Kansas High School Computer Science Teachers' Professional Development

by

Debbra A. Pontier

B.S.E., University of Kansas, 1996

Submitted to the Department of Teaching and Leadership and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science in Education.

Dr. Ron Aust
Professor in Charge

Dr. George J. Crawford

Dr. Jerry D. Bailey
Committee Members

May 9, 2005
Date project accepted

ABSTRACT

Secondary computer science teachers across the nation are faced with teaching curriculum that continually evolves – with minimal or no formal training in computer science. The purpose of this study was to research high school computer science teachers' perceived proficiency and professional development awareness and utilization. Specifically, the research questions were:

1. How do Kansas high school computer science teachers perceive their proficiency in teaching computer programming?
2. Is there a relationship between awareness and utilization of professional development opportunities for Kansas high school computer science educators?

A survey was given to certified computer studies and computer programming high school teachers across the state of Kansas. A total of 42 teacher surveys were returned and analyzed.

The results showed that survey participants scored themselves with a low perceived proficiency rating. Survey results also demonstrated a general lack of awareness of computer science professional development opportunities. However, their participation/utilization of these opportunities was evident when participants were aware of computer science professional development opportunities. In general, the participation/utilization of professional development opportunities was significantly lower than the level of awareness for professional development opportunities.

Furthermore, in two-thirds of the professional development opportunities listed in the survey, future plans for participation/utilization were greater than previous participation/utilization experiences. Teachers consistently and adamantly reported that levels of awareness and utilization of professional development opportunities would increase with additional resource time. An overwhelming 85% of the teachers said that professional development support would help alter their perceived proficiency rating.

TABLE OF CONTENTS

CHAPTER ONE

THE RESEARCH PROBLEM

Introduction

Computer science is taught in high schools throughout the United States. Although the number of students taking computer science courses is only a small percentage of high school students, computer science is an important part of curricula across all grade levels (Weinstein & Resnick, 1999). The computer science field of study is young and growing expeditiously. Multiple challenges are an issue for all involved in the computer science field due to the evolving nature and relative newness of the field. There is a trend that computer science teachers at the secondary level have a minimal formal computer science background. As advances in this field of study continue to evolve, schools and their computer science teachers should not be left behind (Gal-Ezer, 1995).

Purpose of the Study

The purpose of the study was to research high school computer science teachers' perceived proficiency and professional development awareness and utilization. Data was compiled to analyze the following questions:

1. How do Kansas high school computer science teachers perceive their proficiency in teaching computer programming?
2. Is there a relationship between awareness and utilization of professional development opportunities for Kansas high school computer science educators?

A survey was given to high school computer science teachers across the state of Kansas. The survey gathered data on the current state of teacher preparation, teacher responsibilities, teacher support, and professional development in Kansas high school computer science programs. The surveys' inquiries focused on teacher-perceived instructional proficiency and correlation factors for teachers' awareness and utilization of professional development opportunities for high school computer science educators.

Rationale for the Study

Secondary computer science teachers across the United States often start with little academic training in the content of computer science and many times even less training on how to teach computer science. Once these teachers get into their classrooms they are overwhelmed with the task of keeping on top of multiple preparations for several different levels of classes taught in multiple languages, and they are teaching in an isolated environment. To make matters worse, there is probably no one else in their buildings, or maybe even in their school districts, to go to for content support or pedagogic needs/support in this content area. Research results from an August 2002 survey, conducted by the Association for Computing Machinery: K-12 Task Force Committee on Computing Curriculum found that most states have no state mandated/supported computing curriculum (Association for Computing Machinery, 2002). More than half of the school districts are not required to work from specific curriculum guidelines (either state or school mandated). Less than 6% of those surveyed reported a requirement of a Computer

Science Certification to teach computer science in their state/district. In the new licensing system in Kansas, there is no longer a “computer studies” endorsement. According to Martha Gage, Director of Certification and Teacher Education at the Kansas State Department of Education, it will be an employment issue for districts to determine the competency of someone teaching computers (M. Gage, personal communication, July 7, 2003).

Overview

Computer science is a young field of study with much of the research focused at the university level. Secondary computer science teachers endure a deficiency in professional development, starting from the point of teacher preparation, moving on to staying current, and the lack of professional support. Curriculum standards, teaching methods, and language issues have not been addressed in a manner that supports the secondary computer science teacher. Research from the secondary teachers’ points of view could demonstrate there is a need for support and make an appeal to those who influence the professional development opportunities for secondary computer science educators.

Chapter two discusses a review of literature pertaining to high school computer science teachers and their teaching environments. Chapter three specifies implementation criteria of the research participants along with survey development, distribution tactics and analysis methods. Chapter four provides detailed results of the survey data and analysis results. Chapter five presents a summary and conclusion based on survey analysis.

CHAPTER TWO

REVIEW OF THE LITERATURE

Introduction

Although teachers in general are excited about working with students, educators in the field of computer programming "seem uniquely passionate about their work" (Weinstein & Resnick, 1999, p. 42). The dynamic and relatively young disciplines of computer science and computer science education have presented numerous teaching issues and controversies that come with trying to cope with rapid changes (Gal-Ezer, 1995; Schwill, 1997). Gal-Ezer and Harel (1998) note that even two prominent computer scientists have conflicting views of this young and unique field of study: "Computer science has such intimate relations with so many other subjects that it is hard to see it as a thing in itself." – M.L. Minsky, 1979 and "Computer science differs from the known sciences so deeply that it has to be viewed as a new species among the sciences." – J. Hartmanis, 1994.

Stephenson (2002) describes technology as a key factor to the United States' financial well-being. She goes on to suggest that to remain competitive and a leading nation in the tech sector, we must start with strong high school computer science programs (Stephenson, 2002). But within school systems there is a gap that exists between educators and administrators that have a comfort level with technology and their colleagues that are not comfortable with technology (Tucker, 1996). Tucker (1996) believes that the educational community at all levels needs to become technologically savvy so they can integrate technology effectively into curricular subjects. Many high school computer teachers have their current teaching position by default and have limited knowledge of the field they are teaching (Gal-Ezer, 1995). Weinstein and Resnick (1999) indicate that hiring a "programming-savvy" teacher and keeping him/her in the position for a salary in the range of $35,000 a year is unrealistic.

This leads to several areas of concern in computer science and engineering education: general curriculum issues, undergraduate education, graduate education, K-12 education, and coordination within the education community (Tucker, 1996). Stephenson (2002) suggests beginning the reform with issues of teacher training and professional development, curriculum standardization, and time/resource allocation. Stephenson (2002) also reiterates that "our future depends on these things."

Professional Development

In the area of computer science education, just reading a book is not the solution for understanding what isn't known (Lister, 2003). According to Lister's (2003) five orders of teaching ignorance, most computer science educators are in the third order of ignorance, "when you lack the process for finding out that you don't know something." There is no infrastructure set for training and keeping teachers up to date with technology and within the field of technology (Tucker, 1996). In fact, in most cases there are no qualification requirements for high school computer science teachers (Gal-Ezer, 1995). In all areas of teaching it is important to have well trained teachers, and within the dynamic area of computer science it is even more critical (Gal-Ezer, 1995). Not only should these high school teachers be qualified educators, but they also need a broad and deep understanding of computer science (Gal-Ezer, 1995). Gal-Ezer (1995) compares the requirements for high school physics and mathematics teacher qualifications to the lack of high school computer science teachers' qualification requirements when she declares, "No self-respecting school system will give a self-taught teacher or person with only high school education, full responsibility for the high school physics or mathematics program."

Teacher Preparation

The numbers of technology courses are increasing in high schools and one of the biggest obstacles is a lack of qualified computer science teachers (Tucker, 1996; Weinstein & Resnick, 1999). Frequently high school computer science teachers are deficient in skills and training, due to the fact that they have little formal preparation in computer science and computer programming (Gal-Ezer & Harel, 1998; Tucker, 1996; Lidtke & Moursund, 1993). This lack of formal education starts at the teacher preparation level (Lidtke & Moursund, 1993). The research of Lapidot and Hazzan (2003), comprehensive "Methods of Teaching Computer Science in the High School," shows that courses for high school computer science teachers are scarce.

The Office of Educational Technology in the U.S. Department of Education (2005) released the National Education Technology Plan in January 2005. The plan has seven action steps for school systems to consider in their transformation into the technology-driven era, and even refers to the United States as a “Nation on the Move.” The action step that is devoted to “improve(ing) teacher training” provides recommendations to: (1) “Improve the preparation of new teachers in the use of technology,” (2) “Ensure that every teacher has the opportunity to take online learning courses,” (3) “Improve the quality and consistency of teacher education through measurement, accountability and increased technology resources,” and (4) “Ensure that every teacher know how to use data to personalize instruction. This is marked by the ability to interpret data to understand student progress and challenges, drive daily decisions and design instructional interventions to customize instruction for every student’s unique needs” (United States Department of Education, 2004).

Since the 1993 first edition release, the International Society for Technology in Education (ISTE) has been developing technology standards for all teachers (ISTE, 2004). The National Education Technology Standards for Teachers (NETS•T) is ISTE’s third and latest version and includes 23 indicators in the following six categories: I. “Technology Operations and Concepts,” II. “Planning and Designing Learning Environments and Experiences,” III. “Teaching, Learning, and Curriculum,” IV. “Assessment and Evaluation,” V. “Productivity and Professional Practice,” and VI. “Social, Ethical, Legal, and Human Issues.” The goal of ISTE’s NETS project is to “develop national standards for educational uses of technology that facilitate school improvement in the United States” (ISTE, 2004). The NETS project has developed standards for both students and teachers (ISTE, 2004).

ISTE recommends guidelines for accreditation to the National Council for Accreditation for Teacher Education (NCATE) for educational computing and technology teacher preparation programs (ISTE, 2004). NCATE takes these guidelines and develops standards of preparation for computer science and computer literacy teachers (ISTE, 2004). The 1997 guidelines for Secondary Computer Science Education Endorsement and Secondary Computer Science Education Bachelor’s Degree are now discontinued (ISTE, 2004). The 2003 revised ISTE/NCATE Standards for Secondary Computer Science Education (CSED) for endorsement/degree program standards has been approved (ISTE, 2004). These standards are geared to the secondary computer science teachers and focus on teachers being able to prepare their students on the more technical aspects of computing, including “problem analysis, algorithm selection and evaluation; program design, implementation, specification, and verification; and systems analysis” (ISTE, 2004).

Even though computer science education committees have common interests, they do not always interact with each other (Tucker, 1996). The CSED standards are in place, but the call for comprehensive programs to train new computer science teachers at the beginning of their careers and/or the opportunity to take these comprehensive programs some time in the computer science teachers’ careers have not been met (Gal-Ezer, 1995; Gal-Ezer & Harel, 1998; Lapidot &

Hazzan, 2003; Lidtke & Moursund, 1993; Tucker, 1996). Not only do the high school computer science teachers need methods courses and classes, but the Task Force of the Pre-College Committee of the Education Board of the ACM says high school computer science teachers also need a formal education in computer science (Association for Computing Machinery, 1993). The formal computer science education should be as comprehensive as possible, both broad and deep, and include the history of the field along with specific curricula (Gal-Ezer, 1995; Gal-Ezer & Harel, 1998). Gal-Ezer and Harel feel this formal education should be thorough and include at least the material covered by a computer science graduate and preferably a Master's degree in computer science (1998).

Staying Current and Professional Support

Teachers work hard to keep skills current and their classroom engaging and relevant for their students (Stephenson, 2002). Due to the rapid changes in technology the need for continual updating in this field of study can become difficult and often unproductive (ACM, 1993; Gal-Ezer, 1995). Opportunities for continuing education and the need for better mechanisms for sharing education resources are enormous challenges (Gal-Ezer, 1995; Tucker, 1996). In 1997 the Educational Testing Service was getting ready for the 1999 Pascal to C++ language change in the computer science advanced placement test (Fisher, 1997). It was estimated that only 20% of the 1,500 teachers of advanced placement computer science courses would receive the needed training in C++, object-oriented style of programming, and the pedagogy for the new language (Fisher, 1997). At this time most teachers of high school computer science had not worked with object-oriented programming languages (Stephenson, 2002). The high school computer science teachers that taught advanced placement courses in 2004 once again were faced with the challenge that the advanced placement computer science test would undergo another language change, C++ to Java (ACM, 2003). A national study conducted by ACM reported that 80% of the high school computer science teachers felt "ill-prepared to teach Java specifically and object-oriented principles in general" (J. Wroblewski, personal communication, September 29, 2003). In response to survey, in November 2002 ACM and the College Board brought together computer science higher education and secondary education faculty in workshops at various colleges and universities through a program called JETT, Java Engagement for Teacher Training (ACM, 2003; ACM, 2004).

Another sub group of ACM is the Special Interest Group on Computer Science Education, better known as SIGCSE (ACM, 1997). SIGCSE is an international forum for discussions, conferences, quarterly bulletins, proceedings from symposiums, and a web page that also serves as significant resources for a community of professionals who share their experiences as educators (ACM, 1997; Tucker, 1996). SIGCSE provides a means to "discuss concerns about development, implementation, and evaluation of computing programs and courses, as well as syllabi and problem sets" (ACM, 1997). In the first quarter of 2005, ACM launched the new Computer Science Teachers Association, CSTA, to address serious issues in high school computer science education and to ensure an environment where teachers have the tools they need to promote student interest in computer science careers (ACM, 2005). CSTA's purpose is to build a community for computer science educators to share knowledge, have opportunities for high quality professional development, advocate for comprehensive computer science curricula, support projects that promote student interest in computer science, collect and disseminate research about computer science education, support computer science in the high school curriculum through policy recommendations, and raise awareness about computer science educators (ACM, 2005). CSTA's members include middle school teachers, high school computing teachers, college/university computer science faculty and education faculty, and members from industry interested in supporting computer science education and teachers (ACM, 2005).

Although symposiums are important events, many teachers are not attending and there is still a need for "localized, ongoing, curriculum-centered teacher learning" (Stephenson, 2002). To fulfill this need, states must to make funded commitments and give released time to promote mandatory teacher participation for ongoing professional development and attendance at educational state and national conferences (Stephenson, 2002). Strong teaching institutions should "moderate teaching loads and increase their real support" for continued faculty development (Tucker, 1996). Since "industry needs" play a significant role in the popularity of what programming is offered in high schools, internships in industry and research-oriented university settings would also provide support for faculties to remain current (Weinstein & Resnick, 1999; Tucker, 1996).

Curriculum

The push to integrate computers at all levels of education, in numerous ways, has caused confusion and the inability for many to distinguish the difference between computer-assisted instruction, computer applications, computer literacy, and computer programming (Gal-Ezer, 1995; Gal-Ezer & Harel, 1998; Lidtke & Moursund, 1993). Since the 1980s non-programming computer use has been on the rise as programming classes have declined in the field of education (Lidtke & Moursund, 1993). In fact many high schools no longer offer computer programming and computer science courses and instead dominantly offer computer applications and computer-assisted instruction (Lidtke & Moursund, 1993). A portion of this phenomenon, which creates an imbalance between educational opportunities between high schools, is due in part to the lack of understanding by the general educational population in distinguishing between the different computer uses (Gal-Ezer & Harel, 1998; Tucker, 1996).

The field of computer science has its own problems in coming to an agreement even for the name of the field: computer science, informatics, information systems, computer studies, algorithmics, software engineering, etc. (Gal-Ezer & Harel, 1998). Computer programming promotes logic, communication, and critical-thinking skills that are educational goals of many states curricula in math, science, technology, and even language arts (Weinstein & Resnick, 1999; Thomas & Upah, 1996). Along with these positive aspects come the pressures and pitfalls of the language issue, the challenge of getting abstract and detailed notions across, and teaching/facilitating an algorithmic and precision thinking (Gal-Ezer & Harel, 1998). Computer science educators are faced with the challenge of developing pedagogical strategies and curriculum materials that enhance and encourage metacognative skill development (Thomas & Upah, 1996). The discipline and technology are changing so fast that maintaining current curriculum, both what and how we teach students, is a difficult task and is under constant revision (Stephenson, 2002; Tucker, 1996). Since this dynamic and youthful discipline is only around seventy years old, the opportunity to hear the pioneers of the field teach and lecture or even work with them still exists (Gal-Ezer, 1995). Computer science educators need to respond, take a fresh look at their courses, and evolve in how and what they teach as the discipline continues to age (Gal-Ezer & Harel, 1998; Tucker, 1996; Thomas & Upah, 1996).

The changes in the curriculum and programming language of the advanced placement computer science committee have been a driving force in what high schools across the U.S. will offer for computer science classes (Elkner, 2000; Harvey, n.d.). This national following happens partly because there has been a lack of leadership at the secondary level for computer science (Harvey, n.d.). As a result the advanced curriculum, designed for a college population, has set the tone for most of the high school programming instruction in the nation (Elkner, 2000; Harvey, n.d.).

Local/State/National Standards

The requirements for advanced placement computer science were not meant to be the standard curricula, even though many computer science educators feel these are the only standards they are given (Stephenson, 2002). Standards, whether national or state, would provide teacher guidance, guarantee a more consistent student learning experience, and address both introductory and advanced placement computer science courses (Stephenson, 2002). Although some states have computer literacy requirements, many states have passed the technology and computer science curriculum responsibility on to the school districts (Lidtke & Moursund, 1993).

The National Education Technology Plan says that, along with improving student achievements in math and science, we should adequately prepare students in the mastery and applications of technology (U.S. Department of Education, 2005). National standards have been developed and were implemented for math in 1989 by the National Council of Teachers of Mathematics and for science in 1996 by the National Research Council (National Science Foundation, 2005; Thomas & Upah, 1996). The National Education Technology Standards for Students (NETS•S) is ISTE's technology standards for students, but it does not cover any standards for the computer science area consisting of computer programming (ISTE, 2004).

The ACM K-12 Computer Science Curriculum Committee designed and has given its final report for a model curriculum for K-12 computer science (Tucker et al., 2004). The curriculum design includes four levels: “Level I (grades K-8) – Foundations of computer science,” “Level II (grade 9) – Computer science in the modern world,” “Level III (grade 10 or 11) – Computer science as analysis and design,” and “Level IV (grade 11 or 12) – Topics in Computer Science” (Tucker et al., 2003; Tucker et al., 2004). While developing the model for the secondary education sector, the committee integrated programming and applications approaches to computer studies to create core, recommended, and optional curricula to introduce computer science (ACM, 1993).

Currently a coherent secondary computer science curriculum plan has not been put into practice across the nation, and curriculum varies greatly from state to state, district to district, and even school to school (Tucker, 1996; Stephenson, 2002). It is possible for groups of teachers to form their own network, within a single school, the same city, or even the same region, and analyze curricula needs and the fundamental ideas they want to convey to their students (Schwill, 1997). An example of forming computer science curriculum objectives is given by the Shawnee Mission School District, in Shawnee Mission, Kansas. Since their Computer Science Mastery Objectives are not published, they have been included in Appendix C. The Shawnee Mission School District computer science objectives are written for four different classes: Topics in Computer Science, Programming in Java, Programming in Java Advanced, and Advanced Placement Computer Science, and include objectives for 11 strands.

Teaching Methods

Experimenting with diverse models of education is an important strength of the computer science educational community (Tucker, 1996). To find the balance between technology and pedagogy while giving students the content is quite a challenge (Stephenson, 2002). Using conventional teaching methods to reach students in unconventional territory, technology and microcomputers, will result in lost students and frustrated teachers (Garrett, 2002). Teachers will need to be facilitators, experts available for consultation, instead of dispersers of knowledge (Harvey, n.d.). These same teachers will need creative ways to disseminate knowledge so it is both understandable and effective (Garrett, 2002). Luckily, many programming students rank programming very high among their priorities and their natural curiosity is guided by the nurturing of teacher questions and/or comments (Schwill, 1997; Harvey, n.d.). The challenge of communicating fundamental concepts while representing the scope of the evolving and vast area of technology, and all the while encouraging students to be independent learners so they can meet society's long-term needs for a tech-savvy nation, is what falls on the shoulders of computer science teachers (Stephenson, 2002).

The students of computer science have a wide range of learning styles and backgrounds with immediate futures that include exploring an interest in programming, preparing for college level programming, and/or possibly entering the job market as professional programmers (Tucker, 1996; Harvey, n.d.). Programming students need to be nurtured in precise thinking skills and metacognative development (Gal-Ezer & Harel, 1998; Thomas & Upah, 1996). The logic, communication, and critical thinking skills that are developed in programming will extend into an algorithmic way of thinking that will benefit them in other subjects as well (Weinstein & Resnick, 1999; Gal-Ezer & Harel, 1998). The reality of the high school computer science classroom provides a time frame that actually allows for concentration on foundational concepts and skills while focusing on computer-based solutions to problems, algorithms and data structures, design and development, and even ethical & social implications of computer use (Stephenson, 2002; Weinstein & Resnick, 1999). Not only will students be introduced to speaking the language, semantics, syntax, program structure, and code block, they will also explore structured, top-down, and object-oriented programming (Stephenson, 2002; Garrett, 2002; Schwill, 1997). More advanced students will use all the previously taught fundamentals and apply them when they write larger, manageable programs and analyze the complexity of programs (Stephenson, 2002; Schwill, 1997).

Languages

The widely debated and controversial issue of what should be the first programming language that students learn has created something of a culture war (Gal-Ezer & Harel, 1998). The debate varies from teaching programming concepts void of language exposure to making sure students are familiar with two or more languages in the early stages of their computer science education (Gal-Ezer, 1998; Harvey, n.d.). By the mid-1980s BASIC (Beginners All-Purpose Instruction Code) and Logo were on track to be the dominant pre-college programming languages (Lidtke & Moursund, 1993). High-level programming languages opened their doors to computer science education and the continual development and introduction of these languages have created a long list of possible pre-college languages, BASIC, Logo, Pascal, Visual Basic, C, C++, Java, J++, and scripting languages such as HTML (Stephenson, 2002; Weinstein & Resnick, 1999). In 1999 the College Board's advanced placement computer science exam was given in C++ instead of Pascal (Elkner, 2000; Harvey, n.d.). The exam again switched, from C++ to Java, in 2004 (Drysdale, Hromcik & Weiss, 2003; Elkner, 2000; Harvey, n.d.). At each of these intervals, due to the College Board's change in languages and curriculum, many high schools throughout the country decided to change the languages and curriculum being taught (Elkner, 2000; Harvey, n.d.). Currently Visual Basic, C++, and Java, all object based and/or object oriented languages, are the most common languages being used in secondary computer science classes, due to advanced placement curriculum and because they are widely used in industry (Drysdale, Hromcik & Weiss, 2003; Weinstein & Resnick, 1999). Even though Pascal was a good teaching language, its popularity in the classroom has greatly diminished since it is rarely used in the "real world" (Weinstein & Resnick, 1999).

Summary

The topics of professional development and curriculum are two major concerns for the dynamic and young disciplines of computer science and computer science education (Stephenson, 2000; Tucker, 1996; Schwill, 1997; Gal-Ezer, 1995). Responsibility for computer science education issues and decisions are at different levels throughout the nation (Lidtke & Moursund, 1993). In some cases, states have curriculum, literacy, and teacher certification requirements and, in many other cases, these decisions have been passed down to the school districts and even down to the individual school level (Lidtke & Moursund, 1993).

The U.S. Department of Education (2005) has released the National Education Technology Plan that addresses improving teacher training. ISTE has released the National Education Technology Standard for Teachers and the National Education Technology Standards for Students, although these standards address technology in the broad sense and not at the computer science level, (ISTE 2004). NCATE has approved Standards for Secondary Computer Science Education (CSED) for endorsement/degree program standards (ISTE 2004). The ACM K-12 Computer Science Curriculum Committee has released the final report for a model curriculum for K-12 computer science (Tucker et al., 2004). Hopefully, with these five elements coming into place within the last year, Stephenson's suggestion for reform on the issues of teacher training, professional development and curriculum standardization, and the Office of Educational Technology's vision of a "Nation on the Move," will be addressed (Stephenson, 2002; United States Department of Education, 2004).

CHAPTER THREE

METHODOLOGY

Introduction

A survey was given to high school computer science teachers across the state of Kansas. The high schools represented serve an expansive student representation, ranging from low-middle-upper class, college-bound and non-college bound students. The surveys' inquiries focused on teacher-perceived instructional proficiency and correlation factors for teachers' awareness and utilization of professional development opportunities for high school computer science educators.

Participants

All high school teachers listed as computer programming or computer studies certified personnel, using data from the 2002-2003 Certified Personal Report obtained at the Kansas State Department of Education website, were asked to complete a survey about perceived proficiency and professional development awareness and utilization for teaching computer programming. One hundred thirty-four surveys were mailed to sixty-six female and sixty-eight male teachers. The teachers surveyed for this study participated on a voluntary basis and were assumed to have provided honest, accurate, and unbiased responses. Since the participant list was derived from the Kansas State Board of Education website as certified personnel of computer studies and/or computer programming in the 2002-2003 school year at specified high schools, it was assumed that all teachers had taught computer science in the 2002-2003 school year were certified to teach computer studies and/or computer programming. Each high school was assumed to have offered computer science courses, to have provided lab equipment and course necessities, and to have had students that took the courses, since they listed certified personnel for the most recent school year.

Instrument

A survey was developed for teachers to complete that would provide data to answer the research questions. In order to evaluate the effectiveness of the survey, an advisor and two colleagues assessed the survey and provided feedback as to its wording, order, and length. Based on their contributions, modifications were made to the survey to improve its length, clarity, and flow. The finished survey was four pages, consisting of 21 items and a field for additional comments. A copy of the survey is located in Appendix A.

The survey was organized into sections, starting with precursor questions and then followed by three main components. The precursor questions asked for general teacher and school information, such as age, gender, high enrollment numbers, and computer science enrollment numbers. The three main components were divided into teaching experience, teaching environment, and professional development awareness and utilization. Part I requested teaching experience information from each teacher, such as total number of years teaching, number of years teaching high school computer science, and current teaching endorsements. Questions seven through nine in Part I requested what computer science classes and languages the teacher was currently and had previously taught. Two items in Part I focused on the specifics of each teacher's course load for the last five years. These two items asked for the exact numbers of overall course preparations, new course preparations, courses with new textbooks, different programming languages, College Now courses and Advanced Placement courses. The last two inquiries for Part I probed the teachers for information on whether they had computer science teaching experience outside of the high school and/or work experience outside of the educational field.

Part II of the survey contained four questions and addressed the teaching environment. The first question focused on frequency of networking contacts with other computer science teachers and/or administrators. Seven areas of contact were addressed, with teachers choosing from three response choices for each: "never/seldom," "often," or "almost daily." Two questions asked the teachers to rate their teaching beliefs and teaching practice on a continuum, varying from traditional teacher-center to constructivist student-centered. Teachers were also asked to rate their perceived proficiency as a computer science teacher on a five point continuum, varying from "Help … I'm Drowning" to "Totally Awesome." The final question in Part II asked if professional development support would help them alter their perceived proficiency rating. If the answer was yes, teachers were asked to list what professional development would help.

Professional development awareness and utilization was the essence of Part III. Although this section only had three numbered questions, questions nineteen and twenty requested twenty-one responses, ten regarding available training and eleven regarding support resources. Participants were asked to think about the last two years for each response and mark yes or no to whether they were aware of, utilized, and/or had future plans to utilize each item. A focal point of the survey was in the open response question that completed Part III. Teachers were asked what would help increase their awareness and utilization of professional development opportunities in the area of computer science education.

Procedures

During early November 2003 Kansas high school computer programming and computer studies certified teachers received a cover letter, survey, and postage-paid, pre-addressed envelope at their high school via the United States postal service. The teachers were asked to complete and return the survey to the researcher at Shawnee Mission East High School in Prairie Village, Kansas using the postage-paid, pre-addressed envelope they received. A deadline for the survey to be returned was provided, giving teachers approximately two weeks to fill out the survey.

Data Analysis

Forty-two teacher surveys were returned. The quantitative data collected from this survey was analyzed using descriptive statistics and statistical tests. Qualitative data, from questions eighteen and twenty-one, was compiled into a list of professional development support items that would help alter the proficiency ratings and types of support that would help increase awareness and utilization of professional development opportunities. An Excel spreadsheet was used to compile the data, create tables, and create charts.

Within the Excel spreadsheet, tables were created to represent data percentages for questions six and nine, types of current teaching endorsements and programming languages being taught. Tables were also created to represent the mean number of courses taught in reference to teaching load for questions ten and eleven. In question fourteen, types of contact between the computer science teachers surveyed and other computer science teachers and/or administrators were analyzed by frequency percent with a stacked column chart. Continuum diagrams helped visualize the mean for teaching environment in questions fifteen through seventeen that were related to teaching beliefs, teaching practice, and perceived proficiency. The mean for perceived proficiency, question seventeen, was broken down into several categories: overall survey responses, computer programming teachers, two to eight years teaching experience, nine to fifteen years teaching experience, sixteen to twenty-two years teaching experience, twenty-three to twenty-nine years teaching experience, and thirty to thirty-seven years teaching experience. Questions nineteen and twenty addressed twenty-one areas of available training and support resources, and line charts were created to represent the awareness, participation/utilization, and future plans percent of affirmative responses for both questions.

CHAPTER FOUR

RESULTS

Introduction

Data for this study was collected using a survey completed by forty-two Kansas high school computer science teachers. The survey questions measured each teacher's perceived proficiency and professional development awareness and utilization for teaching computer programming. The survey was designed to collect data pertaining to the following research questions:

1. How do Kansas high school computer science teachers perceive their proficiency in teaching computer programming?
2. Is there a relationship between awareness and utilization of professional development opportunities for Kansas high school computer science educators?

The teachers participating in the research study were listed as Kansas computer programming or computer studies certified personnel, according to data from the 2002-2003 Certified Personnel Report obtained at the Kansas State Department of Education website. The surveys were mailed and returned during November 2003. This chapter includes the results from the data analysis of the survey responses.

Limitations

Only Kansas high schools with adequate resources to offer computer science classes were included in the study. The expansive student demographic representation, high school resource disparity, and lack of state or national course standards created a unique diversity factor for this study. Each school district offered a distinctive array of computer science courses, programming languages, lab equipment, and student participants.

The primary data source for this research was the voluntary, unverified survey responses. Although all contributing teachers were listed as computer studies and/or computer programming personnel, the Kansas State Department of Education listing and teachers' responses were not checked for validity.

Overall Results

The preliminary questions gathered background information on each teacher. Thirty-one percent of the surveys mailed were completed and returned. Even though surveys were sent to sixty-six females and sixty-eight males, 49.3% females and 50.7% males, the responses received were from 64.3% females and 35.7% males. The teachers surveyed taught at high schools that had overall student enrollment, for the 2003- 2004 school year, mean of 712.5, ranging from forty-five to two thousand twenty-six students. The computer science course enrollment, for the 2003-2004 school year, ranged from 0 to three hundred ninety-six, with a mean of 93.1 computer science students per school.

Teaching Experience

Part I gathered teaching experience data from each teacher. Total number of years teaching (including 2003-2004) ranged from two to thirty-seven years, with a mean of 16.43 years. The number of years teaching high school computer science (including 2003-2004) ranged from 0 to thirty-three, with a mean of 10.2 years. Table 1 contains the percent responses for current teaching endorsements listed by respondents. Most of the teachers (92.9%) listed multiple teaching endorsements. Three major teaching endorsements were frequently listed, along with a variety of other teaching endorsements. Although 81% of the teachers listed a computer science endorsement, the endorsement was always coupled with an endorsement from another area of teaching. Math (40%) and business (45%) endorsements were a common theme among the listings. Ninety-five percent of the teachers responded that they currently teach a computer science class. Teachers listed three areas of computer science classes that they have been teaching during the last five years. These three computer science areas were classified as computer programming, computer technology (web design, multi-media, computer repair, and Cisco networking), and computer literacy (word documents, spreadsheets, data bases, and presentation software). Teachers often taught in more than one area of computer science (computer literacy 66.7%, computer programming 61.9%, and computer technology 45.2%).

Of the 61.9% of the teachers that listed computer programming as the curriculum basis of their computer science classes, there were several languages entered as being taught during the last five years. Table 2 contains the percent responses for all languages documented by the teachers. Visual Basic and C++ were the most frequently listed languages used within the computer programming curriculum, with Java and QBasic also being listed on a consistent basis.

Table 1

Percent Responses for Current Teaching Endorsements
(2003-2004)

Computer Science (Computer Studies, Computer Programming, Computers)	81%
Business	45%
Math	40%
Other	62%

Note: All percents have been rounded to the nearest whole number.

Table 2

Percent Responses for Programming Languages
(1999-2004)

C++	58%
Visual Basic / Visual Basic.Net	54%
Java	35%
QBasic	35%
Pascal	23%
HTML	15%
Fortran	4%

Note: All percents have been rounded to the nearest whole number.

Questions ten and eleven in Part I addressed teachers' course load specifics for the last five years. Table 3 contains the mean for the quantity of overall course preparations, new course preparations, and courses with new textbooks for each of the five school years from 1999-2004, and also contains the overall mean from 1999-2004. The mean for the quantity of different programming languages, College Now courses, and Advanced Placement courses for each of the five school years from 1999-2004, and the overall mean from 1999-2004, can be found in Table 4. Thirty-three percent of the teachers have taught computer science classes other than the high school level (i.e. community colleges, trade schools, colleges, etc.) and 19% of the teachers have worked in the computer science field outside the education field.

Teaching Environment

Part II of the survey instrument focused on teaching environment. A list of seven previously identified types of contact with other computer science teachers and/or administrators was presented to teachers on the survey instrument and each teacher was asked to mark the frequency of networking contacts. Teachers selected from three response choices for each type of contact: "never/seldom," "often," or "almost daily." The percent of teachers that selected the specific frequencies for each type of contact is displayed in the chart of Figure 1. In all seven types of contact, more than 50% of the teachers registered "never/seldom" for their contact frequency. "Almost daily" contact frequency was obtained 0% – 15% for all seven types of contact.

Table 3

Teaching Load for
Overall Course Preparations, New Course Preparations, Courses with New Textbooks

School Year	Overall Course Preparations #	New Course Preparations #	Courses with New Textbooks #
2003-2004	*4.15	*1.22	*1.27
2002-2003	4.32	.85	.90
2001-2002	4.05	.65	.59
2000-2001	4.17	.80	.63
1999-2000	3.97	.67	.85
1999-2004	*4.13*	*.84*	*.85*

**Mean number of courses taught.*

Table 4

Teaching Load for
Programming Languages, College Now Courses, Advanced Placement Courses

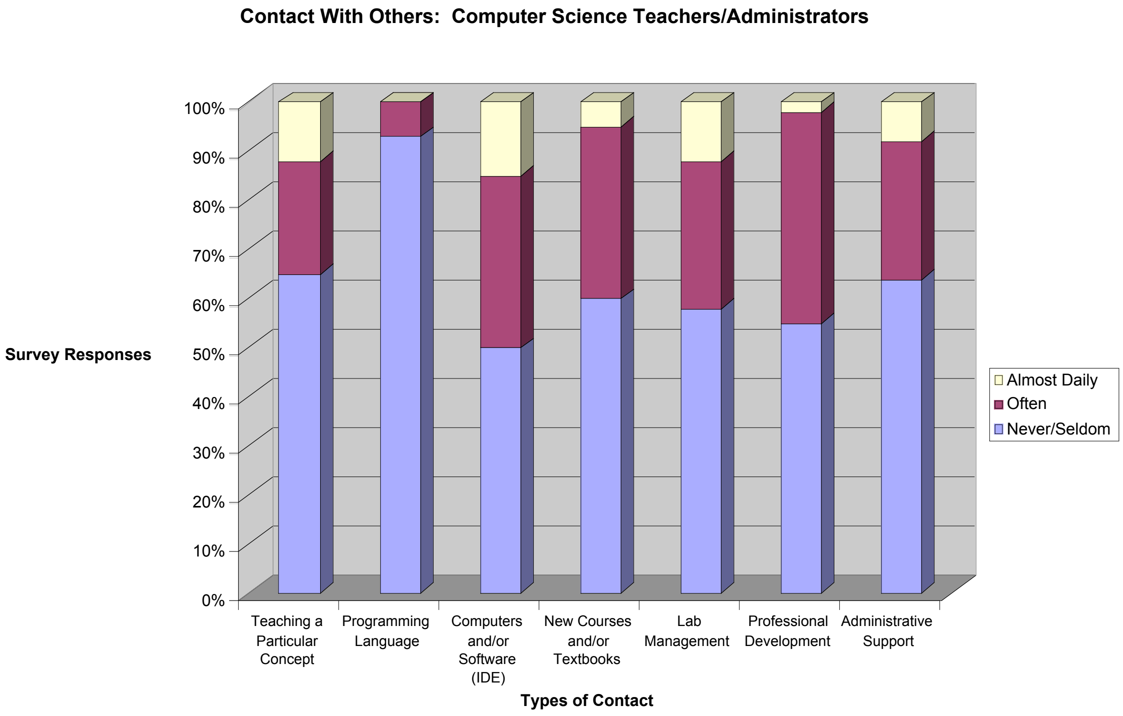
School Year	Programming Languages #	College Now Courses #	Advanced Placement Courses #
2003-2004	**1.435	*.48	*.26
2002-2003	1.36	.46	.32
2001-2002	1.33	.43	.24
2000-2001	1.30	.40	.25
1999-2000	1.30	.40	.28
1999-2004	*1.345*	*.43*	*.27*

**Mean number of courses taught.*
***Mean number of languages taught.*

Figure 1
Percent Responses
Contact with Others: Computer Science Teachers / Administrators

14. How often do you have the following types of contact with other computer science teachers and/or administrators? (*Note: All are referring to computer science topics*.)

	Never/Seldom	**Often**	**Almost daily**
Discussion about teaching a particular concept	❑	❑	❑
Discussion about a programming language	❑	❑	❑
Discussion about computers and/or software (IDE)	❑	❑	❑
Discussion about new courses and/or textbooks	❑	❑	❑
Discussion about lab management	❑	❑	❑
Discussion about professional development	❑	❑	❑
Discussion about administrative support	❑	❑	❑



Teachers rated their teaching beliefs and teaching practice on five point continuums that varied from “Traditional Teacher-centered” to “Constructivist Student-centered.” The mean ratings for each can be found on Figure 2 and Figure 3 respectively. The responses to both questions ranged from two to five on the five-point continuum, with five being the “Constructivist Student-centered”rating. The mean for the teachers’ teaching belief was 3.55 and the mean for the teachers’ teaching practice was 3.63. This indicates that the computer science teaching environment commands a more student-centered classroom.

There was a range of responses when teachers were asked to rate their perceived proficiency as a computer science teacher. None of the teachers venture to the two ends of the continuum, with the responses ranged from two to four on the five point continuum, with one indicating “Help … I’m Drowning” and five indicating “Totally Awesome.” The mean for the teachers’ perceived proficiency was analyzed in an overall participant category and then broken down into four additional subcategories, those specifically teaching computer programming and number of years teaching experience. The mean ratings for each can be found in Figures 4a through 4g. Figure 4h is a column chart that represents the proficiency rating for the overall participant mean and each of the participant sub group means. The mean for the overall teachers’ perceived proficiency was 3.41. This response was given by teachers with a mean teaching experience of 10.2 years in the area of computer science and 16.4 years in overall teaching experience. The computer

Figure 2
Mean Teaching Belief

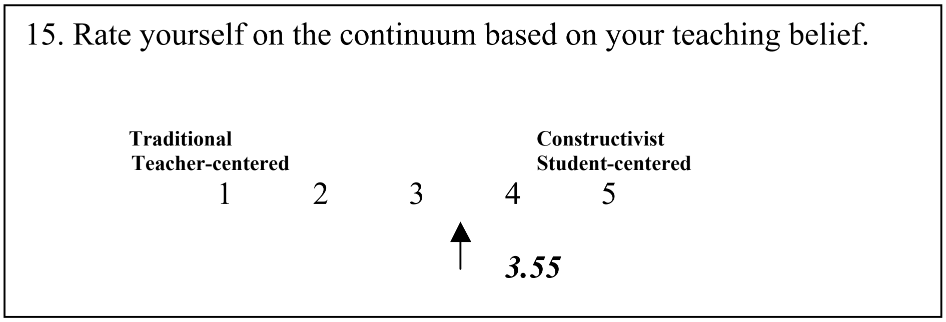

15. Rate yourself on the continuum based on your teaching belief.

Figure 3
Mean Teaching Practice

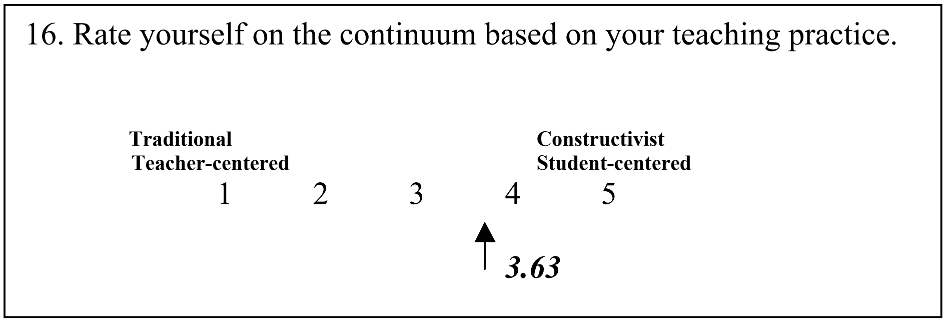

16. Rate yourself on the continuum based on your teaching practice.

Figure 4a
Mean Perceived Proficiency
(all 42 participants)

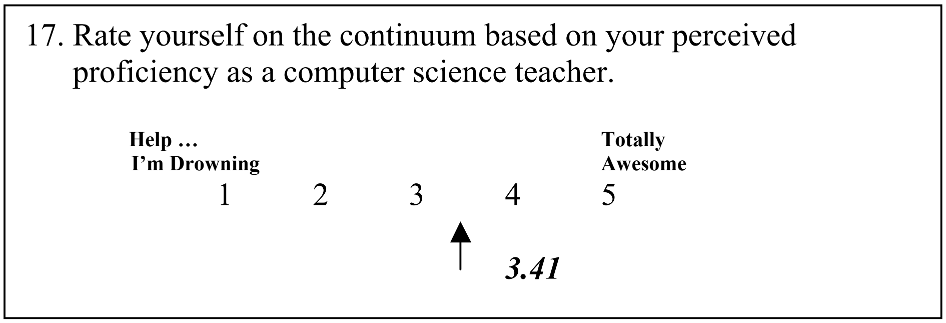

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Figure 4b
Mean Perceived Proficiency
(26 participants teaching computer programming)

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … I'm Drowning				**Totally Awesome**
1	2	3	4	5

↑ ***3.37***

Figure 4c
Mean Perceived Proficiency
(12 participants with 2-8 years teaching experience)

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … I'm Drowning				**Totally Awesome**
1	2	3	4	5

↑ ***3.27***

Figure 4d
Mean Perceived Proficiency
(10 participants with 9-15 years teaching experience)

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … I'm Drowning				**Totally Awesome**
1	2	3	4	5

↑ ***3.20***

Figure 4e
Mean Perceived Proficiency
(7 participants with 16-22 years teaching experience)

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … **I'm Drowning**				**Totally** **Awesome**
1	2	3	4	5

↑ ***3.57***

Figure 4f
Mean Perceived Proficiency
(9 participants with 23-29 years teaching experience)

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … **I'm Drowning**				**Totally** **Awesome**
1	2	3	4	5

↑ ***3.61***

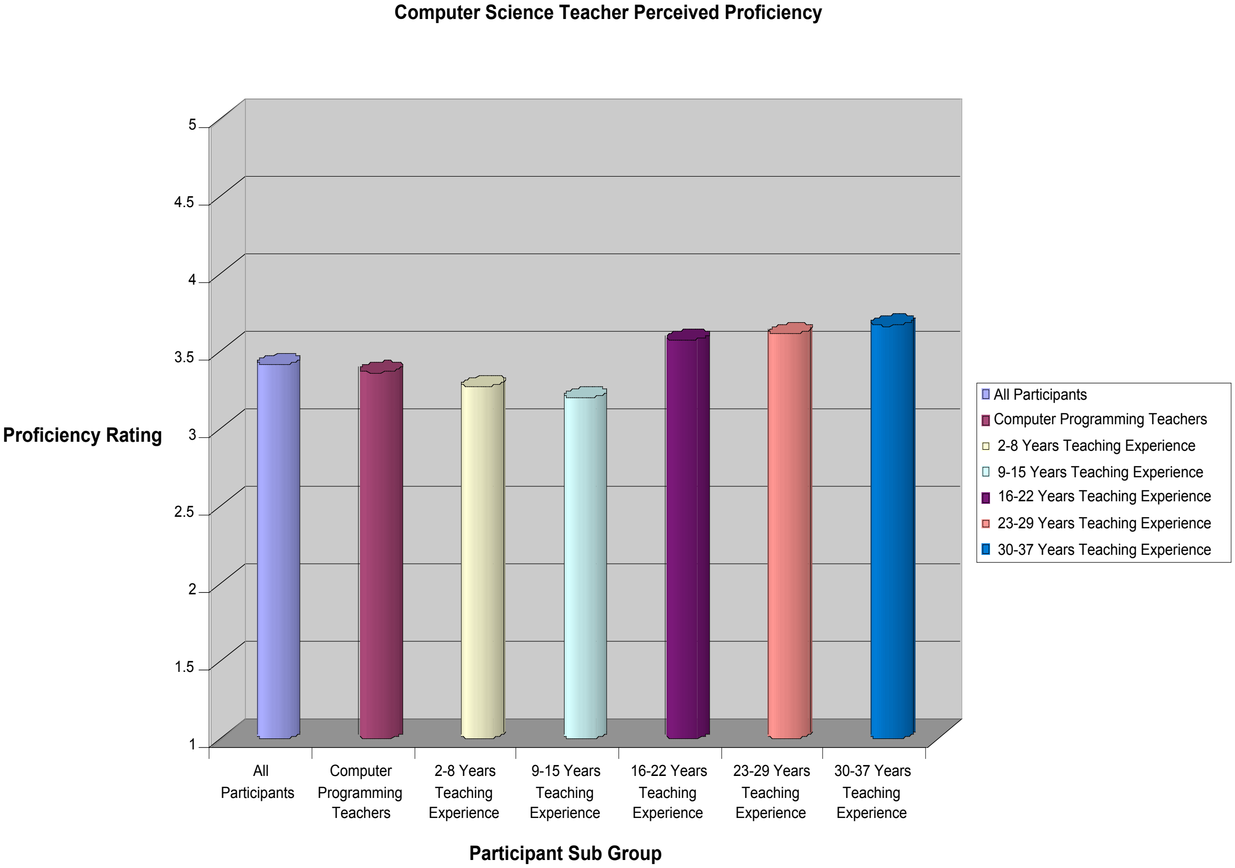
Figure 4g
Mean Perceived Proficiency
(4 participants with 30-37 years teaching experience)

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … **I'm Drowning**				**Totally** **Awesome**
1	2	3	4	5

↑ ***3.67***

Figure 4h



Computer Science Teacher Perceived Proficiency
Proficiency Rating
5
4.5
4
3.5
3
2.5
2
1.5
1
All Participants
Computer Programming Teachers
2-8 Years Teaching Experience
9-15 Years Teaching Experience
16-22 Years Teaching Experience
23-29 Years Teaching Experience
30-37 Years Teaching Experience
Participant Sub Group
All Participants
Computer Programming Teachers
2-8 Years Teaching Experience
9-15 Years Teaching Experience
16-22 Years Teaching Experience
23-29 Years Teaching Experience
30-37 Years Teaching Experience

programming teacher subgroup, twenty-six of the forty-two survey participants, did have a lower perceived proficiency mean at 3.37 than the overall perceived proficiency mean. Participants with two to eight years of teaching experience, twelve of the forty-two survey participants, had a perceived proficiency mean of 3.27. The mean fell to a low of 3.20 for participants with nine to fifteen years teaching experience; ten participants of the forty-two respondents fell into this category. Teachers with sixteen to twenty-two years teaching experience, seven of the forty-two survey participants, had a mean of 3.57 and teachers with twenty-three to twenty-nine years of teaching experience, nine of the forty-two survey participants, had a mean of 3.61. The highest perceived proficiency mean came from participants with thirty to thirty-seven years of teaching experience, four of the forty-two survey participants.

An overwhelming 85% of the teachers said that professional development support would help alter their perceived proficiency rating. Teachers that answered "yes" listed an abundance of professional development ideas that would help them alter their perceived proficiency rating. The professional development requested was compressed into the succinct categories of time, training, networking with others, materials, and funding. Teachers requested more planning time, time to manage grading all the programs students generate, time off to attend professional development, time to network with others, and time to practice concepts that are being taught to the students. Content, language(s), software, technology standards, and lab management training were listed amongst the training needs. Working with others through classes or workshops, brainstorming sessions, and mentoring programs were some of the networking contacts that were included on the survey instrument. Teachers asked for administrative support that would include comprehensive teaching materials, funding for the materials, and funding of professional development.

Awareness and Utilization

Teachers were asked to reflect over their previous two years of teaching in regards to available training and support resources in Part III of the survey instrument. Question nineteen contained ten prelisted training topics and/or specific activities. Respondents were asked to reflect on whether they were aware of, had participated in, and/or had future plans to participate in each of the ten training topics and/or activities. Question twenty contained eleven prelisted support resources. Respondents were asked to reflect on whether they were aware of, had utilized, and/or had future plans to utilize each of the eleven support resources listed. Figure 5 and Figure 6 contain the percent of positive responses for each of the items, available training and support resources, respectively.

In question nineteen, more than 50% of the survey participants responded that they were aware of 60% of the available training items. Only two items, “Integrating Curriculum” and “Programming language(s) (self-trained)”, had more than 50% participation from survey participants. In fact the other eight items only had a maximum participation of 30%. “Integrating Curriculum” and “Programming language(s) (self-trained)” were also the only two items that more

Figure 5
Percent Responses for Available Training

19. Thinking about the last two years, respond to the following regarding available training. Please mark all three areas (awareness, participation, future plans).

	Aware?		**Participated?**		**Future Plans?**	
	Yes	No	Yes	No	Yes	No
Lab management	❑	❑	❑	❑	❑	❑
Integrating Curriculum	❑	❑	❑	❑	❑	❑
Program language(s) (formal-training)	❑	❑	❑	❑	❑	❑
Program language(s) (self-trained)	❑	❑	❑	❑	❑	❑
IDE training (ex. Code Warrior)	❑	❑	❑	❑	❑	❑
Advanced Placement Computer Science Institutes	❑	❑	❑	❑	❑	❑
JETT (Java Engagement for Teacher Training)	❑	❑	❑	❑	❑	❑
Microsoft Certification	❑	❑	❑	❑	❑	❑
Novell Certification	❑	❑	❑	❑	❑	❑
Cisco Certification	❑	❑	❑	❑	❑	❑

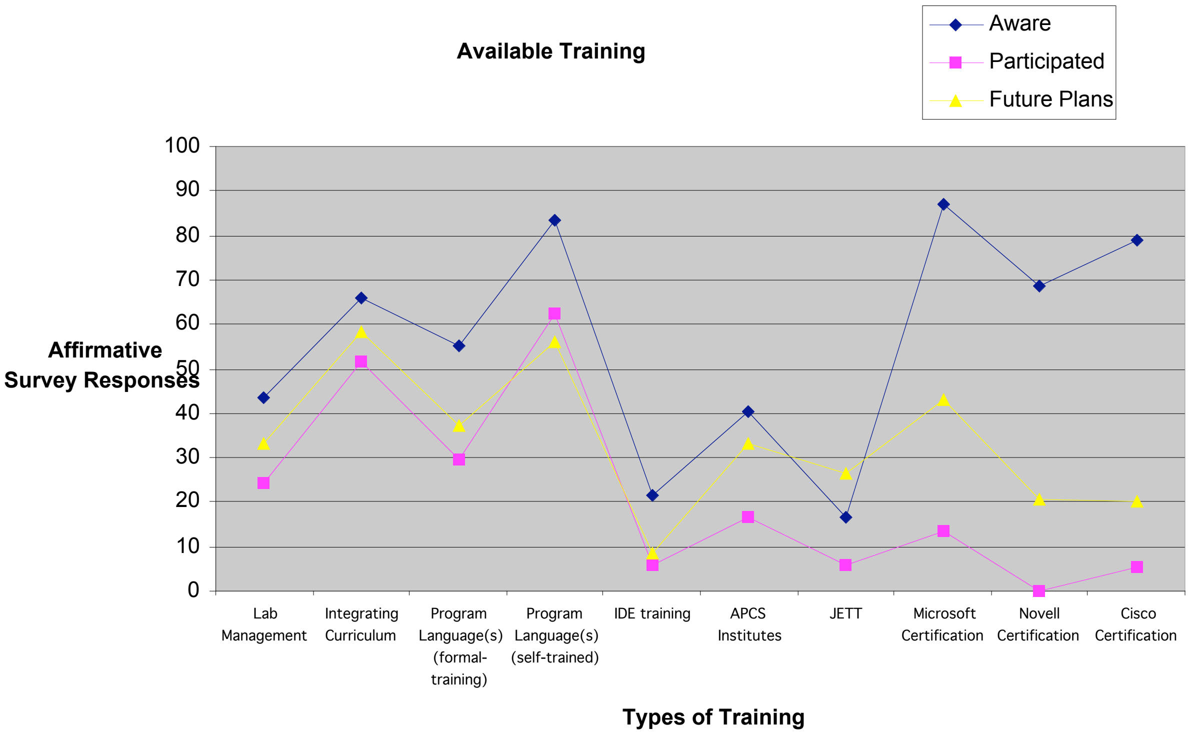
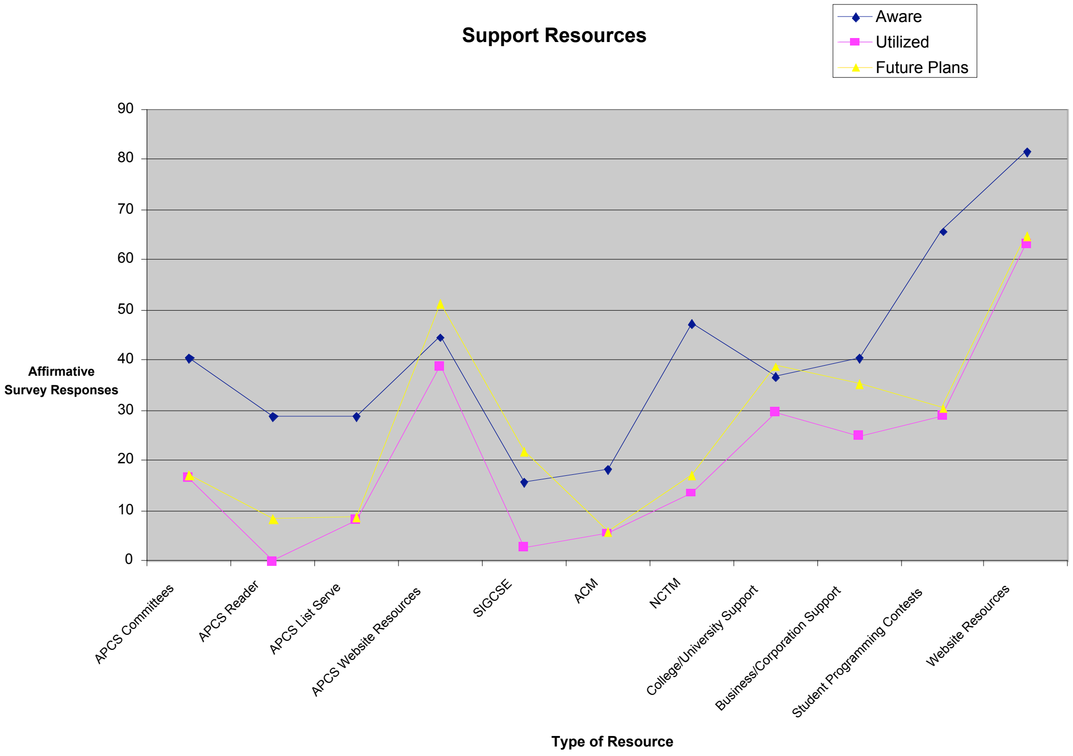


Figure 6
Percent Responses for Support Resources

20. Thinking about the last two years, respond to the following regarding support resources. Please mark all three areas (awareness, utilized, future plans).

	Aware?		**Utilized?**		**Future Plans?**	
	Yes	No	Yes	No	Yes	No
Advanced Placement Computer Science						
Committees	❑	❑	❑	❑	❑	❑
Reader	❑	❑	❑	❑	❑	❑
List Serve	❑	❑	❑	❑	❑	❑
Website Resources	❑	❑	❑	❑	❑	❑
SIGCSE	❑	❑	❑	❑	❑	❑
(Special Interest Group for Computer Science Educators)						
ACM (Association for Computing Machinery)	❑	❑	❑	❑	❑	❑
NCTM (National Council of Teachers of Math)	❑	❑	❑	❑	❑	❑
College/University Support	❑	❑	❑	❑	❑	❑
Business/Corporation Support	❑	❑	❑	❑	❑	❑
Student Programming Contests	❑	❑	❑	❑	❑	❑
Website Resources	❑	❑	❑	❑	❑	❑



than 50% of the respondents had future plans for participation. More teachers (83%) had participated in “Programming language(s) (self-trained)” than the teachers (56%) that had future plans to participate in “Programming language(s) (self-trained).” Teachers were very aware of “Microsoft Certification” (87%), “Novell Certification” (68%), and “Cisco Certification” (79%). The teachers’ future plans were lower than their awareness for these three training areas: “Microsoft Certification” (43%), “Novell Certification” (21%), and “Cisco Certification” (20%). Even fewer of the teachers had participated in these three training areas: “Microsoft Certification” (14%), “Novell Certification” (0%), and “Cisco Certification” (5%). In all categories except two, “Programming language(s) (self-trained)” and “JETT (Java Engagement for Teacher Training),” the teachers’ awareness ranked higher than their plans for future participation and their participation during the last two years ranked lower than awareness and future plans. Since only 17% of the teachers were aware of “JETT (Java Engagement for Teacher Training)” and 26% had future plans for “JETT (Java Engagement for Teacher Training),” it is assumed that the survey influenced these teachers to participate in “JETT (Java Engagement for Teacher Training)” in the future.

Question twenty presented two areas, “Advanced Placement Computer Science Website Resources” and “SIGCSE: Special Interest Group for Computer Science Educators,” where awareness ranked lower than future plans to utilize the resources. In both areas it is assumed that the survey influenced teachers to participate in the future. Teachers were more consistent with their utilization and future plans in five of the eleven elements for support resources. The areas of "Advanced Placement Computer Science Committees," "Advanced Placement Computer Science List Serve," "ACM (Association for Computing Machinery," "NCTM (National Council of Teachers of Mathematics)," "Student Programming Contests," and "Website Resources" received responses that indicate future utilization plans that are consistent with their past two years utilization.

In general, both questions nineteen and twenty showed the most dramatic differences between the awareness and participated/utilized responses. The smallest were the 6% decline with "Advanced Placement Computer Science Website Resources" and 7% decline in "College/University Support." The largest declines, from awareness responses to participated/utilized responses, were the "Cisco Certification" (74%), "Microsoft Certification" (73%), and "Novell Certification" (68%). "Student Programming Contests" (37%) and "NCTM (National Council of Teachers of Mathematics)" (33%) showed a significant difference between awareness and utilization of the support resources. The remaining training and support resources still demonstrated a moderated difference (11% - 29%) between participant awareness and utilization: "Advanced Placement Computer Science Reader" (29%), "Programming language(s) (formal-training)" (25%), "Advanced Placement Computer Science Institutes" (24%), "Advanced Placement Computer Science Committees" (24%), "Advanced Placement Computer Science List Serve" (21%), "Programming language(s) (self-trained)" (21%), "Lab management" (20%), "Website Resources" (19%), "Business/Corporation Support" (16%), "IDE training" (16%), "Integrating Curriculum" (15%), "SIGCSE (Special Interest Group for Computer Science Educators)" (13%), "ACM (Association for Computing Machinery)" (12%), and "JETT (Java Engagement for Teacher Training)" (11%).

In question twenty-one, sixty-two percent of the survey respondents took the time to list factors that would increase their awareness and utilization of professional development opportunities in the area of computer science education. The most frequent and adamantly requested factor was time. More than 50% of the participants requested more preparation time and more training time. Many went on to request that this time be released time from work instead of during their time off from teaching contracts. It also was mentioned that time was needed to seek funding. Other items specifically requested by participants that responded to question twenty-one was funding (38%), skills and confidence (30%), and access (30%). A few participants detailed administrative support, contact with others, flexibility, and a computer science organization for secondary teachers in their responses.

CHAPTER 5

SUMMARY AND CONCLUSIONS

Summary

Secondary computer science teachers across the nation are faced with many challenges that do not exist for teachers of other subjects. Many of these teachers have received minimal or no formal training in computer science, an area of study that is relatively young and continually evolving (Stephenson, 2002; Gal-Ezer, 1995; Schwill, 1997; Tucker 1996). A deficiency in professional development, including teacher preparation, staying current, and professional support, exists for secondary computer science teachers (Stephenson, 2002; Tucker, 1996). Curriculum issues involving local/state/national standards, teaching methods, and languages have previously been inadequate in providing support to secondary computer science teachers (Stephenson, 2002; Tucker, 1996; Association for Computing Machinery, 2002). Within the last year, the issues of professional development and curriculum are beginning to be addressed with the recent development of the Computer Science Teachers Association (ACM, 2005), the National Education Technology Plan (U.S. Department of Education, 2005), the National Education Technology Standards (ISTE, 2004), the Standards for Secondary Computer Science Education (ISTE, 2004), and the final report for a model curriculum for K-12 computer science (Tucker et al., 2004).

The purpose of this study was to research high school computer science teachers' perceived proficiency and professional development awareness and utilization. Specifically, the research questions were:

1. How do Kansas high school computer science teachers perceive their proficiency in teaching computer programming?
2. Is there a relationship between awareness and utilization of professional development opportunities for Kansas high school computer science educators?

A survey was given to high school computer science teachers across the state of Kansas. These teachers were Kansas certified personnel of computer studies and computer programming in 2002-2003. A total of 42 teacher surveys were returned and analyzed. The survey contained four sections. Section one contained precursor questions and requested general teacher and school information. Section two focused on teaching experience, such as teaching background, teaching endorsements, types of courses and languages being taught, and course load specifics. Teaching environment, which included a self perceived proficiency rating, was the focus of section three. Participants also listed types of professional development that would help them alter their perceived proficiency rating. Section four pertained to professional development awareness and utilization. The ending focal point for section four and the survey requested teachers to list what would help increase their awareness and utilization of professional development opportunities in the area of computer science education.

Participant data indicated that teachers with a computer studies or computer programming endorsement also had an endorsement from another area of teaching, usually mathematics or business. Data also identified that three areas of computer science were being taught: computer programming, computer technology, and computer literacy.

Results showed that the mean for the overall teachers' perceived proficiency was 3.41, on a five point continuum, with one indicating "Help … I'm Drowning" and five indicating "Totally Awesome." This response was given by teachers with a mean teaching experience of 10.2 years in the area of computer science and 16.4 years in overall teaching experience. Sixty-two percent of the respondents, the computer programming teachers, had an even lower perceived proficiency of 3.37. Twenty-four percent of the participants had nine to fifteen years of teaching experience and their proficiency rating sank to 3.20. Even teachers with thirty to thirty-seven years of teaching experience had a lower than expected proficiency rating of 3.67.

The participants had a general lack of awareness for computer science professional development opportunities. Only 38% of the computer science professional development opportunities listed in the survey had an awareness factor of more than 50% of the survey participants. When participants were aware of opportunities, their participation/utilization of these professional development opportunities was evident. In general, the participation/utilization of professional development opportunities was significantly lower, an average of 28% lower, than awareness of professional development opportunities. In sixty-seven percent of the professional development opportunities listed in the survey, future plans for participation/utilization were greater than previous participation/utilization experiences. Teachers surveyed consistently and adamantly reported that levels of awareness and utilization of professional development opportunities would increase with additional resource time. An overwhelming 85% of the teachers said that professional development support would help alter their perceived proficiency rating.

Conclusions

The results of this study support the following two conclusions.

1. High school computer science teachers that are provided additional resource time will increase levels of awareness and utilization of professional development opportunities.

2. Professional development support would help alter high school computer science teachers' perceived proficiency rating.

REFERENCES

Association for Computing Machinery. (1993). ACM model high school computer science curriculum. *Communications of the ACM, 36*(5), 86(4).

Association for Computing Machinery. (1997). Members who love to teach. *Communications of the ACM, 40*(5), 14.

Association for Computing Machinery. (2002). ACM k-12 task force research results. *K-12 Task Force*. Retrieved January 10, 2005, from http://www.acm.org/education/k12/research.html

Association for Computing Machinery. (2003). Professional Development. *K-12 Task Force.* Retrieved April 23, 2005, from http://www.acm.org/education/k12/profdev.html

Association for Computing Machinery. (2004). About JETT. *JETT Information.* Retrieved January 10, 2005, from http://jett.acm.org/jettinfo/moreinfoindex cfm?CFID=3086240&CFTOKEN=47582747

Association for Computing Machinery. (2005). *Computer science teachers association*. Retrieved April 23, 2005, from http://csta.acm.org/

Drysdale, S., Hromcik, J., Weiss, M. A. (2003). Java in the morning … java in the evening … java in 2004. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (pp. 271-272). New York, NY: Association for Computing Machinery.

Elkner, J. (2000). *Using python in a high school computer science program.* Retrieved February 14, 2001, from http://www.python.org/workshops/2000-01/proceedings/papers/elkner/pyYHS.html

Fisher, A. L. (1997). *EPWG: Integrating gender equity training and teach retooling for the high school computer science classroom.* Retrieved February 14, 2001, from http://www.education.siggraph.org/nsfcscr/projects/intro/fisher.html

Gal-Ezer, J. (1995). Computer science teachers' certification program. *Computers and Education, 25*(3), 163-168.

Gal-Ezer, J. & Harel, D. (1998). What (else) should cs educators know? *Communications of the ACM, 41*(9), 77-84.

Garrett, T. (2002). Writing the successful computer lesson. *Techniques - Association for Career and Technical Education, 77*(3), 34-36.

Harvey, B. (n.d.), *Symbolic programming vs. the A.P. curriculum.* Retrieved September 18, 2002 from http://www.cs.berkeley.edu/~bh/bridge.html

International Society for Technology in Education. (2004). *ISTE/NCATE standards for education technology programs.* Retrieved April 23, 2005, from http://cnets.iste.org/ncate/

International Society for Technology in Education. (2004). *National education technology standards project.* Retrieved April 23, 2005, from http://cnets.iste.org

Lapidot, T. & Hazzan, O. (2003). Methods of teaching a computer science course for prospective teachers. *SIGCSE Bulletin inroads, 35*(4), 29-34. New York, NY: Association for Computing Machinery.

Lidtke, D. K. & Moursund, D. (1993). Computers in schools: past, present, and how we can change the future. *Communications of the ACM, 36*(5), 84(4).

Lister, R. (2003). The five orders of teaching ignorance. *SIGCSE Bulletin inroads, 35*(4), 16-17. New York, NY: Association for Computing Machinery.

National Science Foundation (2005). *New approaches for new times.* Retrieved April 24, 2005 from http://www.nsf.gov/about/history/nsf0050/education/newapproaches.htm

Schwill, A. (1997). Fundamental ideas—rethinking computer science education. *Learning and Leading with Technology, 25*(1), 28-31.

Stephenson, C. (2002). Computer science education: Looking back and looking ahead. *Learning and Leading with Technology, 30*(2), 6-9, 44-45.

Thomas, R. A. & Upah, S. C., Jr. (1996). Give programming instruction a chance. *Journal of Research on Computing in Education, 29*, 96-108.

Tucker, A. B. (1996). Strategic directions in computer science education. *ACM Computing Surveys, 28*(4).

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). Toward a k-12 computer science curriculum. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (pp. 305-306). New York, NY: Association for Computing Machinery.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2004). *A model curriculum for k-12 computer science: Final report of the acm k-12 task force curriculum committee.* New York, NY: Association for Computing Machinery.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2004). Toward a k-12 computer science curriculum. *Implementation challenges for a k-12 computer science curriculum* (pp. 305-306). New York, NY: Association for Computing Machinery.

United States Department of Education, Office of Educational Technology (2004). *National education technology plan.* Retrieved April 23, 2005, from http://www.nationaledtechplan.org

Weinstein, P., & Resnick, M. (1999). Computer programming revisited. *Technology and Learning, 19*(8), 38-42.

Appendix A

TEACHER SURVEY

Please answer each of the following items by filling in the blank or checking the appropriate box.

1. Age: ___________ Gender: ___________

2. Enrollment in your high school (2003-2004): ____________

3. Enrollment in your computer science courses (2003-2004): ____________

Part I: Teaching Experience

4. Total number of years teaching (including 2003-2004): ____________

5. Number of years teaching high school computer science (including 2003-2004): ____________

6. What are your current teaching endorsements? _________________________
____

7. Do you currently teach a computer science class? Y □ N □

   If yes, please specify the course(s) and number of years taught.

   Course # Years Taught *(including 2003-2004)*

   ____
   ____

8. Have you taught other computer science classes, not listed in #7, during the previous 4 years? Y □ N □

   If yes, please specify the course(s) and number of years taught.

   Course *# Years Taught*

   ____
   ____

9. In the past 5 years, what different computer programming languages have you taught? ____
____

10. How many course preparations, new preparations, and courses with new textbooks have you taught for each of the last 5 years (including non-computer science classes)?

School Year	*Overall Course Preparations #*	*New Course Preparations #*	*Courses with New Textbooks #*
2003-2004			
2002-2003			
2001-2002			
2000-2001			
1999-2000			

11. How many different computer programming languages, College Now courses, and Advanced Placement courses have you taught for each of the last five years (only computer science classes)?

School Year	*Programming Languages #*	*College Now Courses #*	*Advanced Placement Courses #*
2003-2004			
2002-2003			
2001-2002			
2000-2001			
1999-2000			

12. Have you taught computer science classes other than the high school level? (i.e. community colleges, trade schools, colleges, etc.) Y □ N □

If yes, please specify the class, location, and dates.

Course *Institution* *Dates*

____

____

____

13. Have you worked in the computer science field outside of the educational field? Y □ N □

If yes, please specify the position, company, and dates.

Position *Company* *Dates*

____

____

____

Part II: Teaching Environment

14. How often do you have the following types of contact with other computer science teachers and/or administrators? (*Note: All are referring to computer science topics*.)

	Never/Seldom	**Often**	**Almost daily**
Discussion about teaching a particular concept	□	□	□
Discussion about a programming language	□	□	□
Discussion about computers and/or software (IDE)	□	□	□
Discussion about new courses and/or textbooks	□	□	□
Discussion about lab management	□	□	□
Discussion about professional development	□	□	□
Discussion about administrative support	□	□	□

15. Rate yourself on the continuum based on your teaching belief.

Traditional **Teacher-centered**				**Constructivist** **Student-centered**
1	2	3	4	5

16. Rate yourself on the continuum based on your teaching practice.

Traditional **Teacher-centered**				**Constructivist** **Student-centered**
1	2	3	4	5

17. Rate yourself on the continuum based on your perceived proficiency as a computer science teacher.

Help … **I'm Drowning**				**Totally** **Awesome**
1	2	3	4	5

18. Would professional development support help you alter the proficiency rating you gave in #17? Y □ N □

If yes, what professional development would help? ______________________

____

____

____

____

____

____

____

Part III: Awareness and Utilization

19. Thinking about the last two years, respond to the following regarding available training. Please mark all three areas (awareness, participation, future plans).

	Aware?		Participated?		Future Plans?	
	Yes	No	Yes	No	Yes	No
Lab management	□	□	□	□	□	□
Integrating Curriculum	□	□	□	□	□	□
Program language(s) (formal-training)	□	□	□	□	□	□
Program language(s) (self-trained)	□	□	□	□	□	□
IDE training (ex. Code Warrior)	□	□	□	□	□	□
Advanced Placement Computer Science						
Institutes	□	□	□	□	□	□
JETT (Java Engagement for Teacher Training)	□	□	□	□	□	□
Microsoft Certification	□	□	□	□	□	□
Novell Certification	□	□	□	□	□	□
Cisco Certification	□	□	□	□	□	□

20. Thinking about the last two years, respond to the following regarding support resources. Please mark all three areas (awareness, utilized, future plans).

	Aware?		Utilized?		Future Plans?	
	Yes	No	Yes	No	Yes	No
Advanced Placement Computer Science						
Committees	□	□	□	□	□	□
Reader	□	□	□	□	□	□
List Serve	□	□	□	□	□	□
Website Resources	□	□	□	□	□	□
SIGCSE	□	□	□	□	□	□
(Special Interest Group for Computer Science Educators)						
ACM (Association for Computing Machinery)	□	□	□	□	□	□
NCTM (National Council of Teachers of Math)	□	□	□	□	□	□
College/University Support	□	□	□	□	□	□
Business/Corporation Support	□	□	□	□	□	□
Student Programming Contests	□	□	□	□	□	□
Website Resources	□	□	□	□	□	□

21. What would help increase your awareness and utilization of professional development opportunities? (i.e. access, flexibility, funding, administrative support, confidence, skills, preparation time, instructional time)

____

____

____

Additional Comments

Appendix B
Consent Letter for the Teacher Survey

Dear High School Computer Science Teacher,

I am completing my master's degree at the University of Kansas. I also teach computer science in the Shawnee Mission School District. The research community has yet to actively consider the teaching environment and professional development opportunities that exist for high school teachers of the ever-evolving discipline of computer science. For my thesis, I am researching whether high school computer science teachers feel equipped to teach their programming classes and what professional development opportunities they are aware of, receive support for, and utilize. Your responses will provide some important data in this area of study.

The Department of Teaching and Leadership at the University of Kansas supports the practice of protection for human subjects participating in research. The following information is provided for you to decide whether you wish to participate in the present study. Your participation is solicited although strictly voluntary. You may discontinue your involvement at any time without further obligation. The survey is completely anonymous; your name will not be associated with the research findings in any way and only a code number will identify the information. By returning this survey, you are indicating your consent to participate in this study.

I am asking Kansas high school computer science teachers to complete the enclosed survey. I hope that you will choose to participate in this study by completing the survey and returning it to me in the enclosed stamped envelope by Tuesday, November 25, 2003.

If you would like any additional information regarding this study before or after it is complete, please feel free to contact me by phone, e-mail, or mail. Your time and effort are greatly appreciated. Thank you.

Sincerely,

Deb Pontier
Shawnee Mission East High School
7500 Mission Road
Prairie Village, Kansas 66208
(913) 993-6768
debbrapontier@smsd.org

Dr. Susan Gay
Faculty Advisor
University of Kansas
Department of Teaching and Leadership
341 Joseph R. Pearson Hall
1122 West Campus Road
Lawrence, Kansas 66045-3101
(785) 864-9676
sgay@ku.edu

Appendix C

Shawnee Mission School District Computer Science Mastery Objectives

Computer Science Mastery Objectives
Shawnee Mission Schools

Strand	Topics in Computer Science	Programming in Java	Programming in Java, Advanced	Advanced Placement Computer Science
Program Execution Students will enter, modify and compile programs	• Modify and compile an existing program.	• Modify and compile an existing program.		
Algorithms Students will construct, trace, and/or modify algorithms to solve selected problems.	• Know and apply the process for writing Visual Basic programs. • Solve problems using an object-oriented, event-driven approach.	• Construct original algorithms in Java to solve given problems. • Trace the sequence of an algorithm to predict its outcome. • Modify algorithms in Java to extend their usefulness or solve similar problems.	• Construct original algorithms at a difficulty level appropriate for the course to solve given problems. Practice problem analysis. • Represent algorithms using pseudocode. • Verify algorithms by hand and test data. • Modify more complex algorithms to extend their usefulness or solve similar problems.	• Combine multiple algorithms to solve problems. • Determine appropriate use of recursion. • Use "big-O" notation for analyzing algorithms.
Program Development Students will use a variety of techniques for designing, testing and correcting programs.	• Identify the elements in the Visual Basic environment. • Create an appropriate user interface.	• Trace the execution of a program, either by hand or by using the computer, to find the source of an error. • Create and use data to test the possible program alternatives. • Interpret error messages given by the computer to make program corrections. • Divide a problem into	• Pass arguments between modules. • Differentiate between value and reference parameters. • Determine the scope of an identifier. • Design input routines that will accept only specified data. • Display appropriate error messages for incorrect data input.	• Use incremental development (top-down or bottom-up) as appropriate.

Strand	Topics in Computer Science	Programming in Java	Programming in Java, Advanced	Advanced Placement Computer Science
		specific modules.	• Design a routine to display appropriate prompt for user input. • Describe and practice program analysis including strategies such as step-wise refinement, solution by analogy, use of previously developed subalgorithms.	
<u>Control Structures</u> Students will demonstrate knowledge of the control structures used in object-oriented or structured programming.	• Work with control buttons in Visual Basic. • Use if/then structures in program development. • Use loop structures in program development.	• Create or follow a sequence of program statements or instructions in linear programs. • Identify and use a decision structure that allows one of two alternatives. • Use logical operators such as "and" and "or." • Write programs that incorporate both counting and terminating loop structures. • Identify and use nested loops and decision structures. • Identify and use a "cases" or multiple decision structure.	• Select and use the correct control structure for a given problem including input, output, assignment, conditional, looping, subalgorithm calls.	• Use appropriate recursive techniques to solve problems.
<u>Object-Oriented Programming</u>		• Use features of an OOP language including data types, structured types, block structure and scope of identifiers. • Use expressions and assignment statement appropriately.	• Use features of an OOP language appropriately including data types, scope, expressions, assignment. • Input and output operations and formatting to the screen,	

Strand	Topics in Computer Science	Programming in Java	Programming in Java, Advanced	Advanced Placement Computer Science
		- Input and output operations and formatting to the screen, printer, disk files. - Use control structures: sequence, selection, iteration.	printer, disk files. - Use control structures: sequence, selection, iteration.	
Documentation Students will document problem solutions to provide information which assists others in interpreting, evaluating, using, and modifying the solution.	- Provide a brief statement of what the program is supposed to accomplish. - Provide written identification for each variable and its use in a program. - Use a clear and consistent program format. - Include sufficient comments to clarify and describe a statement, structure, or module.	- Provide a brief statement of what the program is supposed to accomplish. - Provide written identification for each variable and its use in a program. - Use a clear and consistent program format. - Include sufficient comments to clarify and describe a statement, structure, or module.	- Include sufficient documentation to describe all aspects of a program.	- Write external documentation for a program appropriate for a user's manual.
String Processing			- Use substrings of a given string. - Concatenate strings. - Use the appropriate functions to determine the length of a string. - Assign character data to a string variable.	
Output Students will organize output in a functional, aesthetically pleasing format.	- Apply Visual Basic properties to develop appropriate output.	- Design programs so that output is labeled correctly. - Write programs to produce appropriately formatted output. - Design programs to output data with specified number of digits and decimal	- Design programs so that output conforms to a given format.	

Strand	Topics in Computer Science	Programming in Java	Programming in Java, Advanced	Advanced Placement Computer Science
		places. • Design programs to output data in multiple columns with formatted column headings.		
Data Structures Students will use a variety of data structures.			• Use one-dimensional arrays for data storage, manipulation, and retrieval. • Use the array as a lookup table. • Use an array as a series of counters or sums. • Write programs using multidimensional arrays. • Create, retrieve, and modify random access and sequential files containing structures of multiple fields. • Create and utilize user-defined data types.	• Use lists, stacks, queues and trees in the development of programs. • Implement linked data structures in programs.
Data Manipulation	• Use formulas in assignment statements. • Find the total number of items in a given data set. • Find the sum of a given data set. • Round a given number to a given number of decimal places. • Use built-in functions.	• Use standard data types appropriately. • Find the total number of items in a given data set. • Find the sum of a given data set. • Round a given number to a given number of decimal places. • Use built-in functions. • Use a random-number generator to model events and predict outcomes.	• Use a variety of search algorithms (e.g. sequential, binary) to locate given data items in a list. • Write a variety of sort algorithms (e.g. insertion, selection, bubble) to sort a list of data.	• Apply advanced search and sort routines. • Use infix, prefix and postfix notation. • Determine the search/sort routine most efficient for a given situation. • Analyze a program and its data for most efficient method of sorting.
Computer Systems Students will identify	• Identify major hardware components of a	• Identify major hardware components of a	• List and describe hardware and software	• Describe the types of memory available and

Strand	Topics in Computer Science	Programming in Java	Programming in Java, Advanced	Advanced Placement Computer Science
components of computers systems and the responsible use of these systems.	computer. • Demonstrate responsible use that respects the legal rights of others.	computer. • Demonstrate responsible use that respects the legal rights of others.	components.	the role of each type in processing data. • Describe how computer memory is organized to support different data types.