universities into businesses challenges the conservative understanding of the humanities. If the liberal arts tradition is understood as a worldview, rather than a collection of courses, it poses a radical challenge to the managerial impulse . . .

Though one might disagree with the choices made in the guidelines, or believe that they are inappropriate for other contexts, there is much to admire in guidelines that embed a humanistic spirit giving primacy to critical reason, rigorous methods, and student engagement in the research process as a fundamental aspect of learning. We recommend this article to all of our readers.

REFERENCES

- GIBBS, N. E. AND TUCKER, A. B. 1986. A model curriculum for a liberal arts degree in computer science. *Communications of the ACM* 29, 3, 202–210.
- LEARS, J. JANUARY-FEBRUARY 2003. The radicalism of the liberal arts tradition. *Academe* 89, 1, 23–29.
- THE JOINT TASK FORCE ON COMPUTING CURRICULA. 2001. Computing curricula 2001. *J. Educ. Resour. Comput. I*, 3.
- WALKER, H. M. AND SCHNEIDER, G. M. 1996. A revised model curriculum for a liberal arts degree in computer science. *Communications of the ACM* 39, 12, 85–95.

Received April 2007; accepted April 2007