Pro Rege

Volume 10 | Number 1 Article 2

September 1981

Computer Science: Brave New Field

Gerald Hoekema
Dordt College

Computer Science: Brave New Field

Gerald Hoekema
Associate Professor of Mathematics
and Computer Science

Gerald Hoekema has taught at Dordt College for the past ten years. Before that he taught at Montreat-Anderson College in North Carolina. He received an M.S. degree in Mathematics from Purdue University and has spent a number of summers at the University of Iowa studying Computer Science. He will be on leave of absence for the 1981-82 academic year and will be spending the year teaching at Calvin College, Grand Rapids, Michigan.

Computer science is relatively new on the academic scene. This is a natural consequence of the fact that the computer itself has only been with us for a short time. The rapid growth of computer technology is well documented and clearly evident even to those who have a limited knowledge of computers. Fantastic increases in computer power, together with similar decreases in size and cost, have occurred at a startling pace. New applications for the microprocessor (the "computer on a chip") seem to bombard us almost daily. The possibilities of growth and for employment in this burgeoning industry appear to be unlimited.

What does all this astounding technological explosion have to do with an academic institution, particularly a Reformed, Christian academic institution? It has much to do with such an institution, particularly when it may well have a profound influence on the whole sphere of education itself. To ignore this phenomenon and hope that it will eventually "go away and leave us alone" is not only unrealistic but also unreformed. We, as Reformed Christians, strongly believe that every area of life must be brought under the Lordship of Jesus Christ, that the mandate to subdue the earth encompasses the whole of creation. Obviously then, computer science can and must become part of the encyclopedia of knowledge to which Christians address themselves. The need for Christian guidance

in the field of computer science is especially urgent because of the growing influence the computer has on all our lives. From all appearances, this influence is not likely to decrease as time goes on, but will continue to grow at an ever-increasing rate.

As must be obvious by now, I feel firmly that Reformed colleges must come to grips with this relative newcomer, this computer science. The time is ripe for a concentrated effort to be made on the part of Christians not only to develop a unique perspective in this field, but also to exercise leadership here. Because of the youth of this field, Reformed colleges have an opportunity to give it direction and claim it for Christ now, rather than to try to reform it at some later date when much of its development will already have become fixed. It is now time to sieze the day in this individual science and to formulate guidelines for the study and application of computers in society.

Unfortunately, many of us feel somewhat hampered. I do not feel that I can speak as an expert in this area since my training is not primarily in this field. When I attended college, there weren't any courses available in computer science. Only in recent years have Reformed colleges begun to offer majors in computer science, and Dordt College is just now beginning to look into this academic area. To become trained to teach computer science I had to take all my computer science training at a secular university. We already have some catching up to do, but I believe we can and must do so.

If we are to begin to provide leadership in this field, it is essential that we equip our students to assume a leadership role. Therefore, we must offer an integrated curriculum that enables our students to cope with this technology. This has ramifications in many areas. For example, a large number of Dordt College students are preparing for careers in elementary and secondary education. Since the computer is being used more and more (at least in the secular world) in education (computer-assisted education, computer-managed remedial courses, motivational computer graphics, etc.), our students must be prepared not only to use competently such techniques and equipment, but also to develop these materials from a Christian point of view. Programmers with a Christian perspective must be trained in our schools to take their places in business and industry to witness there. Systems analysts, computer designers, and electrical engineers with a commitment to Christ and his kingdom must come from the Reformed colleges.

As must be obvious by now, I feel firmly that Reformed colleges must come to grips with this relative newcomer, this computer science. The time is ripe for a concentrated effort to be made on the part of Christians not only to develop a unique perspective in this field, but also to exercise leadership here.

All students, those who will be directly involved in the computer industry and those who will only be peripherally involved, should be adequately trained by our colleges. Having been primarily engaged in the teaching of computer programming I am naturally led to think first of all in terms of the advantages all students can gain from instruction in this particular area of computer science. While programming is not necessarily an easy subject area for all students, there are many benefits that can be gained by students who conscientiously attempt to learn computer programming.

The first benefit is that of increased problem-solving ability. In order for the computer to perform any activity at all, a

series of instructions or a computer program must be written to inform the computer of exactly what must be done and the order in which it must perform these operations. The programmer must decide what the instructions and order of operations must be if the computer is to transform a given data set (the input values) into a desired result (the output values). This involves a thorough understanding of the problem, a realization of what constitutes an acceptable solution to the problem, an algorithm or step-by-step solution method for the problem, and a translation of the algorithm into a higher-level computer language. A student must have or develop a fairly keen analytic ability in order to solve problems with a computer competently. Undisciplined thinking does not, generally, lead to good solution methods. The ability to solve problems (even those that will not be solved using a computer) is an especially desirable ability, and it is one that is sadly lacking in many of our students. This analytic ability is certainly one of those qualities we wish to see in people who call themselves educated.

A second benefit gained by people who learn to program well concerns logical, orderly thinking. The step-by-step instructions given to the computer by way of a computer program must be complete and must necessarily be ordered in a meaningful way. These steps need to be given as a sequence of instructions, so that the execution of one step, followed by the execution of the second step, and so on down to the last step, will provide proper output for the problem. The order of the performed operations (the "flow of control") is as important as what occurs at each step, and this involves a well-thought-out, logical arrangement of the instructions. The interrelatedness of the steps and of the independently performed procedures involved (separate, subordinate program units which are called into action by the main program itself) is also extremely important and must be taken into account by the programmer. Each step has some relevance to the total outcome of the program, and its placement in the sequence of program instructions is very important. Changing its location by only a single statement may drastically alter what the program will do. Students must come to see the relation between the whole and its parts, and they must understand the importance of the order of operations if they are successfully to instruct computers to solve problems.

The ability to solve problems (even those that will not be solved using a computer) is an especially desirable ability, and it is one that is sadly lacking in many of our students. This analytic ability is certainly one of those qualities we wish to see in people who call themselves educated.

A third benefit can be that learning a computer language and instructing computers in such a language can improve the student's communication skills. The inherent ambiguities of everyday language and the meanings derived from context and so forth certainly can lead to a richness of expression and beautiful ways of communicating. However, these ambiguities can also lead to confusion and misunderstanding. What one person says might be quite different from what another person perceives him to say. (This is why mathematicians insist on very rigid, single-meaning definitions of the terms of an axiomatic system.) Ambiguity might lead to totally different interpretations of a given system. Computer languages, while not allowing for richness of expression, certainly must be unambiguous. Higher-level computer languages (such as Pascal and ALGOL) must first be processed by a com-

piler (or translator) before a given program is accessible to the computer (which "thinks" only in terms of 0's and 1's, not in terms of words or phrases). If any ambiguity were present in the language, the compiler might translate a statement one way today and another way tomorrow. This would mean a given program would function in two (or more) entirely different ways. You can imagine what this would mean in something like a payroll processing program! This, of course, cannot be allowed. Languages for computers necessarily have precise syntactical rules, and the grammar is detailed and exact. In many courses, students soon learn that they can gloss over a misunderstood point in papers by verbiage that is lengthy but essentially says nothing. This is not possible in a computer program. Each step must be completely and unambiguously spelled out or an error occurs. One cannot take a solution part way and then wave his hands wildly, hoping the computer will fill in the unknown details. No, each step must be explicitly stated in accordance with a rather strict set of communication rules. Unless this necessity for completeness, precision and detail becomes a part of the student, he or she will not be very successful in writing programs.

All of the above are more or less peripheral benefits that can come from the study of computer programming. But the ability to program obviously has some more direct benefits as well. The student who can program well has a much brighter future in today's job market than non-programmers. Teachers will be much better prepared to cope with computers in the classroom if they have this ability. The computer is a tool that can help man in countless ways. It is already used as a labor and time saving device to free people from many mundane, clerical tasks, enabling them to be more productive. As a scientist, I see one of its more important roles to be related to scientific research. The computer makes "impossibly" long calculations possible. It allows for experiments to be performed rapidly and repeatedly (through a technique called simulation) without waiting days or weeks for results and without actually using the expensive chemicals, white rats, etc., that would have been necessary in an actual experiment. We can simulate dangerous experiments on the computer without worrying about the deleterious effects on the environment that would accompany the experiment itself. Not only should the study of computers come under the cultural mandate, but computers themselves can also be of inestimable value to us in carrying out God's cultural mandate.

The student who can program well has a much brighter future in today's job market than non-programmers. Teachers will be much better prepared to cope with computers in the classroom if they have this ability. The computer is a tool that can help man in countless ways.

Reformed colleges must begin to train students who will eventually become experts in this field. Society needs Christians who can provide leadership in this crucial, life-affecting area. It is also up to Reformed colleges to equip the general student effectively to become masters of these "new-fangled gadgets" that will be intruding into everyday life increasingly as time goes on. This discipline will help them gain logical, disciplined thought processes and problem-solving abilities. It is my hope that this paper might act as a springboard for further action along these lines.