Inservice Education of High School Computer Science Teachers

James Kiper[*] Bill Rouse[†]
Douglas Troy[‡]

[*] Miami University, commons-admin@lib.muohio.edu
[†] Miami University, commons-admin@lib.muohio.edu
[‡] Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE
& SYSTEMS ANALYSIS

TECHNICAL REPORT: MU-SEAS-CSA-1988-008

Inservice Education of High School Computer Science Teachers

James D. Kiper, Bill Rouse, and Douglas Troy

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

Inservice Education of
High School Computer Science Teachers

by

James D. Kiper
Systems Analysis
Miami University
Oxford, Ohio 45056

Bill Rouse
Teacher Education
Miami University
Oxford, Ohio 45056

Douglas Troy
Systems Analysis
Miami University
Oxford, Ohio 45056

Working Paper #88-008 09/88

INSERVICE EDUCATION OF HIGH SCHOOL COMPUTER SCIENCE TEACHERS

James Kiper, Department of Systems Analysis
Bill Rouse, Department of Teacher Education
Douglas Troy, Department of Systems Analysis
Miami University, Oxford, Ohio

1. Introduction

The United States urgently needs strong high school computer science programs. Our economy and the technological level of our society depend upon a continuing supply of high quality scientists, mathematicians, engineers, business people, computer scientists, and technicians, all equipped with a solid foundation in computer science. This supply depends upon a strong computer science program that begins in high school.

Effective skills in computer based problem solving and computer programming are essential for today's college students who major in science, mathematics, engineering, business, and computer science. Although many entering college freshmen have had some experience with microcomputers and BASIC programming, their high school courses have largely ignored structured programming and problem solving and they lack the skills required to use computers effectively in the study of these disciplines. Many enter college with poor programming and problem solving habits that are difficult to break. Because of this condition, colleges must spend the freshman year teaching the fundamentals of structured programming, problem solving, data structures, and program design, which serve as a foundation in computer programming for these students. These fundamentals, that now constitute the course of study for the freshman year in college, should be taught at the high school level, and they can be taught there, provided the high school teachers have appropriate training.

The primary ingredient of a strong high school program is a cadre of highly qualified classroom teachers. Such teachers are needed in order to produce college-bound graduates prepared to enter rigorous programs in science, mathematics, engineering, and computer science and to produce non-college-bound graduates prepared for technical training and entrance into the workforce. Such teachers are needed to provide professional leadership in curriculum development and program implementation.

At a minimum, the computer science expertise of entry level teachers should approximate that of entry level practitioners. Not only is this level of expertise necessary to enable teachers to develop the type of high school curriculum that we have indicated, but it is important for instilling self confidence on the part of the teachers and for developing a reputation of respect on the part of parents and the general public.

2. Problem Definition.

In 1985, an interdepartmental committee on Computer Science Education was organized between the Department of Teacher Education and the Department of Systems Analysis. This action resulted in the establishment of a Project Planning Group composed of representatives of local public and private school systems, local industry, and faculty members from both University departments. A conference with chief administrative officers of local school systems and a questionnaire circulated among computer science teachers indicated a serious need to strengthen high school computer science programs. The questionnaires were mailed to 192 high schools in southwestern Ohio and a response of precisely 50% was obtained. Of the 96 schools responding al but 4 offered some form of computer science course. Virtually every school responding taught BASIC in some form or other. However, based upon analysis of the textbooks used, only 18% were introducing their students to structured programming and logical problem solving. Only 17% were offering an advanced course to prepare students for the College Board's Advanced Placement Examination in computer science. Approximately 85% of the respondents indicated that at least part of their knowledge of computer science was obtained through self-teaching. We believe that the situation in southwestern Ohio is fairly typical of conditions across the United States in general.

As of July 1987, the State Department of Teacher Education and Certification in Columbus has been implementing newly created standards for certification of computer science teachers. With respect to the subject matter component, the standards merely specify that an approved program must contain at least 30 semester hours of work in the content of the subject, with the actual choice of courses determined by the institution, subject to review by a visiting State team every five years. Thus, the selection of computer science courses for the teacher training program at Miami University was left to its faculty. Therefore, consistent with the philosophy expressed above, we included in the program for computer science teachers approximately 75% of the course work required in the program for computer science practitioners.

At the time we initiated the program we recognized that it would be at least four years before the first of these well prepared graduates would be ready to enter the profession, and several years after that before the program would grow to the point of producing graduates in any significant number. In the meantime not only would there be an absence of well trained computer science teachers to offer strong high school programs, but there would be an absence of well trained computer science teachers to help us train these new student teachers. Teachers with strong computer science competence are needed to serve as exemplary role models for teacher candidates and to supervise their clinical training and student teaching assignments. We were faced with a situation in which the student teachers would know more computer science than their supervising teachers.

To make matters worse, the State Department of Education had elected to "grandfather" all high school teachers who happened to be teaching a computer programming course as of January 1987. This resulted in a situation in which certain teachers who had minimal formal training (e.g., a one week workshop at a local computer store) were granted full computer science certification because they were teaching a programming class in BASIC, while other teachers who had completed several substantive college level computer science courses were denied such certification because in January 1987 their teaching assignment happened to be temporarily composed entirely of mathematics courses. Thus there were many high school teachers who possessed computer science certification, but as far as we knew, none who had preparation and knowledge comparable to that specified by our new teacher education program.

However, there was a core of these certified teachers in southwestern Ohio who were personally committed to the improvement of computer science in their schools and who had been attempting over the years to improve their competence by means of workshops, courses, and self study. These teachers recognized the need for stronger programs as well as the shortcomings of their formal college coursework, which was completed before computer science courses were available to teachers. They needed and desired an organized program of inservice education in order to attain their full potential as computer science teachers.

The Project Planning Group developed a proposal to the Ohio Board of Regents for a grant of money to Miami University from funds from the federal Economic Security Education Act to support a program named The High School Computer Science Enhancement Project. The project was funded by the Board in the amount of $35,320 with additional support of $59,965 in cost sharing funds from Miami University, the General Electric Company, and several local public and private school systems. The grant provided for the tuition, textbooks, and mileage reimbursement for up to 30 high school computer science teachers to attend a series of four courses in computer science and computer science education at Miami University.

3. Purpose.

The goal of the Project Planning Group was to enhance computer science programs in southwestern Ohio by providing inservice education to high school teachers who had been granted full teaching certification in computer science by means of "grandfathering", and yet lacked the level of training in computer science that is required of future computer science teachers who are now in training. It was expected that such inservice education would result in:

1. An immediate benefit to high school students who are now in school and who cannot wait until preservice teachers are graduated from college and certified by the State.

2. A long range benefit to high school students of the future who will have the opportunity to study in a high school environment of expanded and improved computer science courses.

3. A long range benefit to college bound students who will be better prepared for their college courses in the scientific, mathematic, and business programs.

4. A benefit to colleges who will be able to improve their offerings in comuter science because of better prepared freshmen.

5. A benefit to teacher education programs and their students by having more capable teachers to supervise student teaching and other clinical activities.

4. Participants.

Participants were selected on the basis of their potential for exerting leadership in the development of computer science in the public and private schools of southwestern Ohio. Factors contributing to the acceptance of applicants included (1) the likelihood that the applicant would be able to apply what was learned in the project as indicated by subjects normally taught and possession of State teaching certification in computer science, (2) the applicant's interest in self improvement as indicated by a history of self study and enrollment in courses and workshops on computer science, and (3) indication of plans and interest for upgrading and extending the computer science curriculum in the applicant's home school and school system. Eight of the 30 positions were reserved for priority appointment of persons of racial minorities, and 15 were reserved for priority appointment of women.

There were 51 applicants for the 30 positions. In order to compensate for anticipated dropouts during the project, 33 participants were selected, with 28 entering with the first course and 5 (who did not need the first course) entering with the second. Tables 1, 2, and 3 show the composition of the group at the beginning and at the end of Courses 1, 2, and 3,respectively, and Table 4 shows the composition of the group at the beginning of course 4. (At the time of writing data was not yet available regarding the end of Course 4.)

	MEN	(MINORITY)	WOMEN	(MINORITY)	TOTAL
BEGINNING	16	(0)	12	(2)	28
END	13	(0)	11	(2)	24

TABLE 1. COMPOSITION OF PARTICIPANTS IN COURSE 1.

	MEN	(MINORITY)	WOMEN	(MINORITY)	TOTAL
BEGINNING	14	(0)	11	(2)	25
END	12	(0)	11	(2)	23

TABLE 2. COMPOSITION OF PARTICIPANTS IN COURSE 2.

	MEN	(MINORITY)	WOMEN	(MINORITY)	TOTAL
BEGINNING	14	(0)	11	(2)	25
END	11	(0)	11	(2)	22

TABLE 3. COMPOSITION OF PARTICIPANTS IN COURSE 3.

	MEN	(MINORITY)	WOMEN	(MINORITY)	TOTAL
BEGINNING	12	(0)	11	(2)	23
END		( )		( )	

TABLE 4. COMPOSITION OF PARTICIPANTS IN COURSE 4.

Attrition was attributed to several factors. Reasons given by participants for dropping out of courses or for not beginning the next course in the sequence included inability to keep up with course assignments due to job pressures, inability to keep up with course assignments due to family pressures, serious illness of the participant, and serious illness in the immediate family.

5. Curriculum

The four courses offered were:

Course 1. Structured Programming and Computer Algorithms (using Pascal)

Course 2 Data Structures (using Pascal)

Course 3. Microcomputer Systems, Architecture, and Assembly Language

Course 4. Instructional Sequences, Topics, and Materials of High School Computer Science

The instructors of the first three courses were members of the Department of Systems Analysis and the instructor of the last course was a member of the Department of Teacher Education, who also served as project director.

Course 1, Structured Programming and Computer Algorithms, was given on Saturday mornings during the spring semester of 1988. The class met for approximately three hours in each session. The course was a fairly standard introduction to programming similar to CS1 in the ACM curriculum. The textbook used was Oh! Pascal!, by Doug Cooper and Michael Clancy. The course objectives were to help the participants:

1. To master some fundamentally sound methods of problem solving.
2. To advance their levels of skill at solving problems of increasing difficulty.
3. To learn the syntax and semantics of the Pascal programming language including control structures, standard and user defined data types and functions, advanced input and output, procedures, etc.
4. To learn some elements of good style in the construction of computer programs, and to begin to appreciate why these elements are necessary.
5. To master some of the standard algorithms used in numerical and non-numerical applications.

The only major component of Pascal which was not covered in this course was the pointer data type. The students completed 10 Pascal programs which varied from a simple numerical calculation to a file handling program.

Since training in programming was to be a major emphasis of this program, one of the most important decisions was the hardware and software to be used. Since the participants were to commute over rather long distances, the use of the university's computers was unreasonable. There was no uniform hardware or software system to which all of the participants had access. For Courses 1 and 2 the decision was made to let each participant choose the compiler and computer to use for program assignments. This permitted the participants to gain more experience on the machine and compiler which they used for teaching. Some time was spent in class explaining some of the idiosyncrasies of some of the commonly used systems. (The two most used systems were the Apple IIe with Apple Pascal and the IBM PC or clone with TurboPascal.)

Courses 2 and 3 were given in the summer in one five week term. All participants were in enrolled concurrently in both courses. Each course met four days per week, 105 minutes per day.

Course 2, Data Structures, used the textbook, Introduction to Data Structures with Pascal, by Naps and Singh. The objectives for this course were to help participants:

1. To design and use internal data structures for better software design and implementation.
2. To design complex and hybrid data structures needed for specific and more advanced applications and simulate these data structures using data structures provided by the language .
3. To become capable of analyzing and evaluating the efficiency and effectiveness of various data structures and algorithms.
4. To master some of the traditional algorithms for manipulation of data structures.

The students were assigned four Pascal programs to complete outside of class to provide some experience in applying the data structures and algorithms described in class. Some class time each week was devoted to work in one of the university's PC laboratories.

Course 3, Microcomputer Systems, Architecture, and Assembly Language Programming, was a study of computer architecture with an emphasis on microcomputers. The objectives of this course were to help the participants develop the ability:

1. To describe alternative computer system implementations as seen by the programmer.
2. To describe the ways that the architectural components of the computer interact.
3. To describe the inter-relationships between a computer's architecture and the machine language executed by that computer.
4. To describe the relationship between the assembly language, the assembler, and the machine language.
5. To describe the relationship between machine languages and higher level languages.

Because of the differences among microcomputer architectures, among the instruction sets of their central processing units, and among the assemblers written for these different microcomputers, it was deemed necessary to focus programming assignments on a common hardware and software system. The Apple IIe was selected as the hardware system, because it is the most commonly used system in our locality. The assembler selected was part of the Apple Assembly Language training package distributed by the Minnesota Educational Computing Consortium (MECC). This package includes a booklet of lessons and reference material along with software that.includes an editor, an assembler, a graphic machine language simulator, and a step and trace debugger. Each student was given a copy of this package . The textbook used was Programming the 6502 by Rodney Zaks. Additional readings were assigned from computer architecture texts and computing journals in order to illustrate alternative computer architectures. This course was taught in a micro computer laboratory so that the students could enter and execute programs in class. This, along with the MECC simulator, was a useful aid to the students.

Course 4, Instructional Sequences, Topics, and Materials of High School Computer Science was given on Saturday mornings during the fall semester. The class met for approximately three hours in each session. No textbook was used for this course, but readings included the ACM's recommended guidelines for high school computer science courses, the College Board's course guide and course description for the advanced placement computer science course, and selected articles from professional journals. The course objectives were:

1. To introduce participants to the topic and techniques of computer based instrumentation as a subject of study for high school students.
2. To help participants develop model curricula for high school computer science programs.
3. To help participants develop topic outlines for model courses in these curricula.
4. To help participants develop effective units of study for these courses.
5. To help participants develop effective exercises and programming assignments for these units of study.

Budgetary and other practical constraints of time limited the program to four courses. These four course were chosen as the four which best satisfied the goals of the program. The first two courses, Structured Programming and Computer Algorithms and Data Structures, were obvious choices in light of the fact that almost all of the participants were primarily experienced in BASIC. Several knew no Pascal; others could write a simple Pascal program; and a very small minority were relatively proficient in Pascal. In most school districts in which the participants work, there is either a Pascal course offered or one on the horizon.

The topics for the remaining two courses presented more difficult choices. The decision could have been made to pursue some special topic more completely. For example, courses in operating systems, data communications, artificial intelligence, data base design, or software engineering could have been selected. However, the choice for a course in computer architecture with assembly language seemed to be more basic. This course helps provide a foundation for further study and supports an understanding necessary to teach the lower level courses. It also gave the participants an introduction to the knowledge necessary to interface microcomputers to other devices. This is a skill that may prove valuable to a school district, especially in the inclusion of computer based instrumentation in science classes.

The fourth course served as a capstone to this program by providing the participants with many ideas about methods of applying what they had learned in the previous courses and by serving as a means of curriculum development for strengthening and extending computer science programs in the local school systems.

6. Problems.

This program has been labeled as a success by both the participants and the teachers. However, it was not without some problems. The most glaring problem was an intrinsic one - the diversity of the participants. As in any group with this many people there was a difference in the intellectual abilities of the participants. This difference particularly apparent in the amount of effort necessary to complete the programming assignments outside of class. The programs assigned in Course 1 were similar to those assigned to freshmen in an introduction to programming course. A frequent complaint from a segment of this class was that the programming assignments were too time consuming.

Even though all were high school teachers involved in the teaching of programming at that level, there was no uniformity in their experience or training. The majority were trained and certified in mathematics. The second largest group were those trained in one of the sciences. Some were certified in English; one in Home Economics. (The diversity of the training speaks to the need for teachers.certified in computer science.) One participant has a consulting business in addition to his teaching position. Others have only recently begun to learn to program.

These diversities of background and abilities were amplified by the diversity in hardware and compilers that the participants used to complete the out-of-class assignments. Those members of the class who were dependent upon the Apple IIe with the Apple Pascal compiler found themselves at a disadvantage, particularly when compared to those using TurboPascal. This disadvantage arose because the primitive architecture and slow speed of the Apple IIe made compiling and debugging on that system arduous.

The travel necessary for the participants to attend class meetings was often exhausting and time consuming. The amount of travel necessary varied from forty to two hundred miles per round trip. Despite the travel, attendance in the courses was outstanding. This distance also exacerbated the difficulty of the out-of-class programming assignments. Many programming problems, which required the student hours of labor to solve, could have been resolved in a matter of minutes with help from the professor. A few students used the telephone to obtain help with programming problems although this was not optimal since the professor did not have a copy of the errant program.

Courses 1 and 4 were given during the school years while the participants were involved in teaching their normal high school loads. (Release time is much less common at the high school than at the college level.) The amount of work outside the classroom which was necessary in Course 1 to acquire the necessary programming skills made this a traumatic semester for the participants.

Courses 2 and 3 were given concurrently in a five week summer term. This created another intense period for the participants. Course 2, Data Structures, required significant outside time because of the programming. The professor attempted to ameliorate this somewhat by devoting some class time each week to work in a PC lab.

Course 3, Microcomputer Systems, Architecture, and Assembly Language, required the participants to master some very unfamiliar concepts. This was the first time that many participants had been exposed to the internal working of a computer. This was a vital, albeit difficult, experience for them. Additionally, there is no text that combines a general introduction to computer architecture with 6502 assembly language.

Course 3 had the added problem of using the Apple IIe as the hardware base. The nature of the course required the use of a uniform architecture by all participants. The Apple IIe seemed to be the obvious choice since many school districts use this machine. However, this machine was not familiar to all participants. Some would have preferred to learn about the architecture of the machine used by their school system. (The most common alternative was the IBM PC.)

There is a disparity in the expectation of the type of course and the amount of outside work necessary for a traditional graduate level course and a typical inservice workshop for high school teachers. This difference required an adjustment in the expectations of both the participants and the professors. (This was especially true for the professors from the System Analysis Department since they have not regularly taught this type of inservice training course.) It was necessary for the professors to make a conscious effort to limit the amount of outside work. In addition, the nature of adult education required the professors to make adjustments. Many of the students seem to require a longer period of time to grasp concepts and skills than younger college students. However, adults seem to have a greater perseverance, and are very conscientious in their work.

5. Conclusions.

Despite these problems, the immediate reaction of the participants and the professors in this effort has been positive. Both groups felt that the goals of the project were achieved The participant teachers have acquired greater skills and an increased level of confidence in those skills. The professors have enjoyed the contacts that they have made with their colleagues at the high school level. They feel that this program will, in the long run, improve the quality of student that they receive from these high schools.

The increased expertise of the teachers and the improved programs and courses that can be expected to develop will enhance these local schools systems as sites for nurturing the university's student teachers.

Another benefit of this program is the increased cooperation between the Teacher Education Department and the System Analysis Department. Neither could have carried the program individually. The Teacher Education Department was the source of contacts with the local school districts and high school teachers; the Systems Analysis Department was the source of the technical expertise in the computer science area. This effort has resulted in a synergism which will make both departments better.

6. Future Plans.

This program was funded for one year. Additional funding will be sought to continue this program on a yearly basis. The need for such a program is still strong and will remain strong until the number of teachers certified in computer science has increased to the demand. We will evaluate and analyze the differences between the results of pre-test and post-test given in two of teh courses. We plan to circulate a newsletter among the participants and other high school computer science teachers. This newsletter will be used as a vehicle for communicating ideas, methods, and resources which are useful to the high school computer science teacher.

7. Bibliography.

Apple Computer Inc., Apple II Reference Manual, Cupertino, California, 1981.

Brink, James and Richard Spillman, Computer Architecture and Vax Assembly Language Programming, Benjamin Cummings, Menlo Park, Ca., 1987, pp.7-14.

College Entrance Examination Board, "Advanced Placement Course Description: Computer Science", CEEB, 1986.

College Entrance Examination Board, "Teacher's Guide to Advanced Placement Courses in Computer Science", CEEB, 1983.

Cooper, Doug and Michael Clancy, Oh! Pascal, second edition, W.W. Norton, New York, N.Y., 1982.

Department of Teacher Education and Certification, "Teacher Education and Certification Standards (Effective July 1, 1987)", State Board of Education, State of Ohio, Columbus, Oh., 1985.

Eckhouse, Richard and Robert Morris, **Microcomputer Systems**, second edition, Prentice-Hall, Englewood Cliffs, NJ, 1979, pp.17-41.

Grehan, Richard and Jane Tazelaar, "What They Did Wrong", Byte, McGraw-Hill, May 1988, pp. 239-248.

Minnesota Educational Computing Consortium, "Apple Assembly Language (Version1)", MECC, Minneapolis, Mn.

Naps, Thomas and Bhagat Singh, Introduction to Data Structures with Pascal, West Publishing Co., St. Paul, Mn., 1986.

Stallings, William, Computer Organization and Architecture, Prentic-Hall, Englewood Cliffs, NJ, 1987, pp.20-48.

Wilson, Pete, "The CPU Wars", Byte, McGraw-Hill, May 1988, pp.213-234.

Zaks, Rodney, Programming the 6502 (Fourth Edition), Sybex, Berkeley, Ca, 1983.