A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States

JUDITH GAL-EZER, Open University of Israel
CHRIS STEPHENSON, Computer Science Teachers Association

This article tells a story of K-12 computer science in two different countries. These two countries differ profoundly in culture, language, government and state structure, and in their education systems. Despite these differences, however, they share the pursuit of excellence and high standards in K-12 education. In Israel, curriculum is determined at the national level. The high-school computer science curriculum has been in place for more than 20 years and is offered in all schools as an elective similar to biology, chemistry, and physics. The picture in the United States is more complex and therefore less amenable to generalization. Because educational policy is set at the state and sometimes even at the school district level, access to computer science courses and the content of those courses can vary even for schools within the same district. This article will describe the development of the curricula/standards in both countries and the current situation, focusing on common issues and challenges in areas such as equity and teacher training.

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

General Terms: Standardization

Additional Key Words and Phrases: Computer science, computer science teachers, computer science educators

ACM Reference Format:
Gal-Ezer, J. and Stephenson, C. 2014. A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. *ACM Trans. Comput. Educ.* 14, 2, Article 8 (June 2014), 18 pages.
DOI:http://dx.doi.org/10.1145/2602483

1. BACKGROUND

1.1. K-12 Computer Science Education—Is It Important?

Recently there has been considerable discussion with regard to the importance of computer science education and more particularly its lack of presence in pre-college education. This article briefly explores the current situation with regard to computer science education at the middle and secondary school levels in two countries (Israel and the United States) and explores the commonalities and differences in how computer science is viewed and the extent to which it is perceived to be part of the fabric of student learning.

In considering the role that computer science currently plays and will continue to play in scientific innovation, the authors of the "Towards SCIENCE 2020" report [Microsoft Research 2005] summed up the views of 30 scientists from many countries and various scientific disciplines, when they noted: "we believe computer science is

Author's address: J. Gal-Ezer, email: galezer@cs.openu.ac.il.

DOI:http://dx.doi.org/10.1145/2602483

poised to become as fundamental to biology as mathematics has become to physics" (p. 8). Although much had been previously written about computer science, the importance of this report is that it indicated a fundamental shift from computers supporting scientists to "do" traditional science to computer science becoming embedded into the very fabric of science and how science is done. In essence this report pointed to a new kind of science.

The report also considered the specific role of computer science in education and pointed to the following critical needs.

(i) Take far bolder measures to interest children in science and then retain their interest in it and its importance for society.
(ii) Urgently and dramatically improve the teaching of mathematics and science in schools.
(iii) Make teaching of computing more than just IT classes and how to use Power Point. Make the *principles* of computer science, such as abstraction and codification, a core part of the science curriculum.

The CSTA K-12 Computer Science Standards [Seehorn et al. 2011] also provide a powerful argument for computer science as a core discipline, noting that a fundamental understanding of its core concepts enables students to be not merely consumers of technology, but innovative creators of computing systems and developers of algorithmic solutions to unsolved problems. Problem solving lies at the heart of computer science. Learning and doing computer science requires students to state problems clearly and unambiguously, write an algorithmic solution for the problem that takes into account all boundary conditions (robustness), determine that the algorithm produces the right answer (correctness), and test that the solution is efficient (complexity considerations). At each step in this process, students are learning core skills that will serve them well in any field in which they choose to study.

Computer science also has powerful links to all other scientific and many humanistic fields. Many, if not most, of today's scientific breakthroughs are enabled through the power of computing and the work of computer scientists. Today's complex problems and solutions require teams with diverse knowledge and skills covering multiple domains, and an understanding of the fundamentals of computer science can enhance critical elements of problem analysis and solution design and implementation [Wing 2006].

The invention of computers in the 20th century has dramatically changed the way we live and work. Computing is now ubiquitous. It facilitates, shapes, and enhances most aspects of our lives. And while it is difficult to predict the future, it is safe to surmise that individuals who do not understand computing or how to harness its power for human good are far more likely to be excluded from its potential benefits. For this reason, many of us in many countries are asking serious questions regarding the teaching of computer science in our public education systems. Are these systems adequately preparing our students to understand and exploit the power of computing or leaving them to be among the exploited?

1.2. High School Computer Science Curricula Development: A Brief History

There has been a considerable activity surrounding university-level computer science curricula, but until recently there has been little focus on rigorous computing at the secondary school level [Tucker et al. 2006]. For many years high school computing has centered on courses in computer literacy or applications, that is, on the use of computers. And when computing has been taught in high schools as a so-called scientific discipline, the emphasis has often been on programming languages and coding.

It is important to note, however, that there have been pioneering works that have focused on actual computer science learning. These include (among others) the ACM

Model High School Computer Science Curriculum, published by Merrit et al. [1994] and the publication entitled “A High-School Program in Computer Science” [Gal-Ezer et al. 1995]. In Canada, a comprehensive computer science and computer engineering curriculum was defined and implemented for all secondary schools in Ontario [Province of Ontario Ministry of Education 1999, 2000]. The first edition of ACM’s Model Curriculum for K-12 Computer Science was published in 2003 and revised in 2006 [Tucker et al. 2006]. This document made a significant contribution to computer science education in the US by providing practical guidelines for educators and a framework that could be used to integrate computer science fluency and competency throughout primary and secondary schools. When ACM formed the Computer Science Teachers Association in 2004, CSTA took on the task of continually revising and updating these guidelines culminating with the most recent the publication of the CSTA K-12 Computer Science Standards [Seehorn et al. 2011].

More than ten years before the “Towards SCIENCE 2020” report was published [Microsoft Research 2005], the Israeli Ministry of Education professional computer science committee [Gal-Ezer et al. 1995], committed to placing computer science in the high school curriculum on par with physics, chemistry and biology. First published in 1995, the Israeli curriculum has been updated several times by the Ministry of Education professional committee. The most recent revision occurred in 2012. This version was not published in any professional journals or conferences. Recently, however, it has been made available in Hebrew, on the Ministry of Education website [Israeli Ministry of Education 2013]. In 2012 the Israeli Ministry of Education also launched a program to enhance science and technology education. This six-year program covering grades seven through twelve introduces a new computer science curriculum for middle schools [Zur-Bargury, 2012; Zur-Bargury et al. 2012, 2013].

Many other parts of the world, including Europe, the United Kingdom [Burns 2013], New Zealand, and Australia, have also been reviewing and updating their computing standards with a goal of increasing computer science/informatics content and rigor. For example, a recent report entitled “Informatics Education: Europe Cannot Afford to Miss the Boat,” by the Informatics Europe and ACM Europe Working Group on Informatics Education [2013, p. 3], noted that “Informatics education, unlike digital literacy education, is sorely lacking in most European countries... Not offering appropriate informatics education means that Europe is harming its new generation of citizens, educationally and economically”.

Despite shared concerns, there are a variety of approaches towards teaching computer science in schools. In additional to differences in language of instruction and educational contexts, there are also differences in terminology (computer science, computing, informatics) and pedagogical approaches. To address this diversity, the ITiCSE2011 working group produced a basis for a conceptual framework—the Darmstadt Model— that on one hand highlights how varied the situation of computer science education in different countries can be, and on the other hand provides a common basis for discussions by the computer science education community [Hubwieser et al. 2011]. Unfortunately, to date there has been no comprehensive research effort to provide a comparison of high school computer science curricula or their implementation across a variety of countries.

2. TWO COUNTRIES—TWO DIFFERENT EDUCATIONAL SYSTEMS

To understand the approach towards computing education and how a curriculum is designed and implemented in a certain country, it is helpful to have insight into that particular education system. In the following sections we briefly describe the education systems in Israel and the United States.

2.1. Israel

The Israeli education system is a centralized one. The Ministry of Education determines the educational policy on all levels, from Kindergarten to higher education, and implements this policy with the assistance of professional committees and professional supervisors. The school system is divided into three stages: six years of elementary school (grades 1 through 6), three years of middle school (grades 7 through 9), and three years of high school (grades 10 through 12). The first ten years of education are mandatory. Study material is divided into units of approximately 90 hours; some of the subjects come in various levels, the most common being three-unit and five-unit programs. These two modes differ in quantity and depth and sometimes, as is the case in mathematics, they also differ in content.

The three years of high school culminate in a set of matriculation exams. The exams are based on a core of required subjects, which currently include Hebrew (language and literature), English as a second language, Bible studies, and mathematics. In addition there is an extensive list of electives including physics, chemistry, biology, and courses in humanities and social sciences. The matriculation exams are crucial for admission to most Israeli universities. Some higher education institutions give bonus points for students who successfully complete certain subjects.

2.2. The United States

The US education system is decentralized. Decisions with regard to essential educational matters such as learning standards, curriculum, and teacher certification requirements are distributed across the national, state, and even the district and school level. As a result, education policy is highly inconsistent and the teaching and learning experience can differ markedly from school to school.

Although there are exceptions in some systems, schooling is divided into three stages: elementary or primary school (Kindergarten to grade 5), middle school (grades 6 through 8) and high school (grades 9 through 12). Like many other aspects of public education, the ages for compulsory education differ from state to state, beginning between five and eight years of age (with five being the most common age) and ending between 14 and 18 years of age. Students tend not to specialize at the secondary school level and instead take a combination of required and elective courses. Traditionally the required courses include English, math, physical education, science, and social science.

In most states, students must meet a set of requirements before they can graduate from high school but these requirements vary from state to state and school to school. The requirements can include successful completion of state-level end-of-course and end-of-program assessments (exams) as well as the completion of a required set of courses. In some states, for example, students must successfully complete four mathematics and four science courses in order to graduate. The requirements for university admission are equally diverse and do not always match state graduation requirements because individual institutions or even programs within those institutions can set their own admissions guidelines.

Where computer science is concerned, it is fair to say that it has been largely ignored by the public education sector. Instead, responsibility for promoting and supporting the teaching of computer science in schools has fallen almost exclusively to individual researchers and professional associations (such as the International Society for Technology Education in the 1980s and 1990s and CSTA since 2004). Although there are current efforts underway to standardize student learning through programs such as the Common Core State Standards initiative[1], these efforts are

[1]http://www.corestandards.org

almost exclusively focused on so called "core" academic subjects and do not include computer science.

To deal with this lack of consistency in student learning, ACM published the first version of A Model Curriculum for K–12 Computer Science in 2003 and a revised version in 2006 [Tucker et al. 2006]. In 2011, CSTA completed a review and public feedback process for the K-12 standards and published the CSTA K–12 Computer Science Standards [Seehorn et al. 2011]. To support the implementation of these standards, CSTA also created a repository of resources (csta.acm.org) and facilitated professional development (PD) events for teachers.

3. K-12 CURRICULUM—PRINCIPLES

3.1. Israel

Computer science has been an academic subject in Israeli high schools since the mid 1970s. At that time, however, it was not yet a fully accepted scientific subject, few high schools offered computer science courses, and the universities did not consider computer science as important as other sciences taken in high school. The mid-70s curriculum included a detailed course in Basic programming, as well as several elective units. Course material was not always available for the electives and those that were available focused on programming in a specific programming language.

In the late 1980s the Ministry of Education recognized the need to redesign and rewrite the high school computer science curriculum. To accomplish this task, the Ministry appointed a professional committee to design a new computer science curriculum and funded teams consisting of teachers, computer science education researchers, and computer scientists to prepare the course material and assess the curriculum implementation. It was also recognized that many computer science teachers lacked sufficient preparation to ensure the successful implementation of the new curriculum having had no formal education in computer science or computer science pedagogy. The Ministry of Education therefore supported mandatory in-service courses for teachers.

The principles that guided the committee's work in the early 1990s were published in the 1995 paper [Gal-Ezer et al. 1995] and have continued relevance today. They are as follows.

— Computer science is a full-fledged scientific subject. It should be taught in high school on a par with other scientific subjects.
— The program should concentrate on the key concepts and foundations of the field. The main notion to be emphasized throughout the program is that of an algorithmic problem, and an algorithm as a solution thereof. To some extent, the more general notion of a system, and the accompanying principles of modularization and abstraction should also be discussed. Other topics are to be viewed as building upon these.
— Two different programs are needed, one for three units and one for five.[2] The first should be designed for students with only a general interest in computer science, and the second program (being deeper and broader) should be designed for students with a deeper interest in computer science. However, the design of the three-unit program should take into account that for many students this might be the only exposure to computer science, so that some attempt at comprehensive coverage should be made.
— Each of the two programs should have required units and electives. While the entire program should consist of central and important topics, some of these are

[2]As mentioned each of the scientific subjects taken in high school comes in two versions, a three-unit version and a five-unit version.

less crucial than others and can be made elective. Moreover, variance and flexibility is important for its own sake.
- Conceptual and experimental issues should be interwoven throughout the program. The word "conceptual" here does not mean 'impractical'. It refers to subjects that are taught in the classroom, rather than in the laboratory. This two-track approach, which we dubbed the "zipper principle", is one of the salient points of our program.
- Two quite different programming paradigms should ideally be taught. It is highly recommended that a student should learn a "mother tongue" first, but then, on a more humble scale, be introduced to another language, of radically different nature, that suggests alternative ways of algorithmic thinking. This emphasizes the fact that algorithmics is the central subject of study.
- A well-equipped and well-maintained computer laboratory is mandatory. This is the responsibility of the school system, and entails setting things up to support laboratory sessions and adequate individual "screen-time" for students.
- New course material must be written for all parts of the program. The teams that are to prepare the courseware must have "real" computer scientists on board, as well as computer science high-school teachers and researchers in computer science education.
- Teachers certified to teach the subject must have adequate formal computer science education. An undergraduate degree in computer science is a mandatory requirement, as is formal teacher training.

To address the decrease in the number of students taking matriculation exams with an emphasis on the sciences, the Ministry of Education launched a study program that placed more focus on the sciences including computer science, beginning in the seventh grade. In 2011, this new focus motivated the development of a new science and technology study program for grades 7 through 12 that resulted in a new computer science curriculum for grades 7 through 9. This curriculum is still in a pilot stage [Zur-Bargury 2012]. The middle school curriculum reflects the following principles.

- The program should provide the students with a set of basic concepts of computer science.
- The program should provide basic concepts in logical thinking and algorithmic thinking.
- The programming environment for the first year should be translatable to different natural languages easily.
- The programming environment should be free of charge (if possible), easy to use and graphically appealing.
- At least two different environments should be taught.
- The program needs to motivate the students to be active learners.
- Alternative, unique and innovative students' assessment techniques will be applied.

Over the last 20 years the computer science high school curriculum has undergone many revisions and in the last two years it was profoundly rewritten, but the guiding principles have remained the same.

Although it is beyond the scope of this article, Israeli high schools also offer a software engineering track. Students taking this track study an additional five units culminating in a large project. This track has also offers alternative courses such as such as graphics, cyber, mobile programming, and more. Only a few hundred students choose to take this track.

It is important to note that the curriculum, however, is only one aspect of ensuring rigorous and widespread computer science education. As Hazzan et al. [2008] noted, there are four essential elements for success:

(1) a well-established curriculum;
(2) a formal requirement for a mandatory computer science teacher certification;
(3) preparation study programs for pre-service and in-service teachers offered by universities and colleges; and
(4) an energetic researchers community.

These four elements are important for success. To keep up with developments in computer science and in pedagogy, the curricula must be regularly revised and rewritten. And like teachers in other scientific disciplines, computer science teachers must have a formal education in computer science and those who do not must receive professional development (best implemented by organizations or educational institutions that focus exclusively on supporting computer science classroom teachers).

Teacher preparation programs must also be in place to ensure that new teachers are adequately prepared. Shulman [1986] and Wilson et al. [1987] argued that teachers require several types of knowledge. For computer science teachers, these include content knowledge, knowledge of other disciplines in which computer science is used, knowledge of learners, knowledge of educational aims, and general pedagogical knowledge. In addition, teacher preparation and certification programs should concentrate on three critical domains: subject matter knowledge, pedagogical content knowledge, and curricular knowledge. As Gal-Ezer and Zur [2007] note, the preparation study programs offered by Israeli institutions follow these recommendations.

Researchers also play an important role in maintaining the effectiveness and relevance of computer science teaching and learning. A dedicated research community can provide critical feedback and recommendations on implementation and assessment, and can explore the difficulties inherent in teaching and learning the basic concepts of the discipline. The Israeli computer science education research community has fulfilled this role by conducting and publishing critical research at all stages of the curriculum development and implementation.

3.2. The United States

Recently, the education system in the US has come under increasing criticism for its failure to teach computer science, especially at the secondary level. A report released by ACM and CSTA called "Running on Empty: The Failure to Teach Computer Science in the Digital Age" [Wilson et al. 2010], for example, exposed the lack of coverage of core computer science concepts in state-level student learning standards across the country, leading the authors to conclude the following.

> Computer science education in K–12 is vital, but without specific intervention at all levels of government to make it stand on its own within the K–12 education landscape it will continue to fade from our schools. This will not only hurt the field of computing but also all the fields that depend on innovations that originate in computing. If we are to remain competitive in the global, high-tech marketplace of the 21st Century, we much revitalize computer science education in K–12 and make it part of the core curriculum for all students. [Wilson et al. 2010, p. 45]

As previously noted, in the absence of common national or state learning standards for computer science, the CSTA K-12 Computer Science Standards [Seehorn et al. 2011] have become the de facto national standards, attempting to define what students should know and be able to do at a given level of study. These standards grow from a foundational perspective that computer science is an intellectually important scientific discipline, that it teaches problem solving, and that a fundamental understanding of computer science enables students to be both educated consumers of technology and innovative creators capable of designing computing systems to improve the quality of life for all people.

The CSTA standards organize student learning using a three-level model that focuses on fundamental concepts. According to the authors, the general goals for the standards are as follows.

— The curriculum should prepare students to understand the nature of computer science and its place in the modern world.
— Students should understand that computer science interweaves concepts and skills.
— Students should be able to use computer science skills (especially computational thinking) in their problem-solving activities in other subjects.
— The computer science standards should complement Information Technology and Advanced Placement computer science curricula in schools where they are currently offered.

(The College Board's Advanced Placement program is similar to the International Baccalaureate and allows students to receive credit or placement in some postsecondary institutions for courses and exams successfully completed in high school.)

The standards include a set of five strands or themes of learning that run across all schooling levels and thorough out the individual secondary school courses. The strands are: computational thinking, collaboration, computing practice and programming, and community, global and ethical impacts. The learning outcomes at the elementary level are intended to be incorporated into cross-disciplinary learning activities. In middle school, they can be taught either in an interdisciplinary manner or in distinct courses. At the secondary level, CSTA organizes the standards into three distinct frameworks: Computer Science in the Modern World, Computer Science Concepts and Practices, and Topics in Computer Science.

4. K–12 CURRICULUM—CONTENT

4.1. Israel

As mentioned previously, in Israel, high school subjects can be studied at various levels. The detailed Israel computer science high school curriculum is published in two main articles [Gal-Ezer et al. 1995; Gal-Ezer and Harel 1999]. As these articles describe, most common are the three-unit and five-unit programs, which differ in the amount of material and conceptual depth. The three-unit program typically requires three weekly hours throughout the three high school years, while the five-unit program requires five weekly hours (one unit requires approximately one weekly hour and a total of 90 hours). The curriculum is constructed from the following modules.

— *Fundamentals of CS 1 and 2*. These two units provide the foundation for the entire program. The units introduce the concept of algorithms and how to apply them in a programming language (originally Pascal, now Java).
— *Second Paradigm*. One unit. This module introduces a different perspective for addressing algorithmic problems. It has a few alternatives including a different programming paradigm such as Logic programming, or a more application-focused perspective such as computer graphics or management information systems.
— *Data Structures*. (originally titled Software Design), One unit. This unit concentrates on data structures and complements the Fundamentals units.

Table I. Reported Content in Introductory High School Course

Topics	2005	2013
Programming	68%	81%
Problem Solving	NA	78%
Ethics and Social Issues	56%	55%
Hardware	60%	47%
Graphics	46%	45%
Web Development	43%	33%
Computer Security	14%	33%
Game Programming	NA	42%
Productivity Software	NA	23%
Databases	35%	23%
Networks	21%	19%
Logic	11%	17%
Other	27	10%

— *Theory*. One unit. This unit exposes students to the theoretical aspects of the discipline and provides students a choice of alternatives including a computational models unit.

The two first units are mandatory for both the three-unit and five-unit models. The students who opt for the three-unit option are typically students who are less interested in pursuing computer science as a career. Students who wish to study computer science in more depth with the possible goal of a career in this area typically take the five-unit option that includes a fourth unit on data structures and a fifth unit on theoretical aspects of computing. All of the units except the theoretical unit interweave conceptual and experimental issues (the zipper principle).

4.2. The United States

In the United States, there is no national curriculum and so schools may offer a wide variety of computer science courses or no computer science courses at all. In addition, the content of the courses offered can vary significantly, especially those offered at the introductory level.

Since 2005, CSTA has been tracking high school computer science education through a series of surveys it issues every two years [CSTA 2005, 2007, 2009, 2011, 2013]. Although these surveys have an internal bias given that schools offering computer science courses are more likely to self-report, they still provide an interesting picture of the content of introductory computer science courses over the years. As Table I shows, the survey results from 2005 and 2013 for introductory computer courses in high school indicate the introduction of new content areas and a shift towards a greater focus on programming and problem solving.

The situation is similar for more advanced courses, with the exception of the current Advanced Placement Computer Science A course. Advanced Placement courses are intended to cover the "information, skills, and assignments found in the corresponding college course" [College Board 2013a] and culminate in a final exam. The current course focuses on object-oriented program design, implementation, analysis, standard algorithms, and computing in context and is taught exclusively in Java. Work is currently under way on a new course entitled "Computer Science Principles" which is expected to be offered in schools as an Advanced Placement course in the 2016–17 school year. This new course will be language agnostic and will focus on the so-called seven Big Ideas of Computer science [College Board 2013b]. These are the following.

(1) *Creativity*. Computing is a creative activity. Students should understand that computer science interweaves concepts and skills.
(2) *Abstraction*. Abstraction reduces information and detail to focus on relevant concepts
(3) *Data*. Data and Information facilitate the creation of knowledge.
(4) *Algorithms*. Algorithms are used to develop and express solutions to computational problems.
(5) *Programming*. Programming enables problem soling, human expression, and creation of knowledge.
(6) Internet. The Internet pervades modern computing.
(7) Impact. Computing has global impact.

5. IMPLEMENTATION: MIDDLE AND HIGH SCHOOL

Setting standards and designing curricula are only the beginning of a long and challenging process of ensuring continual improvement in computer science education. In the following sections we try to highlight/describe a few milestones on this road.

5.1. Israel

5.1.1. Middle School. Israel is only now beginning to address computer science education in middle school. To date, the Ministry of Education has not developed a formal national computer science curriculum for this level (primarily for funding reasons) and as a result, computer science is taught in middle school on a voluntarily base (if a computer science teacher is available). In 2011, however, the Ministry of Education launched a new study program emphasizing science education. The program, called STEP–Science and Technology Excellence Program—begins in the seventh grade [Zur-Bargury 2012; Zur-Bargury et al. 2012, 2013]. In this science-motivated program, computer science, physics, biology, and chemistry play the same role, thus it was necessary to prepare a computer science middle-school curriculum. This middle-school program has been implemented in a stepwise fashion, beginning with 30 seventh grade classes and 832 students. Currently, there are 208 seventh grade classes, 181 eighth grade classes, and 26 ninth grade classes participating.

Despite a gradual implementation, the project encountered serious difficulties in assigning teachers. Experienced computer science high-school teachers were not always adept at teaching the younger seventh grade students. New teachers with less experience often concentrated on the technical level and could not cope with algorithmics and so teacher pre-service and in-service and workshops were critically important. The designers are continuing to assess and improve the curriculum as it will likely continue to evolve. (The current version is available, in Hebrew, on the Ministry of Education's website [Israeli Ministry of Education 2013]).

5.1.2. High School. The Israeli high school computer science curriculum was first implemented in 1995 and has been continually revised since that time. The early updates focused primarily on changing programming languages (from Pascal using a procedural paradigm to C, then C++, and finally to Java). While these updates were only technical, in the last two years, a more profound paradigm shift has occurred which integrates object oriented thinking from the beginning. In addition, the updated curriculum has been designed to more effectively interweave the theoretical and practical aspects of computer science from the very beginning of the two first units of study (*Fundamentals*).

Since its inception, the computer science curriculum in Israel has been the focus of a large body of research conducted by the members of the implementation committee, educational researchers, and the teacher community. The results of these efforts can be found in the work of many researchers including Gal-Ezer and Zeldes [2000], Armoni et al. [2005a, 2005b, 2006], and Meerbaum-Salant and Hazzan [2008].

The Israel National Center for Computer Science Teachers (established by the Ministry of Education as a professional community for computer science teachers) has also played a major role in the implementation process. This center was given the responsibility and funding to provide much-needed in-service workshops to assist the teachers to cope with these curriculum changes and this work has been carried out very successfully.

5.2. The United States

Because the teaching of computer science differs greatly from state to state and even from school to school, it is not possible to generalize with regard to teaching procedures, course contents, and assessments and to date there is no research available which examines these issues on a national basis. The CSTA standards document, however, sets forth a series of broad themes from which the reader may draw some inferences about underlying learning goals. These themes include: the intellectual importance of computer science, its potential for teaching problem-solving, and the ways in which computer science links to and supports other disciplines. The document also highlights the potential for computer science to engage all students and sets forth a number of equitable practices intended to connect students with the curriculum. These include the following.

— All students should have access to rigorous and culturally meaningful computer science and be held to high expectations for interacting with the curriculum.
— Diverse experiences, beliefs, and ways of knowing computer science should be acknowledged, incorporated, and celebrated in the classroom.
— The integration of different interpretations, strategies, and solutions that are computationally sound enhance classroom discussions and deepen understandings.
— The resources needed for teaching and learning computer science should be equitably allocated across groups of students, classrooms, and schools.
— Classroom learning communities should foster an environment in which all students are listened to, respected, and viewed as valuable contributors to the learning process.

Ongoing teacher reflection about belief systems, assumptions, and biases support the development of equitable teaching practices.

5.2.1. Middle School. A CSTA document entitled “Computer Science in K–8: Building a Strong Foundation” [CSTA 2012, p. i] indicated a “growing interest in computer science education at the elementary and middle school level” and noted that this interest seemed to be linked to “a growing awareness of the need for computational thinking skills for all students and the importance of embedding this learning throughout the school experience”. This new focus on computational thinking has been further supported by an educational programming renaissance that has led to the development of new computer science teaching tools (for example Alice, AgentSheet, Blockly, Kodu, and Scratch) and paradigms specifically aimed at engaging a broader diversity of students in hands-on learning. Despite these improvements, the teaching of computer science in middle school remains spotty and inconsistent. While a small number of middle schools provide computer science classes, more often students access computer

Table II. High School Programming Language Use

Programming Language	Percentage
Java	49%
Scratch	34%
Alice	30%
Visual Basic	30%
C++ or C#	18%
JavaScript	15%
Python	14%
AppInventor	13%
Greenfoot	6%
HTML	4%
Game Maker	3%
Jeroo	2%

science knowledge through extra-curricular activities such as after-school clubs and community programs.

5.2.2. High School. The CSTA survey results [CSTA 2005, 2007, 2009, 2011, 2013] indicate the place computer science has at the secondary school level. As of 2013, 74% of respondents indicated that their school offered an introductory level computer science course. This is a significant increase from 2011 and 2009 where the results were 69% and 65% respectively. The 2013 results indicated a similar increase in the number of schools offering an Advanced Placement computer science course, to 46% of respondents from 36% in 2011 and 27% in 2009.

In most schools, there is no computer science department. Rather, computer science is seen as a subprogram of mathematics, business, or career and technology education. The courses also vary widely from school to school, with most introductory courses including elements of programming, problem solving, ethical and social issues, graphics, web development, and applications. They also use a variety of programming languages. In 2013, for example, the respondents indicated use of numerous different languages for their introductory courses as shown in Table II.

While high-school computer science education continues to evolve there are still profound challenges in making rigorous computer science courses available to all students in all schools and ensuring that teachers are adequately prepared to teach these courses.

6. TEACHERS—THE CORNERSTONE

As the critical conveyors of knowledge to students teachers play an essential role in the implementation of any new curriculum. For this reason it is important to consider the preparation of teachers when comparing computer science education in different countries.

6.1. The United States

As might be expected, the absence of computer science from the list of "core academic courses" and the wildly varying state-level learning standards has also created a patchwork of policies and implementations with regard to teacher certification. As a new report entitled "Bugs in the System: Computer Science Teacher Education in the US" [Phillips and Stephenson 2013, p. 4] indicates, current computer science teacher certification in the US is:

> a deeply flawed system, typified by confusion about Computer Science as a discipline, a dearth of clear and relevant certification/licensure requirements, and a profound lack of agreement (or perhaps understanding) about what teachers should know and understand in order to be exemplary Computer Science teachers.

According to this report, education policy makers, administrators, and teachers are deeply confused about the computer science teacher certification requirements in their schools, districts, and states. Where any certification requirements exist, they often require computer science teachers to receive their primary teacher certification in some discipline other than computer science and then meet additional requirements to receive a supplemental computer science endorsement. In addition, both the original and the supplemental requirements may have only a tangential connection to the content of computer science learning or may have requirements (such as successful completion of a computer science methods course) that cannot be met within the state teacher preparation system.

And once teachers do qualify to teach computer science in schools, it can be very difficult for them to access the discipline-specific, classroom-relevant professional development they need to keep their content and pedagogical knowledge current. This however, is one area that has shown enormous improvement in the last three years. Thanks to generous funding from the National Science Foundation (most specifically the Broadening Participation in Computing and Computing Education for the 21st Century programs), from industry leaders such as Google and Oracle, and from CSTA, access to high quality professional development for K-12 computer science teachers has increased enormously. In 2013, the National Science Foundation's Broadening Participation in Computing and Computing Education for the 21st century programs are funding more than 20 professional development programs for computer science teachers, most involving face-to-face workshops, as well as year-long access to follow-on PD opportunities, online resources, and an online community of practice. In 2013, Google funded 58 U.S. workshops through its CS4HS program and CSTA partnered with Oracle to offer 30 workshops for teachers at CSTA regional chapters across the country.

6.2. Israel

The situation in Israel is somewhat different. From the very beginning, the committee that put together the curriculum was adamant that computer science teachers should have at least a Bachelor's degree in computer science and should complete a computer science teachers' preparation study program. Several teacher certification programs were initiated to help current and future teachers meet these requirements by upgrading their computer science knowledge and skills [Gal-Ezer and Harel 1998; Hazzan et al. 2010, 2011]. These programs included study programs, workshops, and new guidelines all intended to assist teachers in addressing the content and pedagogical challenges they faced. The Center for Teachers was also charged with providing professional development and support for ongoing improvements through pre-service and in-service workshops and annual conferences, and with building a strong community of practice.

Teachers in Israel have always been at the heart of activities such as material development and curriculum updates in response to changes in programming language and pedagogy. This inclusion stems from the belief that successful implementation of new materials and practice is only possible with the expertise and cooperation of teachers. At least two teachers have always been, and still are, members of the professional committee responsible for ongoing revisions to the computer science curriculum.

7. EQUITY—THE GRAND CHALLENGE

While the forms it takes in many countries may differ, the under-representation of some part of the population in computer science education and in industry is a problem that many countries share. As Prey and Weaver [2013] noted, this under-representation is deeply concerning because "a lack of diverse perspectives will inhibit innovation, productivity, and competitiveness" (p. 22). While it is beyond the scope of this article to detail the large body of research on the equity issues in computer science education and the workplace, it is worth noting that this challenge is wide spread and very difficult to overcome.

Much of the current research focuses on two distinct strategies or approaches. The first approach centers on making changes to the curriculum, classroom environment, and pedagogy to make computer science more appealing to the specific target audience (women or underrepresented student populations). The second approach focuses on improving all of these aspects of learning without a specific target audience in mind, in the belief that a rising tide will lift all boats [Frieze et al. 2006].

In the U.S. education system, the primary equity challenge stems from a critical underrepresentation of young women, African American, and Latino students. In 2008, researchers from the UCLA laid out a vision of pernicious inequities in their groundbreaking work "Stuck in the Shallow End: Education, Race, and Computing" [Margolis et al. 2008], clearly demonstrating that access to rigorous computer science education remains almost exclusive to wealthy white schools and that computer science knowledge has become a privileged commodity within the U.S. education system. The results of the annual Advanced Placement exams also provided a picture of student participation in rigorous academic computer science courses in the country. According the College Board report on the 2011–2012 school year [College Board 2012], only 19% of the writers of the Computer Science exam were female, 0.3% were American Indian/Alaska Native, 4.5% were African American/Black, and 8.4% were Hispanic/Latino/a. Over the last ten years, many institutions and organizations have worked both independently and collectively, at all levels, and in both formal (in-school) and informal (extra-curricular) education venues to address this problem. Success, as the Advanced Placement figures clearly demonstrate, has been illusive.

One possible solution to the US, equity challenges lies in some recent research conducted by CSTA and the National Research Center for College and University Admissions [CSTA 2013]. Data collected from student surveys indicated that both female and minority students were far more likely to take computer science courses when they could count one of these courses toward state graduation requirements. One might extrapolate from this that increasing the number of US states that allow computer science to count as a graduation requirement could significantly improve the likelihood that schools would make these courses available and that more students would take them. As of September 2013, only 13 states and the District of Columbia counted computer science as a math, science, or computer science graduation credit but efforts are underway to increase this number in many states [Phillips and Stephenson 2013] and it will be interesting to see whether this policy change will impact the number of students studying computer science in specific states and nationally.

In Israel, most of the work on equity issues has focused on the underrepresentation of young women in computer science, particularly their absence from high school computer science classes. Gal-Ezer, Vilner, and Zur [2008] described data published by the Israeli Bureau of Statistics IBS [Israeli Central Bureau of Statistics 2013] (see Table III) that clearly demonstrated that there is disproportionate participation of male and female students in high school computer science courses and that the average grades are even slightly better for women.

Table III. Students and Average Grades in the Computer Science Matriculation Exams

Year	Number of Students		Final Average Grade (0–100)	
	Men	Women	Men	Women
2003	5343 68%	2498 32%	89.08	89.54
2004	4608 68%	2188 32%	88.04	88.48
2005	3883 69%	1774 31%	88.57	88.82

At the request of one of the authors of this article the Israeli Bureau of Statistics provided new data relating to the years 1995–2010 and this new data confirms that the final average grades of the matriculation exams shown in Table III are consistent over the years.

The IBS data further demonstrates that studying computer science in high school impacts educational decisions taken later by the graduates and can perpetuate the underrepresentation of women in higher education and industry. It shows that students who do not take computer science courses in high school are less likely to study computer science at university, and this is especially true for women. (Details regarding the new data provided by the IBS will be presented in a future paper.)

8. SHARED CHALLENGES AND SIGNS OF PROGRESS

The place of computer science within the middle school and high school curriculum is in flux in many countries as policy makers and educators struggle to understand what knowledge and skills today's students require. While some countries are making aggressive moves to ensure that computer science is part of the "academic core" for all students, other countries struggle to move forward in their understanding of what the discipline is and how important it is compared to other academic subjects.

As indicated in the previous sections, Israel's highly centralized education system enables the development and implementation of national-level policies and resources that may provide for a more comprehensive and consistent approach to computer science education. This centralization however, does not protect computer science from shifting governmental education priorities and the need to constantly reassert that computer science should be treated on par with other scientific disciplines. And despite the Ministry of Education's commitment to keeping the curriculum up-to-date and relevant, this attention has not led to a decision to make the study computer science in high school mandatory for all students.

The decentralized education system in the US also has both benefits and drawbacks. Because states and schools are free to develop and offer courses and set graduation requirements based largely upon local needs, exemplary computer science programs have been allowed to flourish in some schools despite the fact that computer science is not considered part of the academic core. The change process for individual schools and teachers may also be more agile. The commitment to local autonomy, however, makes it difficult to achieve the kind of systemic national leap forward in computer science education happening in the UK [Burns 2013] and in other countries.

Despite these systemic differences, Israel and the US share a number of common or similar characteristics with regard to computer science education. As Table IV shows, the similarities are most clearly illustrated by the goals implicit in the previously referenced curriculum and standards documents.

Table IV. Common Goals and Themes in the U.S. and Israel

	United States	**Israel**
Computer science as a discipline and its place in schools	Computer Science considered part of STEM (Science, Technology, Engineering, Mathematics)	On the same par as Biology, Chemistry, Physics
Curriculum/standards content	Focus on Computational Thinking Concept-based student outcomes	Algorithmic thinking; key concept, lasting concepts
The importance of both theory and practice	Computing practice and Programming	The Zipper Principle Well maintained computer Lab
Teacher preparations and qualifications	Calls for more rigorous and consistent teacher certification/licensure	Formal computer science education requirements; Teacher preparation programs; Teacher certification requirements

Israel and the US also share a number of challenges that the authors believe are likely to be encountered by other countries as well. These include the following:

— the lack of understanding of computer science as a scientific discipline;
— a curriculum that must be constantly reviewed and revised;
— a curriculum that engages *all* students;
— ensuring that teachers have the technical knowledge, content knowledge, and pedagogical knowledge needed to teach computer science;
— providing ways for teachers to continually refresh and upgrade their knowledge;
— providing schools with access to the hardware, software, and resources they require in order to teach a rigorous, up-to-date computer science curriculum.

Although they may not have yet addressed all of these issues, there is evidence that several countries are moving in a positive direction. Indications of this progress include[3] the following:

— new national and local curricula focusing on fundamental computer science concepts,
— an increased emphasize on computational/algorithmic thinking,
— new pedagogies aimed at making computer science more attractive to all students including women and underrepresented minority students,
— ongoing updates to standards and curricula,
— increased discourse on computer science teacher preparation and certification,
— the growth of computer science teachers' communities and their efforts to develop standards and curricula, provide professional development, encourage innovation and continued learning, disseminate resources and best practices, and advocate for more and better computer science education.

Education is a complex environment in every country and the pressures on policy makers and administrators are extreme. If the indicators we have listed and additional positive steps to increase student access to rigorous computer science are to continue, then computer science researchers and educators must work together. This is our duty and we are the only people who can make things happen.

[3]New Zeeland Digital Technologies guidelines: dtg.tki.org.nz/; The Australian Curriculum: Australian Curriculum Technologies: http://www.australiancurriculum.edu.au/technologies/rational-aims/technologies; The CSTA K-12 Standards: csta.acm.org/curriculum/sub/k12standards.html.

REFERENCES

Armoni, M. and Gal-Ezer, J. 2005a. Teaching reductive thinking. *Math. Computer Educ. 39*, 2, 131–142.

Armoni, M., Gal-Ezer, J., and Hazzan, O. 2006. Reductive thinking in computer science. *Computer Sci. Educ. 16*, 4, 281–301.

Armoni, M., Gal-Ezer, J., and Tirosh, D. 2005b. Solving problems reductively. *J. Educ. Computing Research 32*, 2, 113–129.

Burns, J. 2013. School ICT to be replaced by computer science program. BBC News: Education and Family. http://www.bbc.co.uk/news/education-16493929.

College Board. 2012. Report to the Nation. http://media.collegeboard.com/digitalServices/pdf/ap/rtn/9th-annual/9th-annual-ap-report-subject-supplement-computer-science-a.pdf.

College Board. 2013a. Computer Science A Course Description. http://apcentral.collegeboard.com/apc/public/repository/ap-computer-science-course-description.pdf.

College Board. 2013b. Computer Science Principles. http://www.collegeboard.com/html/computerscience/.

CSTA. 2012.Computer science in K–8: Building a strong foundation. Computer Science Teachers Association, New York, NY.

CSTA. 2013. Computer science teachers association annual research report 2013. Computer Science Teachers Association and the National Research Center for College and University Admissions, New York, NY.

CSTA. Comparison Results for the 2005, 2007, 2009, 2011, and 2013 National secondary school computer science survey. http://csta.acm.org/Research/sub/HighSchoolSurveys.html.

Frieze, C., Hazzan, O., Blum, L., and Dias, M. B. 2006. Culture and environment as determinants of women's participation in computing: Revealing the women-CSfFit. In *Proceedings of the 37th Technical Symposium on Computer Science Education (SIGCSE'06)*. 22–26.

Gal-Ezer, J. and Harel, D. 1999. Curriculum and course syllabi fro high-school computer science program. *Computer Sci. Educ. 9*, 2, 114–147.

Gal-Ezer, J. and Zur, E. 2007. Reaching out to CS teachers: Certification via distance learning. *Math. Computer Educ. 41*, 3, 250–265.

Gal-Ezer, J., Beeri, C., Harel, D., and Yehudai, A. 1995. A high-school program in computer science. *Computer 28*, 10, 73–80.

Gal-Ezer, J., Vilner, T., and Zur, E. 2008. Once she makes it, she's there!: A case study. *Computer Sci. Educ. 18*, 1, 17–29.

Gal-Ezer, J. and Zeldes, A. 2000. Teaching software designing skills. *Computer Sci. Educ. 10*, 1, 25–38.

Gal-Ezer, J. and Harel, D. 1998. What (else) should CS educators know? *Commun. ACM 41*, 9, 77–84.

Hazzan, O., Gal-Ezer, J., and Blum, L. 2008. A model for high school computer science education: The four key elements that make it! In *Proceedings of the 39th Technical Symposium on Computer Science Education*. 281–285.

Hazzan, O., Gal-Ezer, J., and Ragonis, N. 2010. How to establish a computer science teacher preparation program at your university, The ECSTPP workshop. *ACM Inroads Mag. 21*, 1, 35–39.

Hazzan, O., Lapidot, T., and Ragonis, N. 2011. *Guide to Teaching Computer Science*. Springer.

Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M., Knobelsdorf, M., Magenheim, J., Mittermeir, R., and Schubert, S. 2011. Computer science/informatics in secondary education. In *Proceedings of the 16th Annual Conference Reports on Innovation and Technology in Computer Science Education - Working Group Reports*, L. Adams and J. J. Jurgens Eds., ACM, New York.

Informatics Europe and ACM Europe Working Group on Informatics Education. 2013. *Informatics Education: Europe Cannot Afford To Miss the Boat*. ACM, New York.

Israeli Central Bureau of Statistics Website: http://www.cbs.gov.il.

Israeli Ministry of Education: Computer Science and Information Technologies. 2013. http://www.csit.org.il/Default.aspx?MenuShow=TOCHNITLIM.

Margolis, J., Estrella, R., Goode, J., Jellison Holme, J., and Nao, K. 2008. *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press, Cambridge, MA.

Meerbaum–Salant, O. and Hazzan, O. (2008). Challenges in mentoring software development projects in the high school: Analysis according to Shulman's teacher knowledge base model. *J. Computers Math. Sci Teaching 28*, 1, 23–43.

Merrit, S., Bruen, C., East, P., Grantham, D., Rice, C., Proulx, V., Segal, G., and Wolf, C. 1994. *ACM Model High School Computer Science Curriculum*. ACM, New York.

Microsoft Research. *Towards 2020 Science*. http://research.microsoft.com/towards2020science/downloads/T2020S_ReportA4.pdf.

Phillips, P. and Stephenson, C. 2013. *Bugs in the System: Computer Science Teacher Certification in the U.S*. Computer Science Teachers Association, New York.

Prey, J. and Weaver, A., 2013. Fostering gender diversity in computing. *IEEE Computer*, 22–23.

Province of Ontario Ministry of Education. 1999. *The Ontario Curriculum Grades 9 and 10: Technological Education*. Ontario Ministry of Education. Toronto, ON.

Province of Ontario Ministry of Education. 2000. *The Ontario Curriculum Grades 11 and 12: Technological Education*. Ontario Ministry of Education. Toronto, ON.

Seehorn, D., Carey, S., Lee, I., Moix, D., O'Grady-Cuniff, D., Boucher Owens, B., Stephenson, C., and Verno, A. 2011. *CSTA K–12 Computer Science Standards*. Computer Science Teachers Association, New York.

Shulman, L. S. 1986. Those who understand: Knowledge growth in teaching. *Educ. Teacher 15*, 2, 4–14.

Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., and Verno, A. 2006. *A model curriculum for K–12 computer science*: Report of the ACM K–12 Task Force Computer Science Curriculum Committee, 2nd Ed. ACM, New York.

Wilson, C., Sudol, L., Stephenson, C., and Stehlik, M., 2010. *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. ACM, New York.

Wilson, S. M., Shulman, L. S., and Richert, A. E. 1987. 150 different ways of knowing: Representations of knowledge in teaching. In *Exploring Teachers' Thinking*, J. Calderhead, Ed., Cassel, London, UK, 104–124.

Wing, J. 2006. Computational thinking. *Commun. ACM*, 33–35.

Zur-Bargury, I. 2012. A new curriculum for junior-high in computer science. In *Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'12)*. ACM, New York.

Zur-Bargury, I., Parv, B., and Lanzberg, D. 2013. A nationwide exam as a tool for improving a new curriculum. In *Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'13)*.

Zur-Bargury. I., Haberman, B., Cohen, A., and Muller, O., Zohar, D., Levy, D. and Hotoveli, R. 2012. Implementing a new computer science curriculum for middle school in Israel. In *Proceedings of the IEEE Frontiers in Education Conference (FIE'12)*. 886–891.

Received March 2013; revised July 2013, December 2013, March 2014; accepted April 2014