COMPUTER SCIENCE IS ALIVE AND WELL ON PLANET EARTH

F. Garnett Walters

University of Missouri at Rolla

The time has arrived to respond to the forecasters of gloom and doom with respect to the decline of computer science and the questions concerning the viability of computer science as a discipline. The soothsayers, who continue to publish remarks to the effect that:

a. there is no need for the study of computer science, and
b. the graduates in computer science do not satisfy the needs of industry

need to be answered.

An example of the type of article I have in mind is one by Jim Leeke, entitled "Computer Grads: Do They Make the Grade?", from *PC Week*. His premise is that industry wants personnel with a business background and interpersonal skills. In addition, he embraces the belief that current graduates in computer science do not possess these skills. His conclusion is that industry wants a liberal–arts graduate because they are much more educated and more adaptable.

I will not take the time to attempt to defend the curriculum as it is currently being taught at many institutions across the nation. I would like to attempt to defend the curriculum as it was proposed in Curriculum '78 [2] and modified in succeeding years (Koffman [5] and Koffman [6]). A Task Force on the Core of Computer Science [10], chaired by Peter J. Denning, has proposed a new look at the discipline named Computer Science. This should probably be referred to as the "science of computing". Curriculum for a liberal arts degree was proposed by Gibbs and Tucker [7] while a curriculum with mathematics as the basis was proposed by Berztiss [8]. Knuth [1], Ralston and Shaw [3] and Ralston [4] wrote concerning the relationship between mathematics and computer science. The one exception in the defense of curriculum in computer science would be the lack of strong evidence to support computer science as a mathematical discipline. Too many schools have taken Curriculum '78 literally, and to not require the mathematical maturity necessary to support the discipline. These schools are graduating a product which may be productive in a particular working field but may be limited in their advancement in the field. I describe these graduates as "processors of data" and not as computer scientists. The best way to describe the difference between these two terms is by an example.

The problem at hand is to sort a given set of numbers in descending order. No restriction on the sorting routine. A sample set of data is provided to demonstrate that the solution does, in fact, satisfy the problem statement. The data is: 175, 100, 225, 25, 150, 50, 125, 200, and 75. The computer scientist chooses an appropriate algorithm and codes the algorithm in his favorite language, tests the program on this set of data and others of his choice and proceeds to the next assignment. The "processor" of data looks at the set of sample data, observes that each member of the data set is divisible by 25, does the division, observes that the set is now just the set of positive integers from 1 to 9, and outputs these numbers as the results. Because this is the correct result for this set of data, he codes the solution in his favorite language, tests the program on this set of data and proceeds to the next assignment.

It is my belief that the graduates who are described in the article by Jim Leeke are closely related to the "processors of data" and should not be called computer scientists. The example is sited of a company looking for a new DP trainee and after interviewing, the most sought after person was a 27–year–old woman with a master's degree in library science with some experience. I would take exception to the comparison of a 27–year–old applicant with a master's degree and some years of experience, in a field with a high profile using communication skills, to a new graduate with a bachelor's degree and no experience. I would certainly expect the additional education obtained with a master's degree, the maturity of a 27–year–old and the years of work experience to cause some managers to rate the applicant higher than a new graduate. I have serious doubts about the validity of the scientific analysis in this comparison. I would challenge the managers to compare a 27–year–old applicant with a master's degree in computer science and several years experience to the applicant described above. The thinking person knows the results of that comparison.

Before anyone thinks that I am not in touch with the main stream of what industry wants, let me defend the need for graduates with better people skills and a working knowledge of the business world. The two items which add the most to a technically qualified graduate are speaking skills and writing skills. At some point in the life of an employee with a company, they will be required to write a concise, to the point report, for management. If the report is acceptable, then an oral presentation will need to be made. No one is interested in reading endless pages or listening for hours to a presentation which could be done in a much shorter time.

It is time that we took a long hard look at what we are calling computer science at our colleges and universities. To successfully do this, we need a definition of computer science. I prefer one put forth by Peter J. Denning [9], which says:

> "Computer science is the body of knowledge dealing with the design, analysis, implementation, efficiency, and application of processes that transform information."

Many other definitions have been proposed and most will come close to this statement. Computer science has deep roots in mathematics and logic. As we examine the programs at our colleges and universities, we must distinguish the non–mathematical based programs from the mathematical based programs. A heated discussion will follow about who gets to retain the name computer science. I do not believe that what we call a program has anything to do with its validity as a discipline. It may take several years before a name is accepted and causes confidence in the graduates. My preference would be to use the name computer science for the mathematical based programs and use names such as Information Systems, Data Processing, Systems Analysis, Business Data Processing, Computer and Information Systems, and Systems Engineering, etc., for all other programs. The exception would be that a program in Computer Engineering could be a separate and distinct mathematical based program. Given that a definition has been accepted for the discipline called Computer Science and that this name is applied to the mathematical based programs, then what message do we, as professionals in the field, send to industry and to our students. I believe the proper message is:

Computer Science

a) is a mathematical based discipline
b) is more than just programming
c) is more than just processing data
d) is a systematic study of algorithms and data structures
e) requires knowledge about operating systems

f) requires knowledge about data abstraction
g) requires knowledge about databases
h) requires knowledge about data structures
i) requires knowledge about file processing
j) requires knowledge about expert systems.

Certainly we can insure the literature we produce for distribution to prospective students contains this information. Recruiting students to a program which is different than advertised borders on being dishonest. With the current lack of uniformity in what a name means, we are probably all guilty, to some extent, of "false advertising". I have been told of a school which advertises a degree in computer science which has only two faculty members. One faculty member has an MBA while the other holds a BS degree in Business. I have severe reservations about the ability of that department to offer a quality degree program. However, the graduates of that program may serve the needs of the local clientele, and therefore, serve a useful purpose. Students enrolling in our programs deserve to know where they may find employment upon graduation. It is incumbent upon the recruiters for industry to be fully aware of the scope of the programs where they recruit employees to work as computer scientists. Recruiting employees at schools with weak programs and then complaining that the employees do not perform at an acceptable level is unethical. Not everyone who completes a degree at a quality institution will produce at an acceptable level, however, there is a better chance of success for this employee.

The question of how to send this message is more difficult to answer. I believe the message can be sent to the prospective student through the process of accreditation by the Computer Science Accreditation Board (CSAB). Once a program has been accredited, then both students and industry can have confidence in the program. The prospective student will know about the quality of the program before entering the program and industry will know about the educational background of the graduates from that program.

The other method to insure that the students and industry are properly informed is by carefully choosing the names of our programs. To accomplish this will require an agreement on which program will be given which name. I seriously doubt that any agreement can be reached on the topic of names for programs. The problem with names of programs will probably be solved by a higher authority, in our case in Missouri, the Coordinating Board for Higher Education.

How can we keep computer science alive and well in Missouri?

The first step is to address the problems before we have a solution dictated from above. Accreditation is easy to address by making a formal request to CSAB for an accreditation visit [9]. It will require some time to prepare for the visit. However, the problem with names will probably require a statewide meeting, of those interested, to begin discussions on how to classify, evaluate and name existing or proposed programs in the state. As professionals, in the field of computer science, we must exert our influence to see that our profession is well monitored.

References

1. D. E. Knuth, Computer Science and Its Relation to Mathematics, *The American Mathematical Monthly* 81, April 1974, 323–343.
2. ACM Curriculum Committee on Computer Science. Curriculum '78 – Recommendations for the undergraduate program in Computer Science. *Communications of the ACM* 22, 3, March 1979, 147–166.
3. A. Ralston and M. Shaw, Curriculum '78 – Is Computer Science Really that Unmathematical?, *Communications of the ACM* 23, 2, February 1980, 67–70.
4. A. Ralston, Computer Science, Mathematics, and the Undergraduate Curriculum in Both, *The American Mathematical Monthly* 88, August–September 1981, 472–485.
5. E. B. Koffman, P. I. Miller, and C. E. Wardle, Recommended Curriculum for CSC 1, 1984: A report of the ACM Curriculum Task Force for CSC 1. *Communications of the ACM* 27, 10, October 1984, 998–1001.
6. E. B. Koffman, D. Stemple, and C. E. Wardle, Recommended Curriculum for CSC 2, 1984: A report of the ACM Curriculum Task Force for CSC 2. *Communications of the ACM* 28, 8, August 1985, 815–818.
7. N. E. Gibbs and A. B. Tucker, A Model Curriculum for a Liberal Arts Degree in Computer Science, *Communications of the ACM* 29, 3, March 1986, 202–220.
8. A. Berztiss, A Mathematical Focused Curriculum for Computer Science, *Communications of the ACM* 30, 5, May 1987,

356–365.

9. T. Booth and R. E. Miller, Computer Science Program Accreditation: The First–Year Activities of the Computing Sciences Accreditation Board, *Communications of the ACM* 30, 5, May 1987, 376–388.

10. Task Force on the Core of Computing Science, Computing as a Discipline, *Communications of the ACM* 32, 1, January 1989, 9–23.