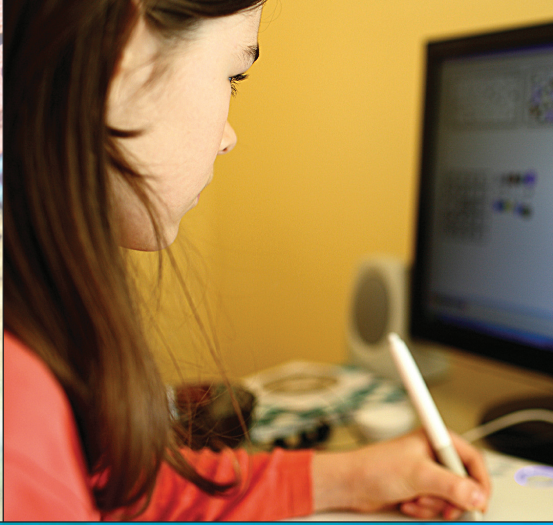


Running On Empty:

The Failure to Teach K–12 Computer Science in the Digital Age

Association for Computing Machinery

csta Computer Science Teachers Association

Running on Empty:

The Failure to Teach K–12 Computer Science in the Digital Age

The Association for Computing Machinery
The Computer Science Teachers Association

Cameron Wilson, ACM
Leigh Ann Sudol, Carnegie Mellon University
Chris Stephenson, The Computer Science Teachers Association, Member of ACM's Education Policy Committee
Mark Stehlik, Carnegie Mellon University, Member of ACM's Education Policy Committee

acm Association for Computing Machinery

csta Computer Science Teachers Association

http://www.acm.org/Runningonempty/

http://csta.acm.org/Runningonempty/

Guide to the Organization of this Report

This report presents results of a study conducted by the Association for Computing Machinery (ACM) and the Computer Science Teachers Association (CSTA) in 2009 and 2010 to determine the extent to which computer science education is incorporated into current state education standards, and to what extent states allow computer science courses to count as a graduation credit in a required or "core" subject. The intent of the authors of this report is to provide:

- a comprehensive report of the research result;
- a description of the current educational issues that underlie these results; and
- a set of recommendations for addressing the critical national and local issues that this data uncovers.

This report has five main sections

The *Executive Summary and Findings* provides a quick overview of the results of the study and their major educational implications. It is intended as an overview of both the results and the educational issues that surround the role of computer science education within the current K–12 curriculum. This section also contains the authors' recommendations for policy makers (National Call to Action, Recommendations).

The *Introduction* offers concise details of the research study and describes the U.S. K–12 educational policy framework. It includes a description of the role of learning standards and assessments and of state-level graduation requirements and their impact on student course selection.

The *K–12 Computer Science Education Background and Issues* section explores key issues impacting computer science education including the current confusion with regard to computing education and terminology. It provides a succinct definition for computer science specifically grounded in K–12 education and explores the critical place of computer science within STEM education and K–12 education in general. It also looks at the critical issue of teacher certification.

The *Findings* section provides a detailed examination of the research study, including its methodology, findings, and the limitations of the study.

The report also includes an extensive *Appendix*, which provides the results of the study on a state-by-state basis. Essentially, each appendix presents a state-level report card detailing the extent to which computer science standards are incorporated into current state standards, where these standards can be found, and whether computer science counts as a core subject area.

Executive Summary and Findings

- National Call To Action
- Recommendations
  - For All Levels Of Government
  - Federal Government
  - State And Local Governments

Executive Summary and Findings

Computer science and the technologies it enables now lie at the heart of our economy, our daily lives, and scientific enterprise. As the digital age has transformed the world and workforce, U.S. K–12 education has fallen woefully behind in preparing students with the fundamental computer science knowledge and skills they need for future success. To be a well-educated citizen as we move toward an ever-more computing-intensive world and to be prepared for the jobs of the 21st Century, students must have a deeper understanding of the fundamentals of computer science.

Paradoxically, as the role and significance of computing has increased in society and the economy, quality computer science education is being pushed out of the K–12 education system in the U.S. While there are many excellent K–12 computer science courses being taught across the country, in the past five years there has been a marked decline in the number of introductory and Advanced Placement computer science courses being taught in secondary schools. Most startlingly, this decline is occurring when national, state, and local policy makers are seeking to expand the capacity and quality of science, technology, engineering, and mathematics (STEM) education in the U.S. (See Table 1)

Numerous factors are contributing to this growing crisis, but this study finds that current federal, state, and local government policies underpinning the K–12 education system are deeply confused, conflicted, or inadequate to teach engaging computer science as an academic subject. Quality instruction always depends on knowledgeable and well-prepared teachers, on instructional materials that are engaging and carefully developed to enable student learning, and adequate resources and infrastructure to support teachers and student learning. These goals must be supported by a policy framework that sustains teacher development; certification and continuing education; appropriate curriculum development; and student access and interest. When it comes to computer science education, this framework is failing.

This study gathered data from all 50 states (and the District of Columbia) on the extent to which computer science standards (see Figure 1) are incorporated into existing state education standards and where these

TABLE 1

Secondary schools offering introductory (or pre-AP) Computer Science courses, change from 2005 baseline

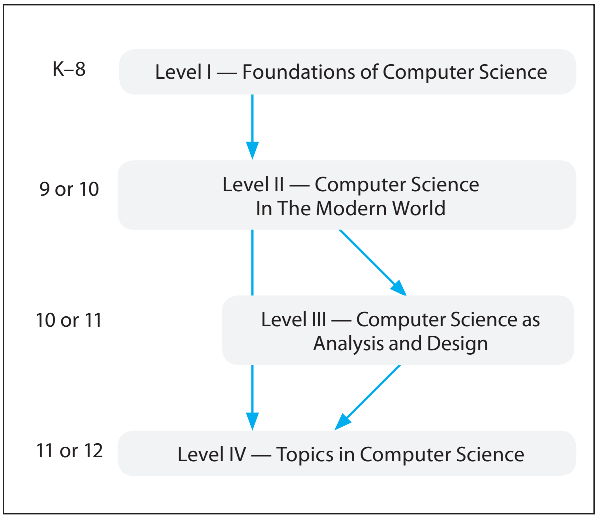
	2007	**2009**
Yes	-6%	-17%

Secondary offering AP Computer Science courses, change from 2005 baseline

	2007	**2009**
Yes	-20%	-35%

► *Source: Computer Science Teachers Association survery data of high schools*

FIGURE 1 Framework for ACM/CSTA Model Computer Science Standards


K–8
Level I — Foundations of Computer Science
9 or 10
Level II — Computer Science In The Modern World
10 or 11
Level III — Computer Science as Analysis and Design
11 or 12
Level IV — Topics in Computer Science

computer science outcomes might be found if they do exist in those standards. We also collected data on whether computer science courses taken at the secondary level count toward graduate credit requirements in a required discipline (such as math or science) or as simply an elective credit. This policy framework forms much of the backbone of the education landscape in the U.S. and provides a highly useful way of measuring the current state of computer science education at the secondary school level.

We find there are numerous and significant gaps between current state secondary education standards and nationally recognized computer science standards.[i] **Further, few states allow computer science courses to count toward a student's required credits for graduation.**

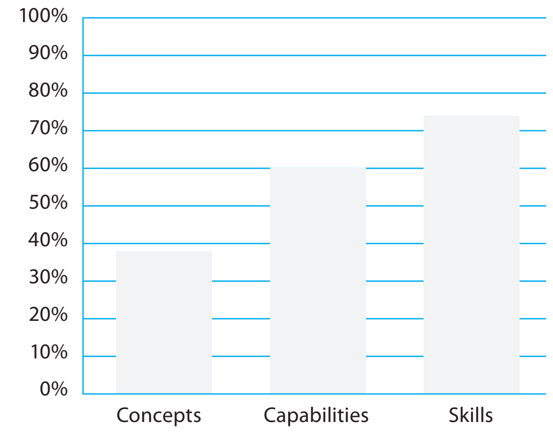
More specifically:

- Consistent with efforts to improve "technology literacy," states are focused almost exclusively on skill-based aspects of computing (such as, using a computer in other learning activities) and have few standards on the conceptual aspects of computer science that lay the foundation for innovation and deeper study in the field (for example, develop an understanding of an algorithm). See Figure 2.
- Major gaps exist in the adoption of computer science standards at the secondary (high school) level. **Only 14 states have adopted secondary state education standards for computer science instruction to a significant degree (defined as more than 50% of ACM and CSTA's national model computer science standards)**, leaving more than two-thirds of the entire country with few computer science standards at the secondary school level. Further, 14 states (and the District of Columbia) do not have even one upper-level standard for computer science instruction as part of their secondary education standards.
- Only 9 states allow computer science courses to count as a required graduation credit for either mathematics or science, as noted in Table 2. Further, no states require a computer science course as a condition of a student's graduation despite national broad-based education studies calling for all stu-

TABLE 2

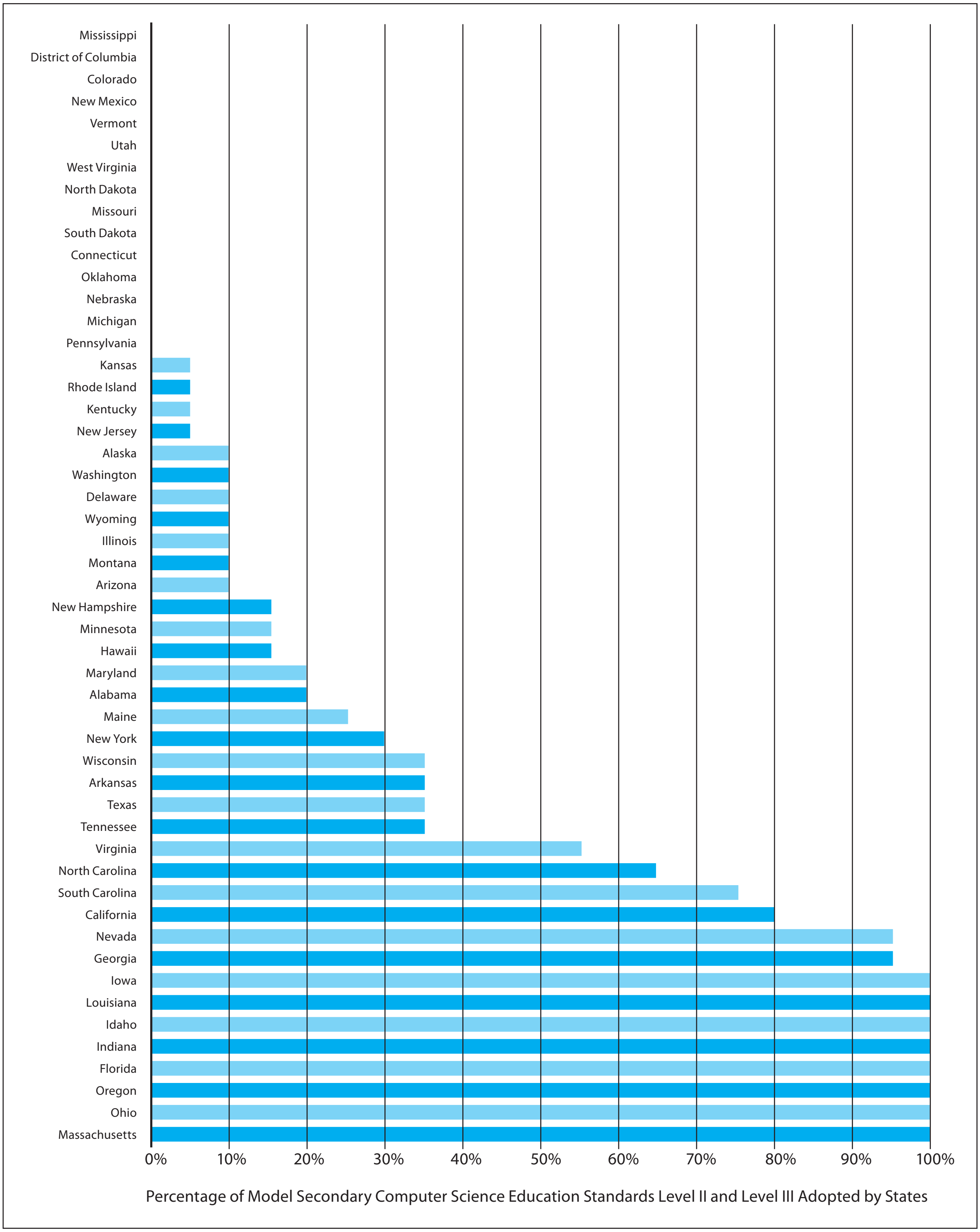
State	Computer Science Science as part of the "core"
Georgia	Science
Missouri	Math
New York	Math
North Carolina	Math
Oklahoma	Math
Oregon	Math
Rhode Island	Math
Texas	Math
Virginia	Math

FIGURE 2 National Snapshot: Adoption of Computer Science Standards*



*categories of computer science education standards

FIGURE 3 **Secondary School Standards Level II and Level III Adoption by State**



dents to be required to take some computer science in secondary education.[ii]

- There is deep and widespread confusion within the states as to what should constitute and how to differentiate *technology education, literacy and fluency; information technology education;* and *computer science as an academic subject*. We have elaborated on this critical issue in the section titled "K–12 Computer Science Education Background and Issues."

These findings stem from one key fact about computer science courses: generally, computer science is not considered by states and/or local school districts as part of the "core" curriculum that students must take in order to graduate from secondary school. School administrators, as well as federal and state program managers, need to make tough decisions about how to allocate scarce resources. Being part of the core curriculum often makes the difference between courses (and teachers) that are given resources and those that are not. How courses count in a student's academic career and whether subject-matter standards exist to measure progress often determine whether a subject is considered part of the core curriculum that a student must complete. And we have shown (in Table 2) that computer science courses typically do not count as either required mathematics or science courses and are not part of the core.

There is also a deeper and more troubling trend stemming from the lack of K–12 computer science education. In 2008, 17% of Advanced Placement (AP) computer science test takers were women, even though women represented 55% of all AP test takers.[iii] Further, only 784 African American students nationwide took the AP Computer Science exam.[iv] There is a significant lack of access to upper-level computer science courses for many under-represented populations, creating a major equity issue for access to this critical knowledge and this problem is growing. Lack of access to K–12 computer science education, or "privileged knowledge," is what education researchers[v] have described as a significant social justice issue for the 21st Century.

In addition, another study[vi] released in December 2008 by the Computer Science Teachers Association (CSTA) exposes another crucial problem in the overall education policy framework. State certification programs for computer science teachers are either non-existent or deeply flawed. The crisis in computer science teacher certification can be attributed to two key factors:

- a lack of clarity, understanding, and consistency with regard to current certification requirements, and
- where certification or endorsement requirements do exist, they often have no connection to computer science content.

It is clear there are major gaps in education policy needed to support quality computer science instruction for all K–12 students. Here, we offer detailed recommendations for all levels of government to address this growing crisis and to put computer science in the core of a student's education.

National Call to Action

No other subject will open as many doors in the 21st Century, regardless of a student's ultimate field of study or occupation, as computer science. At a time when computing is driving job growth and new scientific discovery, it is unacceptable that roughly two-thirds of the entire country has few computer science standards for secondary school education, K–8 computer science standards are deeply confused, few states count computer science as a core academic subject for graduation, and computer science teacher certification is deeply flawed. These are national failings and ones that we can ill afford in this digital age.

Parents must ask difficult questions about how computer science is being introduced to their children in K–12 education and demand that schools move beyond the current basic technology literacy curriculum. Policy makers at all levels need to review how computer science is treated within existing policy frameworks and schools, and ensure that engaging computer science courses based on fundamental principles of the discipline are part of the core curriculum. Now is the time to revitalize K–12 computer science education and ensure universal access to computer science courses by making it one of the core academic subjects students require to succeed in the 21st Century.

Recommendations for Federal, State, and Local Governments

For All Levels of Government

- Clearly define computer science education.

Federal Government

- Support state planning and implementation grants to improve computer science education;
- Build partnerships and national networks of support;
- Create pre-service and professional development opportunities for computer science teachers;
- Appoint blue ribbon commission to review the computer science teacher certification crisis; and
- Expand K–12 computer science education opportunities within existing federal programs.

State and Local Governments

- Create a well-defined set of K–12 computer science standards based on algorithmic/computational thinking concepts;
- Count computer science courses toward a student's core graduation requirements either as a computer science credit or as a mathematics or science credit;
- Develop courses to implement new computer science standards;
- Develop assessments for computer science education;
- Categorize computer science courses as academic courses;
- Expand professional development opportunities and recruit new computer science teachers;
- Expand access to computer science courses for under-represented populations; and
- Create flexible certification programs for computer science education grounded in the knowledge of the field.

Recommendations

In the U.S., most of the authority for determining what education is required of students and how their knowledge is measured is held at the state or at the school district level. The federal government also provides a layer of funding and policies that heavily influences education. Because the levers for education reform rest at multiple levels of government, policy makers at all levels must work toward creating a much more robust, clear, and coherent structure committed to ensuring that computer science is a core academic subject. Here are our specific recommendations on how federal, state, and local policy makers can close the significant gaps we have identified in this report.

For All Levels of Government

Clearly define computer science education.

As schools have increasingly stepped up the integration, use, and teaching of information technology, distinctions between these areas and computer science have blurred. Educators and policy makers consistently confuse the *use of technology and teaching of technology literacy* with *teaching computer science as a core academic discipline* within the STEM fields. In fact, this confusion is a fundamental reason behind many of the policy issues we have identified in this report. These related but distinct concepts must be clearly defined. Below we offer the following definition of computer science as a discipline and elements that can be considered appropriate for computer science curricula.

Computer science refers to the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society; and,

Computer science education includes the following elements: design (both software and hardware), creation of digital artifacts, abstraction, logic, algorithm development and implementation, programming paradigms and languages, theoretical foundations, networks, graphics, databases and information retrieval, information security and privacy, artificial intelligence, the relationship between computing and mathematics, the limits of computation, applications in information technology and information systems, and social impacts of computing.

Federal Government

Support state planning and implementation grants to improve computer science education.

We must incentivize states to develop specific plans to improve computer science education. As our research reveals:

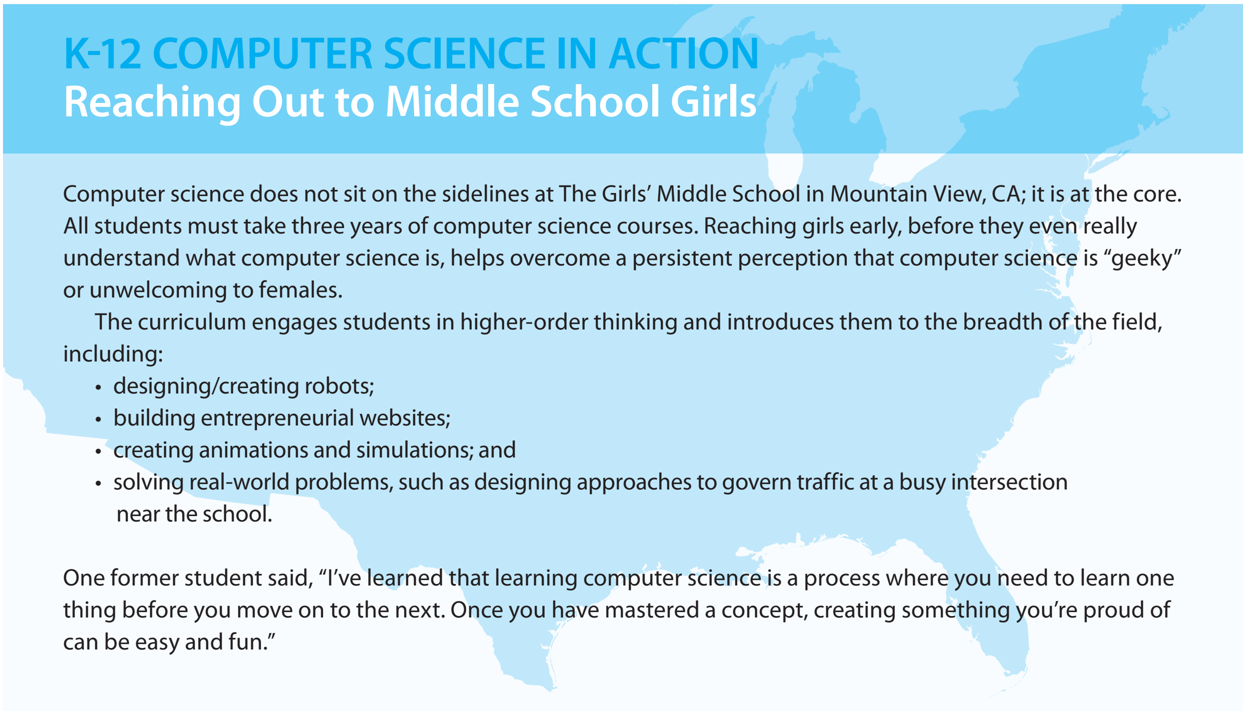
- Few states have defined K–12 learning standards for computer science education and, where they exist, they are focused predominantly on developing computing skills instead of teaching computer science concepts;
- Few states are ensuring computer science's place in the secondary school graduation requirement rubric;
- Computer science teacher certification programs do not exist in many states and when they exist are often flawed; and
- There are significant gender and equity issues regarding access to computer science education that must be addressed.

The federal government needs to catalyze action in the states to address the lack of computer science standards; assessments; graduation credits/requirements; and teacher preparation, professional development, and certification. **We recommend Congress create a grant program to fund states to assess the current state of computer science education, develop plans for its improvement, and implement state and local reforms.** Further, these grants should prioritize access to computer science education for under-represented groups.

Build partnerships and national networks of support.

There are a number of stakeholders outside of the education system that can be brought together to improve computer science education. A broader capacity initiative would build support for the goals and efforts of state planning and the implementation of grants. **We recommend Congress create a K–12 computer science education partnership program for local education agencies, institutes of higher education, non-profits, and businesses** that would:

- Develop curriculum;
- Implement outreach programs;
- Develop teacher support networks; and
- Evaluate state and local computer science education.



K-12 COMPUTER SCIENCE IN ACTION
Reaching Out to Middle School Girls

Computer science does not sit on the sidelines at The Girls' Middle School in Mountain View, CA; it is at the core. All students must take three years of computer science courses. Reaching girls early, before they even really understand what computer science is, helps overcome a persistent perception that computer science is "geeky" or unwelcoming to females.

The curriculum engages students in higher-order thinking and introduces them to the breadth of the field, including:

- designing/creating robots;
- building entrepreneurial websites;
- creating animations and simulations; and
- solving real-world problems, such as designing approaches to govern traffic at a busy intersection near the school.

One former student said, "I've learned that learning computer science is a process where you need to learn one thing before you move on to the next. Once you have mastered a concept, creating something you're proud of can be easy and fun."

Create pre-service and professional development opportunities for computer science teachers.

Despite the existence of National Council for the Accreditation of Teacher Education accreditation requirements for computer science, very few pre-service teacher preparation programs have the current capacity or coursework developed to prepare computer science teachers. As course offerings in computer science grow, particularly with plans to introduce a new Advanced Placement Computer Science Principles[vii] course into schools, we need a program to expand the number of certified K–12 computer science teachers with computer science expertise. **We recommend Congress create a program that would make competitive awards for schools of education working in collaboration with relevant programs in computer science to develop coursework leading the K–12 certification of computer science teachers.**

Appoint a blue ribbon commission to review the computer science teacher certification crisis.

Certification programs and requirements for computer science teachers must be reformed or built from the ground up. Often they do not exist and when they do they are typically not connected to actual computer science content knowledge. **We recommend the Secretary of Education establish a blue ribbon commission of state officials, discipline experts, practitioner organizations, and practitioners to review state computer science teacher certification requirements and share best practices.** A panel could draw on existing studies for models of computer science teacher certification and identify best practices for states to adopt.

Expand K–12 computer science education opportunities within existing federal programs.

The federal government has numerous programs focused on improving or expanding both formal and informal K–12 STEM education programs. We have consistently found that these programs focus resources on the "core" curriculum implemented in schools. These biases are usually subtle, ranging from requiring teacher certification (which is an issue for computer science teachers) to requirements to measure improvement from baseline assessments in "science" or "mathematics." **We recommend Congress and the Administration define K–12 STEM education programs to explicitly include computer science and expand opportunities for K–12 computer science education within the Department of Education and the National Science Foundation.**

In particular, we recommend Congress ensure computer science education is part of work carried out by NSF's Education and Human Resources Directorate, including its Math and Science Partnership program and at the Department of Education through its National Center for Educational Statistics. Further, we recommend Congress expand funding for K–12 computer science education within the Computer and Information Science and Engineering (CISE) Directorate at NSF as CISE is currently funding several critical projects in support of computer science education.

Many of the recommendations for the federal government are embodied in legislation recently introduced by Representative Jared Polis, the Computer Science Education Act (H.R. 5929), and we call on Congress to pass this legislation.

State and Local Governments

Create a well-defined set of K–12 computer science standards based on algorithmic/computational thinking concepts.

The confusion surrounding existing computing-related standards is impeding much-needed reform of computer science education. State standards have focused largely on technology literacy leaving computer science concepts mostly ignored. Where computer science standards do exist, they are usually part of various other academic subject areas. Since standards establish the framework for education carried out within the schools, **we recommend all states review their existing education standards and create a set of independent, grade-appropriate K–12 computer science standards informed by the ACM/CSTA Model Curriculum for K–12 Computer Science.** These standards should seek to integrate algorithmic/computational-thinking concepts across the curriculum at the K–8 level and allow for academically rigorous and independent computer science courses at the secondary school level.

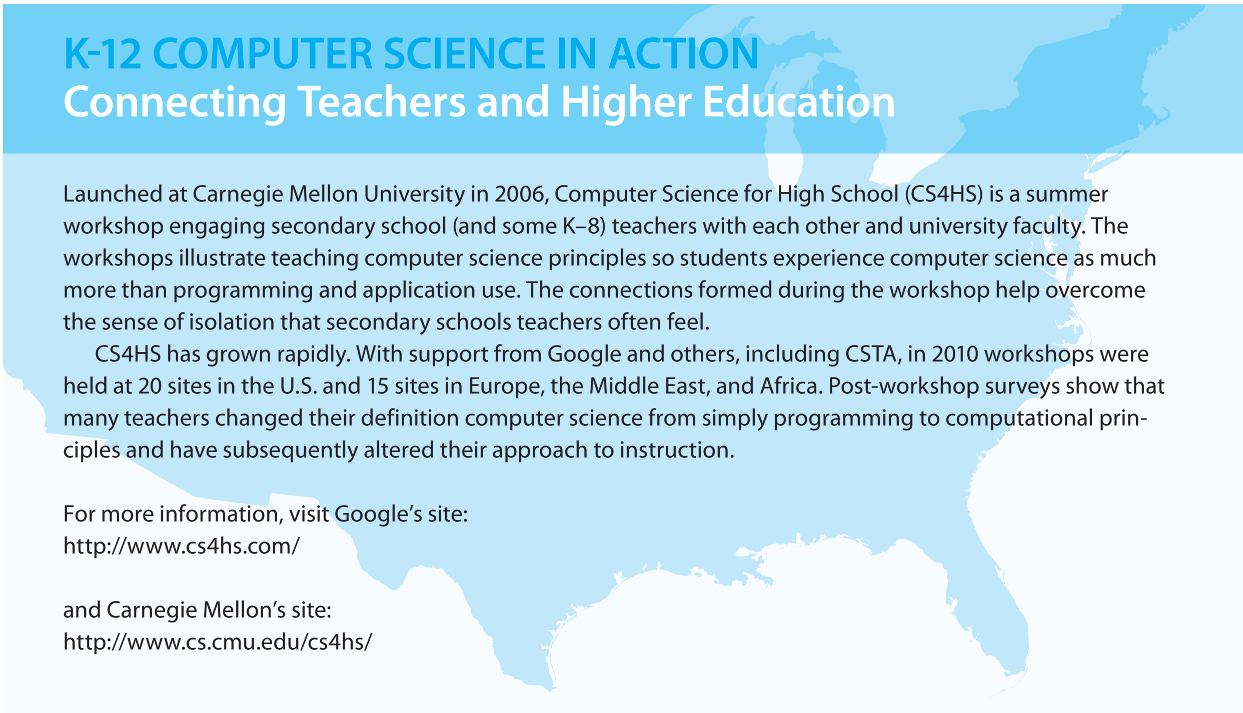
Count computer science courses toward a student's core graduation requirements either as a computer science credit or as a mathematics or science credit.

Current or new computer science courses often do not "count" as part of a student's required coursework. Quite often computer science courses are counted as nothing more than electives. Given the academic demands on college-bound students, it is unlikely that secondary school students can afford to take computer science as an elective. Further, because computer science is not part of the "core" curriculum, administrators are less inclined to introduce new courses or invest scarce resources into existing courses.

Being part of the core curriculum is made even more important as efforts are underway to expand the core. National organizations such as Achieve.org and the National Council of Teachers of Mathematics have pushed states to adopt requirements that students take four-course sequences in English, mathematics, science and social studies (sometimes called the "4x4" model) at the secondary level. Computer science should be listed as one of the courses that counts as either mathematics, science or computer science credit within these efforts. ACM and CSTA have made some progress in this area. Achieve's America Diploma Project (which is a model many states are following) lists Advanced Placement Computer Science as a capstone course in a four-course mathematics sequence. [viii]

In many states computer science resides within the Careers and Technology Education area and the technology credit has been the only way in which students can apply computer science to meet graduation credit requirement. Some states (most recently Texas) have removed the requirement for graduating students to complete a "technology" credit. The most common rationale for the elimination of this requirement has been that it was originally intended to ensure that students achieve a basic level of computing proficiency by graduation and that these requirements are now being met by students before they enter secondary school. The elimination of the technology credit has therefore, inadvertently, *reduced* the likelihood that college-bound students will choose to study computer science.

We recommend states and/or local districts (depending on state law) allow computer science to count as a



K-12 COMPUTER SCIENCE IN ACTION
Connecting Teachers and Higher Education

Launched at Carnegie Mellon University in 2006, Computer Science for High School (CS4HS) is a summer workshop engaging secondary school (and some K–8) teachers with each other and university faculty. The workshops illustrate teaching computer science principles so students experience computer science as much more than programming and application use. The connections formed during the workshop help overcome the sense of isolation that secondary schools teachers often feel.

CS4HS has grown rapidly. With support from Google and others, including CSTA, in 2010 workshops were held at 20 sites in the U.S. and 15 sites in Europe, the Middle East, and Africa. Post-workshop surveys show that many teachers changed their definition computer science from simply programming to computational principles and have subsequently altered their approach to instruction.

For more information, visit Google's site:
http://www.cs4hs.com/

and Carnegie Mellon's site:
http://www.cs.cmu.edu/cs4hs/

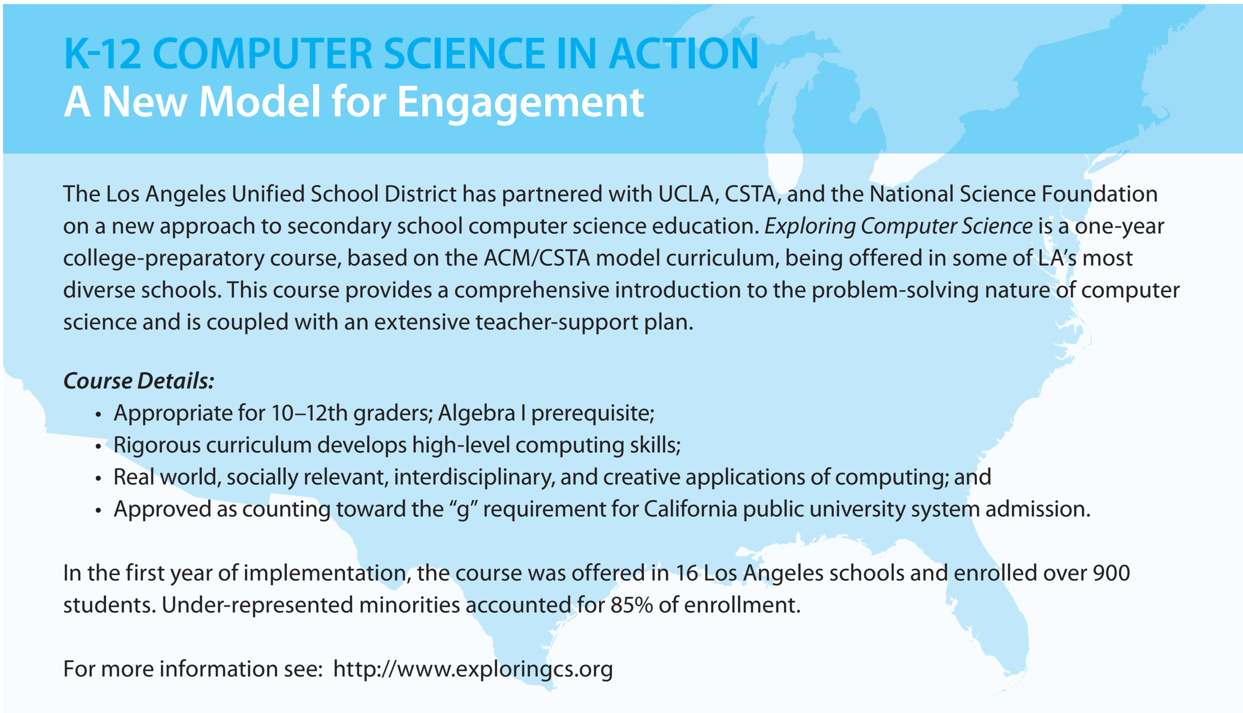
required mathematics or science credit, or specifically require a computer science credit to graduate. Further, we recommend that where a technology literacy course requirement exists, states and local districts allow a computer science course to count toward this requirement.

Develop courses to implement new computer science standards

Standards provide the overall framework for educational progress, but without courses and support for teachers, they are meaningless. The *ACM/CSTA Model Curriculum for K–12 Computer Science* provides a four-part framework (as noted in Figure 1) for integrating computer science standards throughout the curriculum at the K–8 level and, building upon that, a two or three course sequence at the secondary level. Given the crowded curriculum in today's schools, we realize that it is not always possible for schools to consistently provide the full roster of computer science courses. **We recommend, at a minimum, educational authorities fund the development of courses that integrate ACM/CSTA Level I standards into courses at the K–8 level and provide students with the Level II course and either the Level III course or the Computer Science AP course (Level IV) at the secondary level.** In this report, we have highlighted several courses that were developed using the *ACM/CSTA Model Curriculum for K–12 Computer Science* currently taught at some schools.

Develop assessments for computer science education

The recent provisions focused on measuring student performance in the Department of Education's Race to the Top Fund[ix] and the Administration's *Blueprint for Reform*[x] are continuing a now well-established trend that data should underpin curriculum decisions. In fact, a deeper look into many programs (such as the NSF's Math and Science Partnership) shows that, as a prerequisite for funding, there must be baseline assessments to measure student performance. Assessments for computer science education are virtually non-existent, putting these courses at a significant disadvantage for funding from programs that require data and for administrators who seek data. **We recommend states develop computer science assessments to measure student performance against the standards they have developed.**



K-12 COMPUTER SCIENCE IN ACTION
A New Model for Engagement

The Los Angeles Unified School District has partnered with UCLA, CSTA, and the National Science Foundation on a new approach to secondary school computer science education. *Exploring Computer Science* is a one-year college-preparatory course, based on the ACM/CSTA model curriculum, being offered in some of LA's most diverse schools. This course provides a comprehensive introduction to the problem-solving nature of computer science and is coupled with an extensive teacher-support plan.

Course Details:

- Appropriate for 10–12th graders; Algebra I prerequisite;
- Rigorous curriculum develops high-level computing skills;
- Real world, socially relevant, interdisciplinary, and creative applications of computing; and
- Approved as counting toward the "g" requirement for California public university system admission.

In the first year of implementation, the course was offered in 16 Los Angeles schools and enrolled over 900 students. Under-represented minorities accounted for 85% of enrollment.

For more information see: http://www.exploringcs.org

Categorize computer science courses as academic courses

A major contributing factor to the confusion about computer science education is that computer science or "computing" courses are organized into various departments within schools. For example, some are placed in the mathematics or science departments and some are within the vocational education departments. When computer science courses are placed within vocational education, they are rarely part of the "core" curriculum a student must take. Further, the curriculum for these courses tends to be focused on broader IT or technology skills rather than deeper computer science concepts. **We recommend school districts classify computer science courses as academic courses rather than as vocational or technical courses.**

Expand professional development opportunities and recruit new computer science teachers

As noted previously, there are very few pre-service teacher education programs preparing teacher candidates to teach computer science. In addition, anecdotal evidence strongly suggests that computer science teachers already working in the profession are afforded far fewer opportunities for relevant professional development and that these opportunities are lessening as state, district, and school budgets shrink. **We recommend as states adopt standards, develop new courses, and address teacher certification issues, they expand the opportunities for in-service and pre-service computer science teachers.**

Expand access to computer science courses for under-represented populations

Almost as startling as the disappearance of K–12 computer science courses nationally has been the lack of access to computer science courses for under-represented minorities[xi] (African Americans, Latinos, and Native Americans) and socio-economically challenged groups. Also, there exists a significant gender gap in students enrolling in advanced computer science courses.[xii] As computing becomes the foundation for innovation in the 21st Century, providing access to the skills and knowledge computer science offers will create opportunities for all groups. Denying access creates major equity issues. Further, diversity improves innovation. A study has found that computing teams comprised of both women and men had higher success rates than male-only teams.[xiii] **We recommend state and local governments dedicate resources to ensure all students, regardless of gender, race, or socio-economic status have access to at least one rigorous computer science course in secondary education.**

Create flexible certification programs for computer science education grounded in the knowledge of the field

Many states do not have certification programs for computer science teachers and, when they do, these programs often have no connection to actual computer science content and pedagogical knowledge.[xiv] K–12 teachers come to the computer science classroom from a variety of pathways and are typically from one of the following constituencies:

- new teachers: college or university students working towards their first teacher certification;
- veteran teachers with a certification in another area who have never taught computer science;
- veteran teachers with a certification in another area who have experience teaching computer science; and
- individuals coming from business with a computer science background and no teaching experience.

We recommend states adopt a multi-level model that provides the requisite knowledge (both technical and pedagogical) for computer science teachers while balancing the diversity of teacher backgrounds. Any preparation program for computer science teachers must include the following four major components:

- Academic requirements in the field of computer science;
- Academic requirements in the field of education;
- Methodology (a methods course) and field experience; and
- Assessment to document proficiency in general pedagogy, for example the Praxis II Principles of Learning and Teaching Test.

The CSTA report *Ensuring Exemplary Teaching in an Essential Discipline: Addressing the Crisis in Computer Science Teacher Certification* has detailed models to meet these goals.

Examples of States Moving Toward Reform

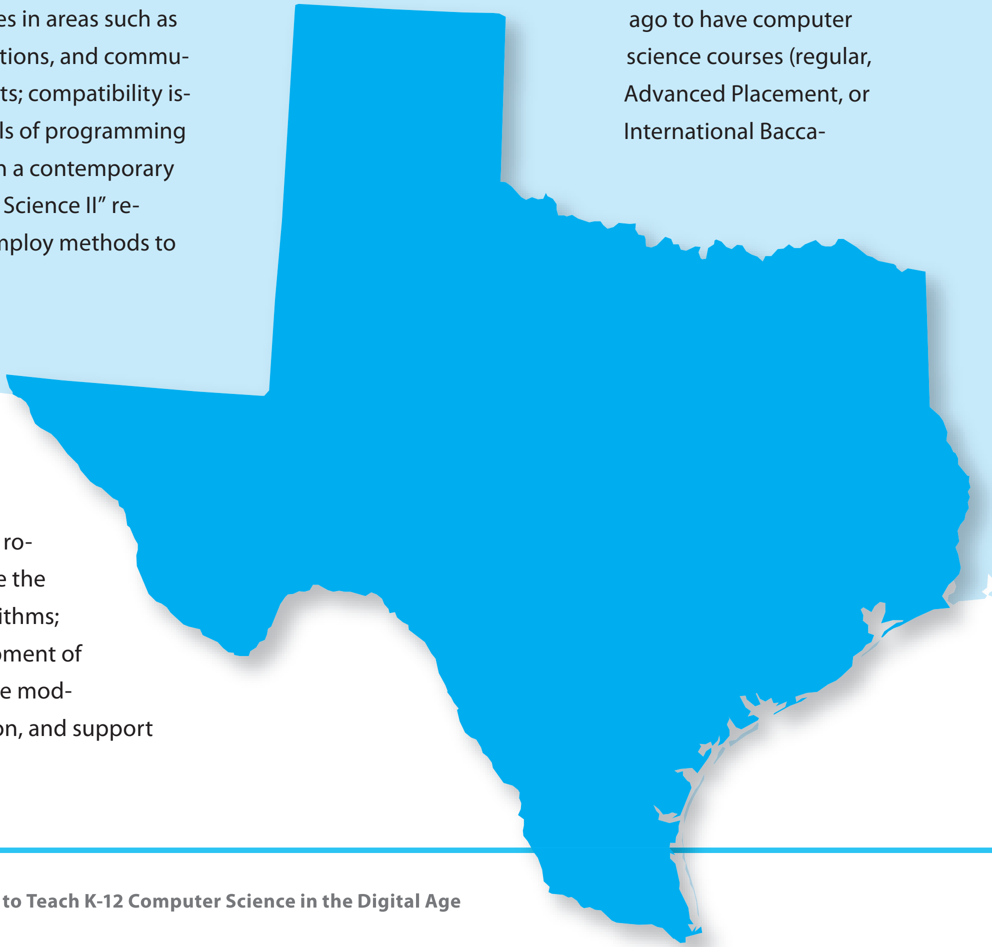
Ascertaining the extent to which computer science standards are truly incorporated into student learning in any given state is challenging on many levels. As mentioned previously, computer science standards, when they exists within state standards documents, are often not classified as computer science standards. Rather, they are classified under any possible number of discipline areas including math, science, technology, and business. This makes them difficult to find and even more difficult to assess. And where computer science standards do exist, there is no guarantee they are actually being addressed in the classroom; that is, while courses that cover these standards may be on the books, they are not necessarily being taught in every school. As the following two state exemplars demonstrate, even in states that have taken leadership and created a solid core of computer science standards and courses to address and access these standards, there are still systemic issues that can derail the best of curricular intentions.

TEXAS

Texas has a long history of computer science education in K–12 and has deeply embedded computer science standards at the high-school level. In Texas, the computer science standards are embedded within the Texas Essential Knowledge and Skills for Technology Applications standards. The required standards are addressed in a series of courses at the elementary, middle school, and high-school level. These include "Technology Applications, Kindergarten–Grade 2," "Technology Applications, Grades 3–5," "Technology Applications (Computer Literacy), Grades 6–8," "Computer Science I," and "Computer Science II."

Texas also provides standards for a number of additional computing technology courses. "Computer Science I" includes learning outcomes in areas such as operating systems, software applications, and communication and networking components; compatibility issues; differentiation among the levels of programming languages; and coding proficiency in a contemporary programming language. "Computer Science II" requires students to determine and employ methods to evaluate the design and functionality of the process using effective coding, design, and test data; use appropriately and trace recursion in program design comparing, iterative, and recursive algorithms; manipulate data structures using string processing; create robust programs; identify and describe the correctness and complexity of algorithms; and analyze models used in development of software including software life cycle models, design objectives, documentation, and support among other learning outcomes. Although the standards are not included within the state standards documents, Texas also offers the Advanced Placement Computer Science course in many schools. Texas also strives for a close alignment of its standards with both teacher certification requirements and professional development opportunities for teachers within the state.

Computer science education in Texas, however, continues to face a number of key challenges. Texas recently removed the Technology Applications credit from all three of its high school diploma plans, significantly reducing the incentive for students to take computer science courses in high school. Efforts several years ago to have computer science courses (regular, Advanced Placement, or International Bacca-

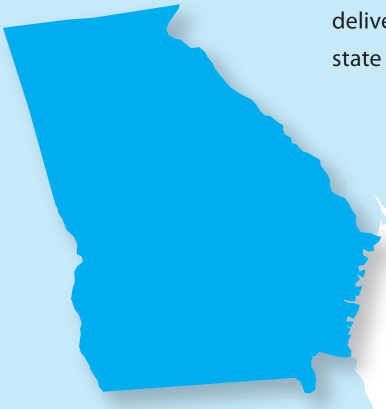


laureate) count as the fourth year science or math graduation credit resulted in approval as a fourth year math credit only for the recommended diploma program and not the higher 'recognized' diploma. In addition, Texas also continues to face a shortage of teachers who meet the current teaching certification requirements.

GEORGIA

Georgia serves as an excellent example of a state where significant gains have been made with regard to the incorporation of computer science standards into the state standards. Efforts to develop these standards began with a review of the framework and standards presented in the *ACM/CSTA Model Curriculum for K–12 Computer Science.* This framework was then adjusted to more accurately reflect local learning needs and priorities. There are four courses in the Georgia computing pathway: "Computing in the Modern World," "Beginning Programming," "Intermediate Programming," and "AP CS A."

"Computing in the Modern World" was designed to provide all secondary school students with the opportunity to gain an understanding of the diversity of the computing technology that surrounds them, as well as engage them in the exploration of topics such as Web design, problem solving and critical thinking, data (both understanding and storage), and an introduction to the power of programming. This course is superior to many of the applications courses currently offered in other states because it is intended to have students become active developers of computing technology rather than passive users and consumers.



The "Beginning Programming" course offers many of the topics found in a computer science course, including an introduction to hardware and software components; the design and use of algorithms to solve computational problems; the use conditional statements, iterative statements, and data structures such as variables, arrays, lists, stack, and queues. The "Intermediate Programming" course builds upon and extends knowledge and skills gained in the previous course, requiring students to build an interpreter and compiler, demonstrate knowledge of the key concepts of software engineering, and apply problem solving techniques to advanced problems. It requires students to use advanced programming techniques and constructs such as convditional statements, iterative statements, variables, arrays, lists, stacks, queues, and advanced mathematical expressions as well as to demonstrate knowledge of advanced object-oriented concepts such as polymorphism, interface, inheritance, encapsulation. This course also requires a deeper understanding of the limits of computation. Students in Georgia also have the opportunity to take a higher-level Advanced Placement Computer Science course. The standards for this course are described and stipulated in curriculum guidelines provided by the College Board.

In addition to developing a solid set of computer science standards and courses to address those standards, Georgia has also worked to closely couple its standards with its professional development programs for teachers, thus ensuring that teachers are better prepared to both deliver content and assess performance relating to these state standards.

Finally, Georgia is one of a handful of states that allows computer science courses to count as a "core" credit toward a student's graduation requirements. Students earn a science credit by completing a computer science course.

This is not to say, however, that Georgia has not faced challenges with regard to the implementation of these standards. The fact that the computer science content standards reside within business education standards has created challenges with regard to teacher certification and has required the state to take a flexible position where teacher eligibility is concerned. Also, fiscal problems within the state have led to the reassignment of teachers and the cancellation of rigorous computer science courses in many schools.

Introduction

- About This Study
- The Role Of Learning Standards And Assessments
- The Role Of Graduation Requirements

About this study

ACM and CSTA undertook this study to answer two basic questions related to the state of K–12 computer science in the U.S.:

1. To what degree have the *ACM/CSTA Model Curriculum for K–12 Computer Science* computer science standards been adopted by the states?
2. As national organizations such as Achieve.org and the National Council of Teachers of Mathematics have pushed states to adopt requirements that students take a four-course sequence in mathematics and a four-course sequence in science in secondary education, to what extent do states allow computer science to count as a graduation credit in a required or "core" subject?

The results are presented in the Findings section of this report and state-by-state report cards are attached in the Appendix.

The Role of Learning Standards and Assessments

Learning standards generally describe learning targets students should meet at specific grades, and guide teachers toward consistency, especially in the states with mandated state-level assessments (for example, the New York state regents exams). In most states, it is up to the school districts to establish curriculum that implements these standards. States and school districts often assess whether students learn the concepts through testing.

The growing awareness of severe inequalities in America's educational system has led to the enactment of legislation to hold schools, and by extension teachers, more accountable for meeting the standards set forth by their states. As is normally the case, the inequities existed mostly in underprivileged schools and neighborhoods. With this emphasis and pressure, standards have become ingrained as a part of a teacher's everyday practice. Teachers are expected to label lesson plans with the standards that they address and are often asked to justify classroom activities by citing appropriate standards.

Growing concern about differing and disparate standards among the states in some subject areas—particularly in "core" subject areas such as mathematics, science, and English—and how these standards compare to international benchmarks led two leading education organizations—the Council of Chief State School Officers and the National Governors Association—to propose the Common Core State Standards Initiative[xv] (CCSSI). Further, the National Research Council recently released a draft framework for science, engineering, and technology concepts.[xvi]

Similar to the ACM/CSTA standards for computer science, these efforts seek to provide a national framework of standards in mathematics and English Language Arts that states and local districts can adopt. Notably, between these two efforts, computer science is not mentioned as a specific discipline in any of the proposed standards. However a draft Appendix to the CCSSI (the final had not yet been released as of this writing) includes a model for a four-course sequence for mathematics in which computer science is listed as an appropriate fourth-year course.

Standards not only establish the state's expectations of teachers and students, they also, whether intentionally or not, establish an intrinsic hierarchy that impacts how administrators choose to allocate their resources. The existence of standards in some disciplines and subject areas and not in others privileges some at the expense of others. And this differentiation may be just as, or even more, critical to long-term academic and career success.

The Role of Graduation Requirements

Each of the 50 states and the District of Columbia has a set of minimum requirements that students must meet in order to graduate from secondary school. A set of requirements is a list of subject areas and the number of credits (usually in full- or half-year increments) that a student must take in each of those subject areas. For example, a state may have a requirement that the "core" education include:

- four years of English;
- four years of mathematics;
- three years of science;
- three years of social studies;
- one year of physical education;
- one year of Fine Arts;
- 1/2 year of technology literacy; and
- six elective credits.

TABLE 3 Graduation Requirements for the Top 5 Populated States – In Credits (c) or Years (y)

State	California (y)	Illinois (y)	Texas (y) (minimum)	Texas (y) (recommended)	New York (c) (Regents)	New York (c) (Advanced Regents)	Florida (c)
English	3	4	4	4	8	8	4
Math	2	3	3	4	6	6	4
Science	2	2	2	4	6	6	3
Soc. Studies	6	2	2.5	3.5	8	8	4
Foreign Language	1*	0	0	2	2	6	0
Fine Arts	1*	0	1	1	2	2	1
Health/PE	2	0.5	1	1	5	5	1
Electives	0	1	8.5	5.5	7	7	8
Other	other coursework as district determines	0	economics .5 speech .5	economics .5 speech .5	0	0	0

► *Source : Review of Requirements Found on State Education Authority Websites.*
**Students may take one year of Foreign Language or Fine Arts.*

Most states set the course requirements for graduation; however, a number of states defer to individual school districts (for example, Colorado and Massachusetts). Additionally, even states that centralize their requirements and course mappings give local school boards freedom to set additional local standards. In these cases, the state standards are seen as a minimum requirement which local districts can augment with additional standards to better meet local needs. (See Table 3 for high school graduation requirements for the five most populated U.S. states.)

There has been a flurry of activity in the last few years to overhaul state graduation requirements as a consequence of the push to improve education in the face of global competition. Arkansas, Colorado, Connecticut, Delaware, Florida, Massachusetts, Michigan, Texas, and Virginia have either proposed or adopted new graduation requirements that typically involve moving to a fourth required year in English, mathematics, science, and social studies (often called the 4x4 requirement). However, despite the push for additional mathematics and science, and the recognition that curricula and graduation requirements must reflect the demands of a 21st Century workplace, computer science is still predominantly relegated to an elective credit.

As a result, not only are students less likely to perceive computer science as relevant to their academic and career success, even students who are eager to take computer science courses are unable to fit them into their increasingly overcrowded school schedules. Administrators are also less likely to allocate funding to provide staff and resources for elective courses.

K–12 Computer Science Education Background And Issues

- Understanding Computer Science Education, Information Technology, And Technology Literacy
- What Is Computer Science Education?
- Why Teach Computer Science Education In K–12?
- Where Is Computer Science In “STEM”?
- Computer Science Teacher Certification

Understanding Computer Science Education, Information Technology, and Technology Literacy

The confusion over the swirl of terms and curriculum related to computing, technology, and computer science is one of the fundamental reasons why computer science standards, course classification and credits, teacher certification programs, and curriculum are inadequate or non-existent.

Despite the incredible diversity of the U.S. workforce, it is clear that most of today's jobs depend on some knowledge of, and skills to use, computing technologies. It is also clear that this trend is growing as computing becomes embedded more deeply in everyday commerce and society. What is unclear to many educators is what curriculum will support this growing trend.

Various studies and groups advocate for differing curriculum and standards that cloud these waters. Some have broad definitions of "technology literacy" that encompass many fields (National Assessment Governing Board's Technology and Engineering Literacy Framework for the 2014 National Assessment of Educational Progress), some link technology and engineering education (National Academies Engineering in K–12 Education: Understanding the Status and Improving the Prospects), some focus on computing or information technology certification (Cisco Networking Academy), and some focus on ensuring technology is used throughout the curriculum (International Society for Technology Education's NETS' standards). This study argues for specific computer science standards and courses that are academically rigorous, widely engaging, founded upon the body of knowledge for the discipline, and that are part of the core curriculum for all students.

This swirl of terms is daunting to anyone trying to make decisions about what students need to learn in the field of computer science. We hope the following definitions will provide some assistance for educators, policy makers, and parents who are trying to cope with this confusion and complexity.

Computer science: An academic discipline that encompasses the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society. In the sections that follow we detail the ACM/CSTA standards that should underpin the teaching of computer science. These standards serve as the baseline for measurement against existing state standards in this study.

Technology literacy and fluency: A spectrum of curricula ranging from literacy (understanding how to use technology), to fluency (the ability to express ideas creatively, reformulate knowledge, and synthesize new information and technology).

Information technology: A broad and diverse set of topics, but typically focused on applying the components of information technology to solve a business information problem, such as network or database administration.

Educational technology *or computing across the curriculum:* The integration of technology into teaching in order to advance student learning across academic disciplines.

Computing education: Another broadly used term that, depending on the educational context, may encompass only one of the noted areas above to all of them and more.

The reality is all of these efforts are worthwhile and have a place in K–12 education, but we must be clear about how they are related and how they are distinct to ensure adequate workforce and college preparation.

Goals to embed technology use in education or to ensure technology literacy are often focused on achieving a basic knowledge of IT or a specialized knowledge (in the case of IT courses) of a narrow tool. These are clearly important skills considering computing's everyday uses in today's economy.

Our research has shown that most states are focused on lower-level skills instead of deeper computer science concepts and capabilities. However, this is not enough in the 21st Century. Further, few states now allow computer science courses, if they exist, to count as a core mathematics or science requirement. But states currently require students to take some form of technology literacy or computing application courses. As our economy becomes more dependent on computing technologies,

TABLE 4 Employment by Occupation, 2008 and Projected 2018

BLS Job Categories	2008 Total employment	2018 Total employment	Percent change	Numeric change	*Total job openings due to growth and net replacements 2008-18
Total, all occupations	150,931,700	166,205,600	10%	15,600,125	50,928,500
Total, all computing occupations	3,792,000	4,606,900	21%	814,900	1,504,100
Computer and information scientists, research	28,900	35,900	24%	7,000	13,200
Computer programmers	426,700	414,400	-3%	-12,300	80,300
Computer software engineers, applications	514,800	689,900	34%	175,100	218,400
Computer software engineers, systems software	394,800	515,000	30%	120,200	153,400
Computer support specialists	565,700	643,700	14%	78,000	234,600
Computer systems analysts	532,200	640,300	20%	108,100	222,800
Database administrators	120,400	144,700	20%	24,300	44,400
Network and computer systems administrators	339,500	418,400	23%	78,900	135,500
Network systems and data communications analysts	292,000	447,800	53%	155,800	208,300
Computer and information systems managers	293,000	342,500	17%	49,500	97,100
All other computer specialists	209,300	236,800	13%	27,500	72,600
Computer hardware engineers	74,700	77,500	4%	2,800	23,500

► *Source: Compiled from the Bureau of Labor Statistics Occupational Employment Statistics Database.*
**Total job openings represent the sum of employment increases and net replacements. If employment change is negative, job openings due to growth are zero and total job openings equal net replacements.*

it is clear that simply having a rudimentary knowledge of computing is not enough.

Where computer science education differs from basic technology literacy/IT goals is that it teaches fundamental concepts of computing, just as an academic course in physics will teach a student the fundamental laws of motion and energy. Computer science teaching should sit on a continuum from basic computing concepts that can be attained at elementary and middle school levels to deeper knowledge, skills, and practices more appropriate for secondary school. Some of its topics overlap with technology literacy and IT curricula, while some are completely different. For example, the complexity of algorithms is a fundamental idea in computer science but probably would not appear in technology literacy or IT curriculum. Gaining a deeper knowledge of computer science and its fundamental aspects is essential not only to have a clear understanding of "what is going on under the hood" of computer software or hardware, but also to develop critical thinking skills that will serve a student throughout his or her career.

The future workforce picture provides a compelling case as to why schools need to move toward improving computer science education. The U.S. Bureau of Labor Statistics projects that the computing sector will have 1.5 million job openings over the next 10 years, making this one of the fastest growing economic fields (see Table 4). While there are many pathways into these jobs, a deeper look at the fastest growing occupations within this field (such as computer software engineers) shows they either will require a computer science or related degree or greatly benefit from the knowledge and skills imparted by computer science courses.

But improving the outlook for students going into computing careers is only one reason for addressing K–12 computer science education issues. The knowledge and skills imparted by computer science also enables innovation and opens doors. Many fields of science and business depend on computer science. Modeling brings together many of the fundamental concepts of computer science and is used on a daily basis in many fields. New subfields of science, such as computational physics or computational biology, have been created by bringing together computer science with an existing scientific discipline. If K–12 schools are seeking to make students college- and career-ready, computer science must be part of the core curriculum.

What is Computer Science Education?

To clear up the confusion of terms surrounding computer science education, we need to first define what education standards underpin its teaching.

In 2003, ACM created a task force of recognized experts to address the critical need to provide a definition of computer science that would be relevant for K–12 educators and to develop a set of learning standards that would meet the educational needs of students at different grade levels. The task force developed and published *A Model Curriculum for K–12 Computer Science: Final Report of the ACM K–12 Task Force Curriculum Committee,*[xvii] which provided a set of model learning standards for computer science education. This document (revised in 2006) clearly defined, for use by any state or school district, the framework of grade-appropriate standards underpinning K–12 computer science education.

This framework focuses on fundamental concepts and has the following general goals:

1. The curriculum should prepare students to understand the nature of computer science and its place in the modern world.
2. Students should understand that computer science interleaves principles and skills.
3. Students should be able to use computer science skills (especially algorithmic/computational thinking) in their problem-solving activities in other subjects (for example, the use of logic for understanding the semantics of English in a language arts class).
4. The computer science curriculum should complement IT and AP computer science curricula in any schools where they are currently offered.

Level I (recommended for grades K–8) provides elementary school students with foundational concepts in computer science by integrating basic skills in technology with simple ideas about algorithmic thinking. The Level I standards largely paraphrase the International Society for Technology in Education's National Education Technology Standards (2002) but have been augmented to include computational/algorithmic thinking concepts at lower grades to help provide the foundation for further study of computer science in secondary education. These standards can be addressed either in discrete computing courses or by adding short modules to existing science, mathematics, and social studies courses.

Level II (recommended for grade 9 or 10) provides students with a coherent and broad understanding of the principles, methodologies, and applications of computer science in the modern world. This program of study can best be offered as a one-year course accessible to all students, whether they are college-bound or workplace-bound.

Level III (recommended for grade 10 or 11) builds on the knowledge acquired at Level II but it places particular emphasis on the scientific and engineering aspects of computer science—mathematical principles, algorithmic problem-solving and programming, software and hardware design, networks, and the social impacts of computing. This course allows students to explore their interest in and aptitude for computer science as a profession.

Finally, the ***Level IV*** (recommended for grade 11 or 12) offering is an elective that provides depth of study in one particular area of computer science. This may take a number of forms including, for example, an AP computer science course, a projects-based course in multimedia design, or a vendor-supplied course that leads to professional certification. (An analysis of gaps in Level IV standards was not included in this study because there are no standards listed, instead potential course topics are offered.)

Below are the specific learning standards broken out by various grade levels.

Level I Standards (K-8)

Grades K–2:

- Use standard input and output devices to successfully operate computers and related technologies.
- Use a computer for both directed and independent learning activities.
- Communicate about technology using developmentally appropriate and accurate terminology.
- Use developmentally appropriate multimedia resources (for example, interactive books, educational software, elementary multimedia encyclopedias) to support learning.
- Work cooperatively and collaboratively with peers, teachers, and others when using technology.
- Demonstrate positive social and ethical behaviors when using technology.
- Practice responsible use of technology systems and software.
- Create developmentally appropriate multimedia products with support from teachers, family members, or student partners.
- Use technology resources (for example, puzzles, logical thinking programs, writing tools, digital cameras, drawing tools) for problem solving, communication, and illustration of thoughts, ideas, and stories.
- Gather information and communicate with others using telecommunications, with support from teachers, family members, or student partners.
- Understand how 0s and 1s can be used to represent information, such as digital images and numbers.
- Understand how to arrange (sort) information into useful order, such as a telephone directory, without using a computer.

Grades 3–5:

- Be comfortable using keyboards and other input and output devices, and reach an appropriate level of proficiency using the keyboard with correct fingering.
- Discuss common uses of technology in daily life and the advantages and disadvantages those uses provide.
- Discuss basic issues related to responsible use of technology and information, and describe personal consequences of inappropriate use.
- Use general-purpose productivity tools and peripherals to support personal productivity, remediate skill deficits, and facilitate learning throughout the curriculum.
- Use technology tools (for example, multimedia authoring, presentation, Web tools, digital cameras, scanners) for individual and collaborative writing, communication, and publishing activities to create presentations for audiences inside and outside the classroom.
- Use telecommunications efficiently to access remote information, communicate with others in support of direct and independent learning, and pursue personal interests.
- Use online resources (for example, email, online discussions, Web environments) to participate in collaborative problem-solving activities for the purpose of developing solutions or products for audiences inside and outside the classroom.
- Use technology resources (for example, calculators, data collection probes, videos, educational soft-

ware) for problem-solving, self-directed learning, and extended learning activities.

- Determine which technology is useful and select the appropriate tool(s) and technology resources to address a variety of tasks and problems.
- Evaluate the accuracy, relevance, appropriateness, comprehensiveness, and bias that occur in electronic information sources.
- Develop a simple understanding of an algorithm, such as text compression, search, or network routing, using computer-free exercises.

Grades 6–8:

- Apply strategies for identifying and solving routine hardware and software problems that occur during everyday use.
- Demonstrate knowledge of current changes in information technologies and the effects those changes have on the workplace and society.
- Exhibit legal and ethical behaviors when using information and technology and discuss consequences of misuse.
- Use content-specific tools, software, and simulations (for example, environmental probes, graphing calculators, exploratory environments, Web tools) to support learning and research.
- Apply productivity/multimedia tools and peripherals to support personal productivity, group collaboration, and learning throughout the curriculum.
- Design, develop, publish, and present products (for example, Web pages, videotapes) using technology resources that demonstrate and communicate curriculum concepts to audiences inside and outside the classroom.
- Collaborate with peers, experts, and others using telecommunications tools to investigate educational problems, issues, and information, and to develop solutions for audiences inside and outside the classroom.
- Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems.
- Demonstrate an understanding of concepts underlying hardware, software, algorithms, and their practical applications.
- Discover and evaluate the accuracy, relevance, appropriateness, comprehensiveness, and bias of electronic information sources concerning real-world problems.
- Understand the graph as a tool for representing problem states and solutions to complex problems.
- Understand the fundamental ideas of logic and its usefulness for solving real-world problems.

Level II Standards (9 or 10)

Students should gain a conceptual understanding of the following topics:

- Principles of computer organization and the major components (input, output, memory, storage, processing, software, operating system, among others).
- The basic steps in algorithmic problem-solving (problem statement and exploration, examination of sample instances, design, program coding, testing and verification).
- The basic components of computer networks (servers, file protection, routing protocols for connection/communication, spoolers and queues, shared resources, and fault-tolerance).
- Organization of Internet elements, Web page design (forms, text, graphics, client- and server-side scripts), and hypermedia (links, navigation, search engines and strategies, interpretation, and evaluation).
- The notion of hierarchy and abstraction in computing, including high-level languages, translation (compilers, interpreters, linking), machine languages, instruction sets, and logic circuits.
- The connection between elements of mathematics and computer science, including binary numbers, logic, sets, and functions.
- The notion of computers as models of intelligent behavior (as found in robot motion, speech and language understanding, and computer vision), and what distinguishes humans from machines.
- Examples (like programming a telephone answer-

ing system) that identify the broad interdisciplinary utility of computers and algorithmic problem solving in the modern world.

- Ethical issues that relate to computers and networks (including security, privacy, intellectual property, the benefits and drawbacks of public domain software, and the reliability of information on the Internet), and the positive and negative impact of technology on human culture.
- Identification of different careers in computing and their connection with the subjects studied in this course (for example, information technology specialist, Web page designer, systems analyst, programmer, CIO).

Level III Standards (11 or 12)

By the end of this course, students should understand or have a working knowledge of these topics:

- Fundamental ideas about the process of program design and problem solving, including style, abstraction, and initial discussions of correctness and efficiency as part of the software design process.
- Simple data structures and their uses.
- Topics in discrete mathematics: logic, functions, sets, and their relation to computer science.
- Design for usability: Web page design, interactive games, documentation.
- Fundamentals of hardware design.
- Levels of language, software, and translation: characteristics of compilers, operating systems, and networks.
- The limits of computing: what is a computationally "hard" problem? (for example, ocean modeling, air traffic control, gene mapping) and what kinds of problems are computationally unsolvable (for example, the halting problem).
- Principles of software engineering: software projects, teams, the software life cycle.
- Social issues: software as intellectual property, professional practice.
- Careers in computing: computer scientist, computer engineer, software engineer, information technologist.

Level IV (11 or 12)

At this level, interested and qualified students should be able to select one from among several electives to gain depth of understanding or special skills in particular areas of computer science. These electives include, but are not necessarily limited to:

- Advanced Placement (AP) Computer Science;
- A projects-based course in which students cover a topic in depth; and
- A vendor-supplied course, which may be related to professional certification.

Why Teach Computer Science Education in K–12?

Education leaders want curriculum that is preparing students with the knowledge and skills they will need to succeed in the workforce or college. This intention is often squeezed by the limitations of the school day and fierce competition among subjects for inclusion. As education policy makers make tough decisions about which subjects will meet the needs of citizens and industry in the 21st Century, they want to know how a subject might make their students successful. Teaching computer science in K–12 meets students' needs in three ways:

- Students gain a deeper knowledge of the fundamentals of computing, which—as computing becomes ubiquitous—is critical foundational knowledge that will serve them well throughout their lives;
- Students are exposed to a field that drives innovation and in which job prospects remain strong despite the current economic challenges; and
- Students gain critical knowledge and skills proven to bolster their success in higher education academic pursuits.[xviii, xix]

Computer science education is strongly based upon the higher tiers of Bloom's cognitive taxonomy as it involves design, creativity, problem solving, analyzing a variety of possible solutions to a problem, collaboration, and presentation skills. Through studying computer science, students develop and extend their logical thinking and problem-solving skills. These skills can then be applied to real world problems—mathematical and otherwise. Further, students who study computer science at the secondary level and have previous experience with technology demonstrate improved readiness for post-secondary studies.[xx]

Computer science underpins the technology sector, which has made tremendous contributions to the domestic economy, as well as numerous other economic sectors that depend on innovative, highly skilled computer science graduates. Computing touches everyone's daily lives. Securing our cyber-infrastructure, voting in elections, protecting national security, and making our energy infrastructure more efficient are among numerous issues dependent on computing and a strong computing-savvy workforce.

Where is Computer Science in "STEM"?

The national urgency to improve science, technology, engineering, and mathematics (STEM) education is palpable as officials from the President of the United States to local school boards have called for reforms in these areas. What is not clear to policy makers at all levels is that computer science is frequently left out of these initiatives. Typically, it is neither explicitly nor discretely part of the "core" courses within STEM. And the evidence of this problem is clear. The Common Core State Standards Initiative and the National Research Council framework for science, technology, and engineering concepts have tried to define the standards and concepts that underpin STEM education. Computer science does not exist in these documents despite being taught across the United States, having an Advanced Placement exam, and in some states having computer science courses count as either a mathematics or science credit in secondary education.

Many observers assume that computer science is the "T" in STEM, but this is, by and large, not the case. Computer science education focuses on teaching fundamentals of computing and computational thinking just as core mathematics, physics, chemistry, and biology courses teach fundamental concepts. K–12 technology education focuses primarily on the use of computing as a tool to solve problems in other fields, specifically the use of computing applications in pursuit of that goal. Computer science is not now, nor has it ever been, just about the use of computers or computer applications. It includes the knowledge and skills necessary to build the next generations of software and hardware tools that the world needs.

The "T" in STEM is a diverse space that includes many aspects, only some of which relate to computing. Further, when computer science courses are considered part of technology education, they often focus on post-secondary school vocations in information technology versus the fundamental knowledge of computing that college-bound students require. And, unlike science and mathematics courses, technology courses are not in the "core" of what students must take to graduate.

Computer science is driving innovation in all STEM disciplines but it is also a distinct discipline with an extensive body of knowledge and needs to be treated as such within federal, state, and local policies. Efforts to merely incorporate computer science into other STEM disciplines do a disservice to all sides. They ignore the complexity of the discipline and the importance of the knowledge and skills that it provides. Ultimately, the failure to recognize and support computer science as a distinct entity within STEM will exacerbate an already growing workforce and innovation crisis and fail students who depend upon our foresight to prepare them for the future.

Computer Science Teacher Certification

In 2004 CSTA launched a national survey of 14,000 teachers who identified themselves as computer science, computer programming, or Advanced Placement Computer Science teachers. As part of this survey, teachers were asked to indicate whether or not they were required by their state to have either a computer science certification (identifying computer science as their major teachable subject) or an endorsement (a secondary certification in another discipline) in order to teach computer science at the secondary school level. Upon first look, the data returned by the survey participants seemed to make sense. Half the teachers indicated that their state required either a major certification or an endorsement in computer science and the other half indicated that no certification was necessary.

Researchers only realized that there was a problem with the data when they began examining the responses on a state-by-state basis and discovered that half the teachers in each state were saying 'Yes' and half were saying 'No.' In their report on the 2005 survey results, Eric Roberts of Stanford University and Greg Halopoff[xxi] of the Clarke County School District reported these results as follows.

"Nationally, the results tend to cluster in the middle, with about the same number of negative and positive responses to each of the yes/no questions. One's intuition would be that this sort of balance masks much more significant diversity at the state level. For example, if half of the states required certification and half did not, the overall numbers would tend to hover around 50 percent without providing any interesting insights. That situation, however, is not supported by the state-by-state breakdowns. The responses within most states show a surprisingly inconsistent perception. Nine states, including some with reasonable numbers of respondents like Colorado, split perfectly down the middle on this question, with exactly 50 percent saying that their state considered computer science to be certifiable and the other half taking the opposite view. The only conclusion that seems to jump out of these data is that the teachers themselves often have a poor understanding about rules and administrative structures within their own state, at least insofar as computer science certification is concerned."

Fearing that they had not provided sufficient explanation of the terms "certification" and "endorsement," CSTA revised the question (providing the missing explanatory text) and included it again in its 2007 national survey. Despite these revisions, the results were the same. Clearly, at least 50% of the teachers in each state had no idea what the certification requirements in their states actually were.

Because of this overwhelming evidence of teacher confusion, CSTA decided that the only way to gather reliable information on computer science teacher certification was to survey the person in each state directly responsible for overseeing compliance with the state's teacher certification regulations. This research took more than two years and uncovered an entirely unexpected series of difficulties. First, in many states it was exceedingly difficult to identify and contact the person of responsibility. Even once these people had been identified and contacted, the researchers were appalled to learn that many of these people had no idea what computer science was. In her report on this study, Ghada Khoury[xxii] reported the following:

"Many states did not seem to have a clear definition or understanding of the field "Computer Science" and exhibited a tendency to confuse Computer Science with other subject areas such as: Technology Education/Educational Technology (TE/ET), Industrial or Instructional Technology (IT), Management Information Systems (MIS), or even the use of computers to support learning in other subject areas."

In addition, this study revealed that computer science teacher certification not only varied markedly from state to state, but that even within a given state, the reporting of requirements by the states was inconsistent. For example, some states indicated that they do require teachers to have either computer science certification or endorsement to teach a computer science course but then indicated "not applicable" when asked to indicate the levels at which this certification or endorsement was required (elementary, middle, or secondary school).

The profound lack of clarity regarding the current teacher certification requirements is not limited to teachers but is systemic at all levels of the education system, including at the levels where policy is made and enforced. It is also consistent with, and in many ways perpetuates, the continuing crisis in computer science education in this country. When the teachers and administrators do not understand the policies regarding teacher preparation or those policies reflect a profound disconnect with the actual body of knowledge of the discipline, how is it possible to ensure that the teachers bring to the classroom the required knowledge to promote much-needed student learning? And when poor or non-existent certification requirements dissuade teacher preparation programs from preparing teachers to teach in one of the few areas where there are consistent job growth and worker shortages, how can we expect to prepare students adequately for the future?

Findings

- Methodology
- Findings On Standards
- Findings On Graduation Requirements
- Limits Of This Study
- Conclusion

Methodology

Data Collection

During 2009 researchers gathered data from the websites of the appropriate state education authority in all 50 states and the District of Columbia. Curriculum standards were sought for grades 9–12 in math, science, and technology. Technology standards were defined either as educational technology student learning outcomes/expectations or standards for supplemental programs (career training) that involved computing. In some states the technology standards were embedded within the math and science standards.

The researchers updated the data again in early 2010 and in the Spring of 2010, the data was vetted through the CSTA's Leadership Cohort, which is a group of two expert computer science teachers from each of the 50 states.

Data Analysis

In gathering the standards data, the researchers noted where it was found within the state standards (i.e. technology, mathematics, or science). A standard from the *ACM/CSTA Model Curriculum for K–12 Computer Science* was marked as being adopted if it was found within the state standards for technology, mathematics, or science. For example, the ACM/CSTA Level II standard for grades 9–10 "The basic steps in algorithmic problem-solving" was found within the technology standards in Alabama, within the mathematics standards in Louisiana, and did not exist in Colorado's standards. The results varied with some states having adopted ACM/CSTA standards within all three curriculum areas and some states in only one.

The researchers were very liberal in the analysis; that is, they required minimal evidence to consider a particular ACM/CSTA standard as adopted. If the state standards made any reference to the general idea detailed in the ACM/CSTA standards, it was marked as adopted. This methodology tended to yield a very generous estimate of adoption, but it also makes the gaps identified by the researchers even more startling.

The results were then categorized in two ways:

- First, the adopted standards were organized according to the Level I, II, and III classification in the *ACM/CSTA Model Curriculum for K–12 Computer Science* (see "What is Computer Science Education?" on p. 26 for more details).
- Second, the researchers then classified the standards using the descriptors provided in the National Academies report *Being Fluent with Information Technology: Concepts, Capabilities and Skills*.

Findings on Standards

Despite a national push for preparing students to succeed in the 21st Century, on average only 55% of the ACM/CSTA model computer science education standards appear in the state standards for grades 9-12 across the nation. The ACM/CSTA standards are broken out by level. For Level I (K-8th grade and very similar to ISTE's NETS standards) there is a 70% average adoption. For Level II there is a 35% adoption and for Level III there is only a 30% adoption (See Figures 4,5,6,7).

Turning to the states, there are 16 states with no model curriculum standards adopted at Level II and 22 states with no model curriculum standards adopted at Level III. Figure 5 shows the percent of standards adopted for each state at Level II and Figure 6 shows the same for Level III.

These adoption rates are likely to get worse as states move to adopt the Common Core State Standards Initiative (CCSSI).[xxiii] This initiative seeks to harmonize state content standards for mathematics and English Language Arts, and 34 states and the District of Columbia have agreed to adopt them. Given that many states have some computer science standards within the mathematics framework and that the CCSSI has no mention of most of our model K–12 standards, it seems likely that states will drop some of the computer science standards within their state educational plans.

While looking at state standards by Level gives an interesting picture of the types of courses that are implemented, the IT Fluency breakdown offers a window into the way that technology is valued as a part of the educational development of our students (See Figure 8). Recall the three IT Fluency categories as defined by the ACM/CTSA's K–12 model curriculum report are Concepts, Capabilities and Skills.

Using that taxonomy yields the following results:

Concepts: emphasize one of the 10 basic ideas that, at a high level, define modern computers, networks, and information (see Figure 9). Examples include computer organization, information systems, networks, digital representation

FIGURE 4 State by State Adoption of Level I Computer Science Standards

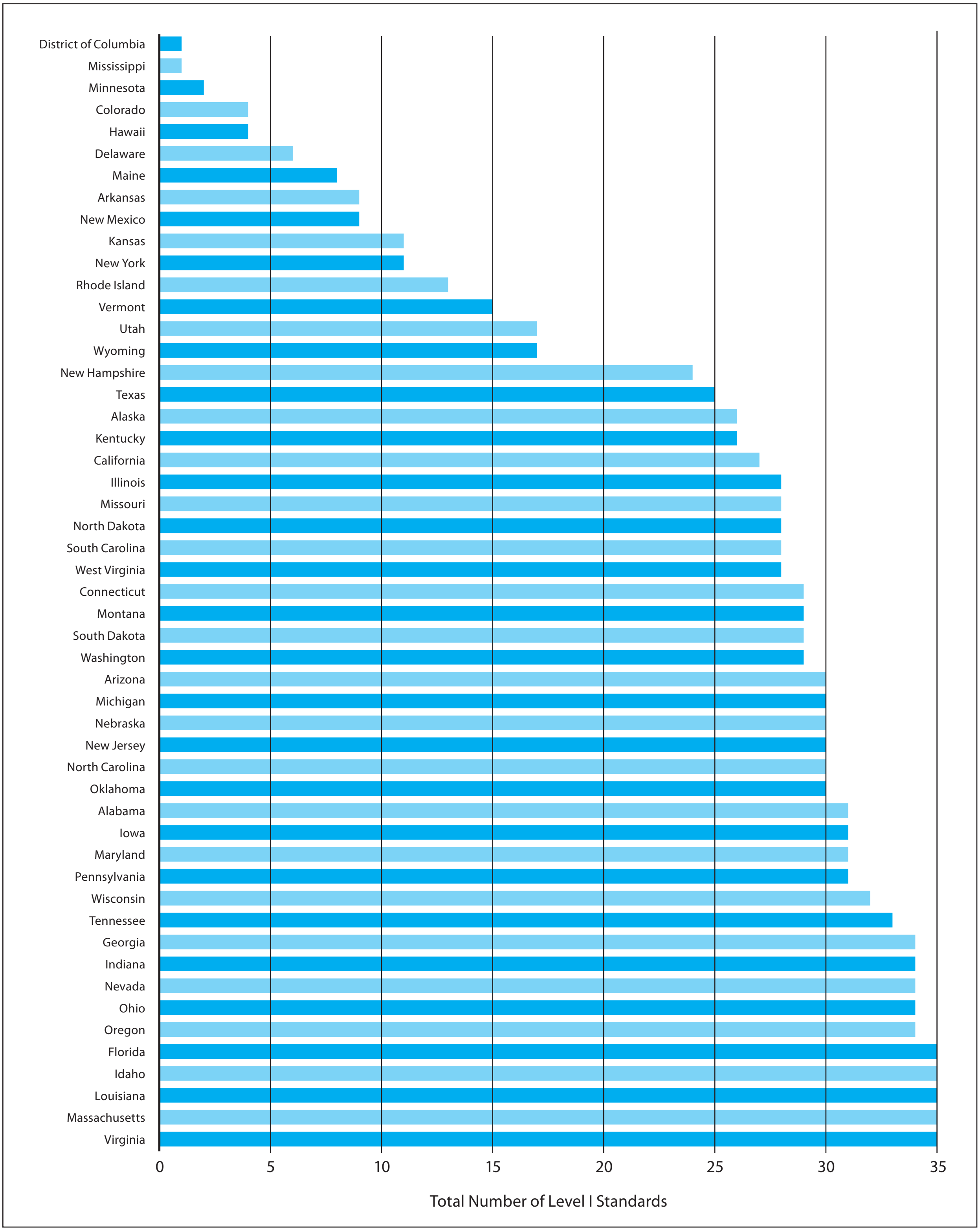
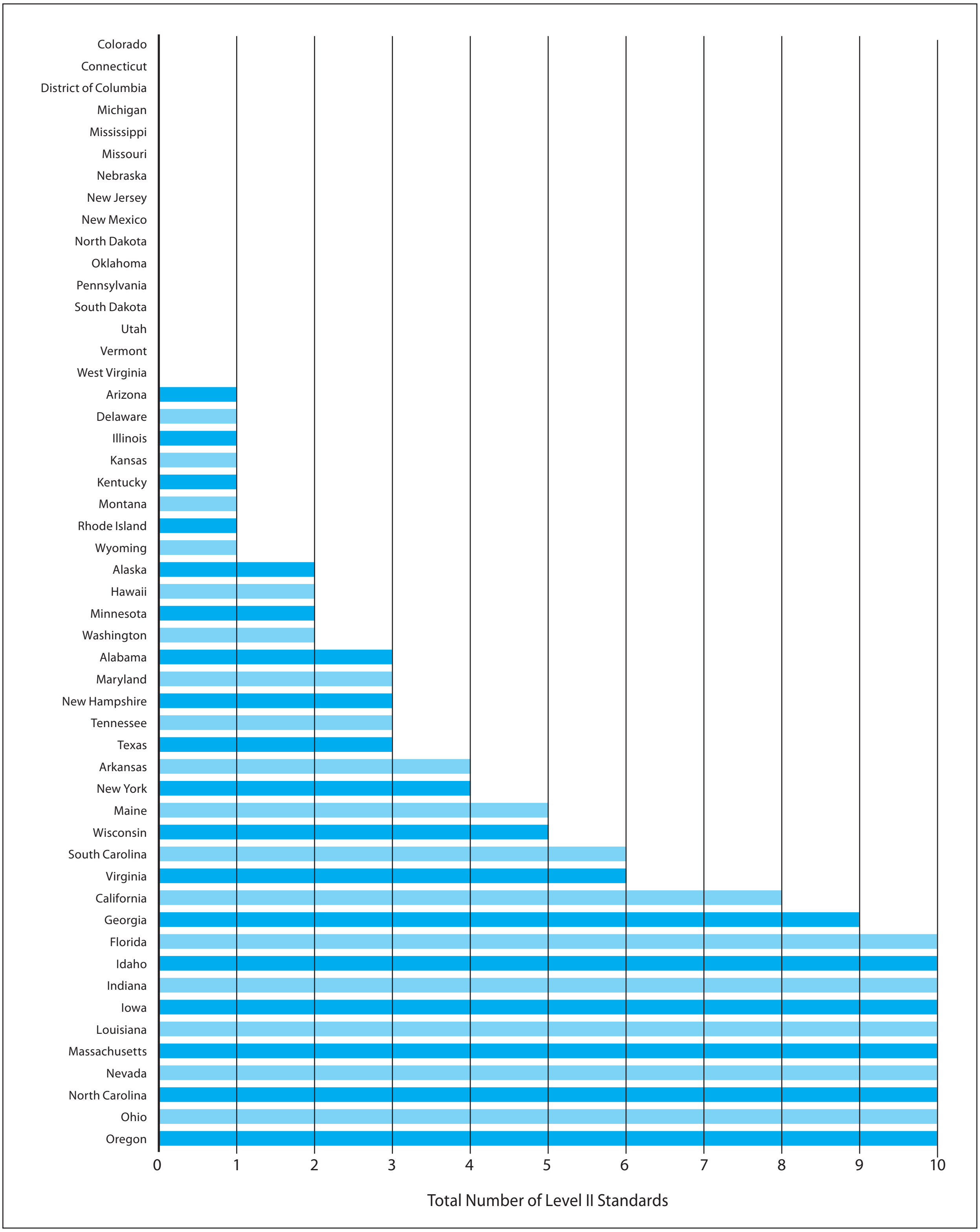


FIGURE 5 State by State Adoption of Level II Computer Science Standards



of information, information organization, modeling and abstraction, algorithmic thinking and programming, universality, limitations of information technology, and societal impact of information technology.

Total Concepts Standard Adoption: 37%

Level 1: (41%)

- Understand how 0's and 1's can represent information (24%)
- Understand how to arrange/sort information (29%)
- Develop an understanding of an algorithm (39%)
- Understand hardware, software, and algorithms (31%)
- Understand graphs (100%)
- Fundamentals of logic (41%)

Level 2: (36%)

- Principals of computer organization (IO, memory) (29%)
- Basic steps in algorithmic problem solving (45%)
- Components of computer networks (31%)
- Organization of Internet elements, Web design, search engines (33%)
- Hierarchy and abstraction in CS (31%)
- Connections between CS and math (29%)
- Examples of interdisciplinary CS (51%)

Level 3: (28%)

- Simple data structures and their uses (33%)
- Discrete mathematics (35%)
- Hardware design (23%)
- Characteristics of compilers/OS/networks (23%)
- Limits of computing (23%)
- Principals of software engineering (31%)

Capabilities: emphasize one of the 10 fundamental abilities for using computing to solve a problem (see Figure 10). Examples include the ability to engage in sustained reasoning, manage complexity, test a solution, manage faulty systems and software, organize and navigate information structures and evaluate information, collaborate, communicate to other audiences, expect the unexpected, anticipate changing technologies, and think abstractly about IT.

Total Capabilities Standard Adoption: 60%

Level 1: (72%)

- Work cooperatively while using technology (67%)
- Positive social and ethical behaviors (75%)
- Responsible use (75%)
- Use technology resources for problem solving (88%)
- Gather information and communicate with others using technology (71%)
- Discuss common uses of technology in every day life (76%)
- Discuss issues about responsible use (78%)
- Select tools appropriate for problems (88%)
- Evaluate accuracy, relevance, of electronic information (67%)
- Identify and solve routine hardware and software problems (37%)
- Understand changes in IT and their effects (71%)
- Legal and ethical behaviors (73%)
- Collaborate using telecommunications and develop solutions (67%)
- Select appropriate tools and technologies to solve problems (92%)
- Evaluate accuracy of electronic information sources (57%)

Level 2: (27%)

- Computers modeling intelligent behavior and difference between humans and computers (22%)
- Ethical issues relating to networks, IP and public domain (27%)
- Identification of careers in CS (31%)

Level 3: (28%)

- Software design process (29%)
- Design for usability (22%)
- Social issues (31%)
- Careers in computing (31%)

Skills: emphasize one of the 10 abilities to use today's computer applications in one's own work (see Figure 11). Examples include the ability to set up a personal computer; use basic operating system features; use a word processor and create a document; use a graphics or artwork package to create illustrations, slides, and images; connect a computer to a network; use the Internet to find information and resources; use a computer to communicate with others; use a spreadsheet to model simple processes or financial tables; use a database system to set up and access

FIGURE 6 State by State Adoption of Level III Computer Science Standards

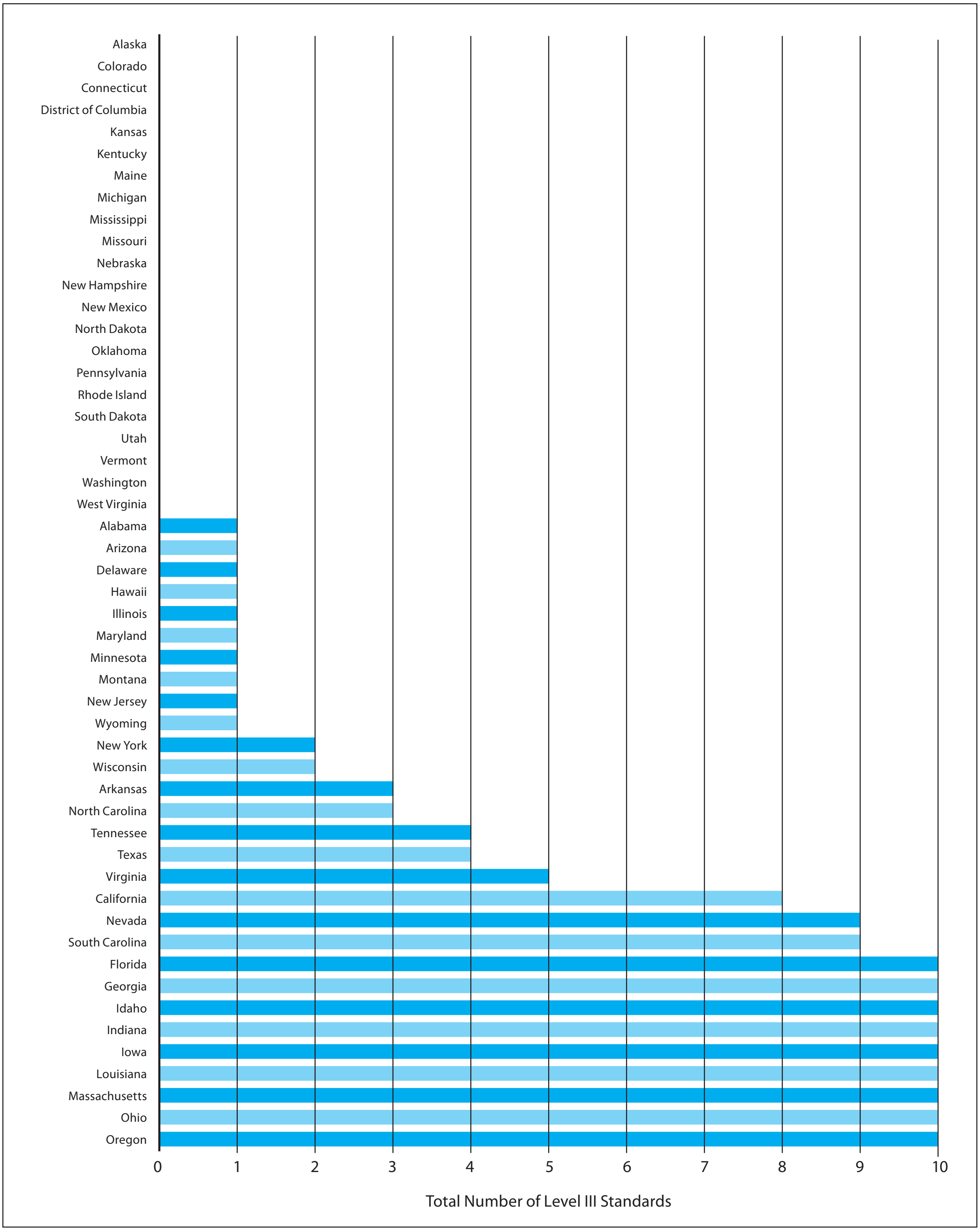


FIGURE 7 **State by State Adoption of Computer Science Standards by Level**

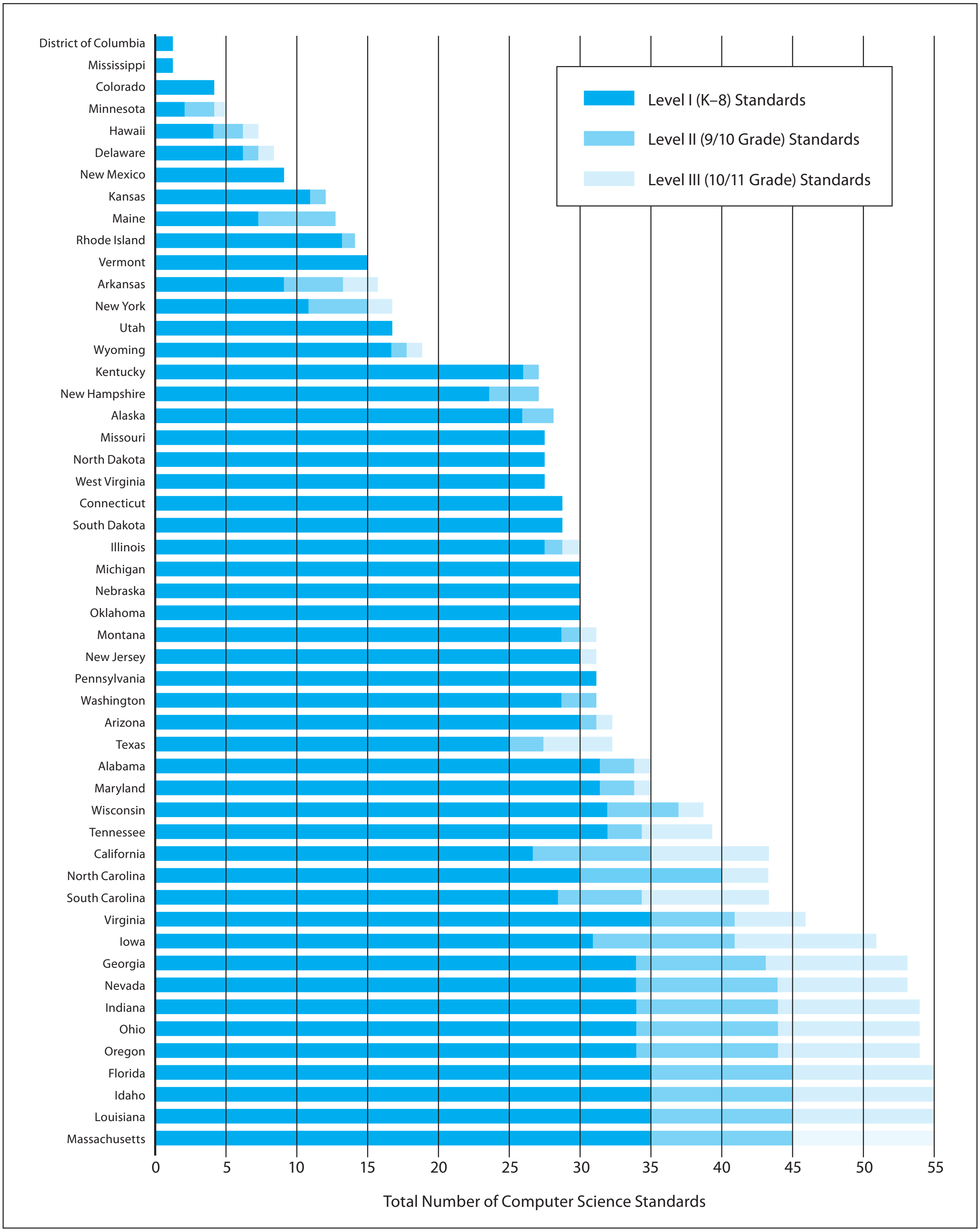


FIGURE 8 State by State Adoption of Computer Science Standards by Fluency Category

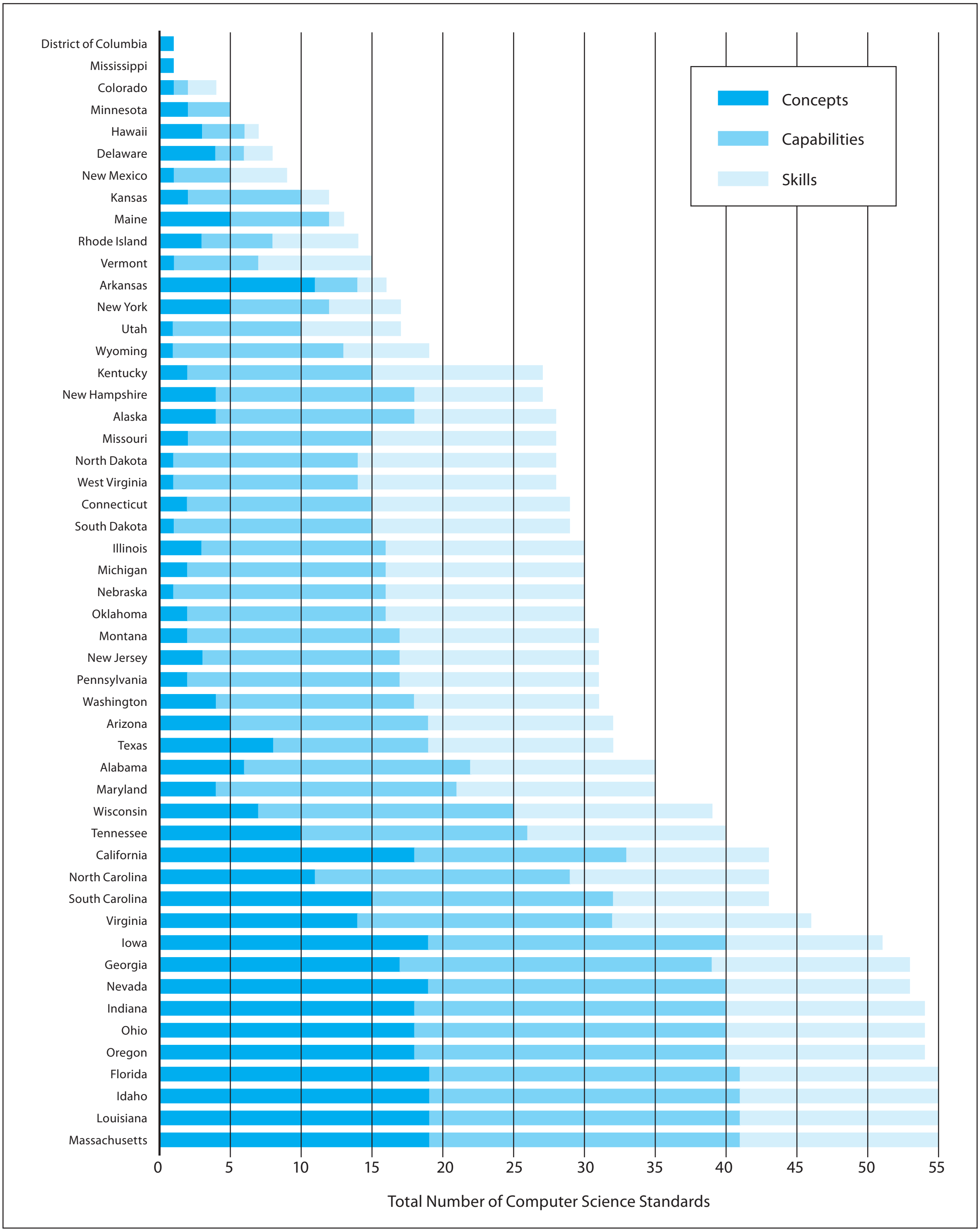


FIGURE 9 **State by State Adoption of Computer Science Standards — Concepts**

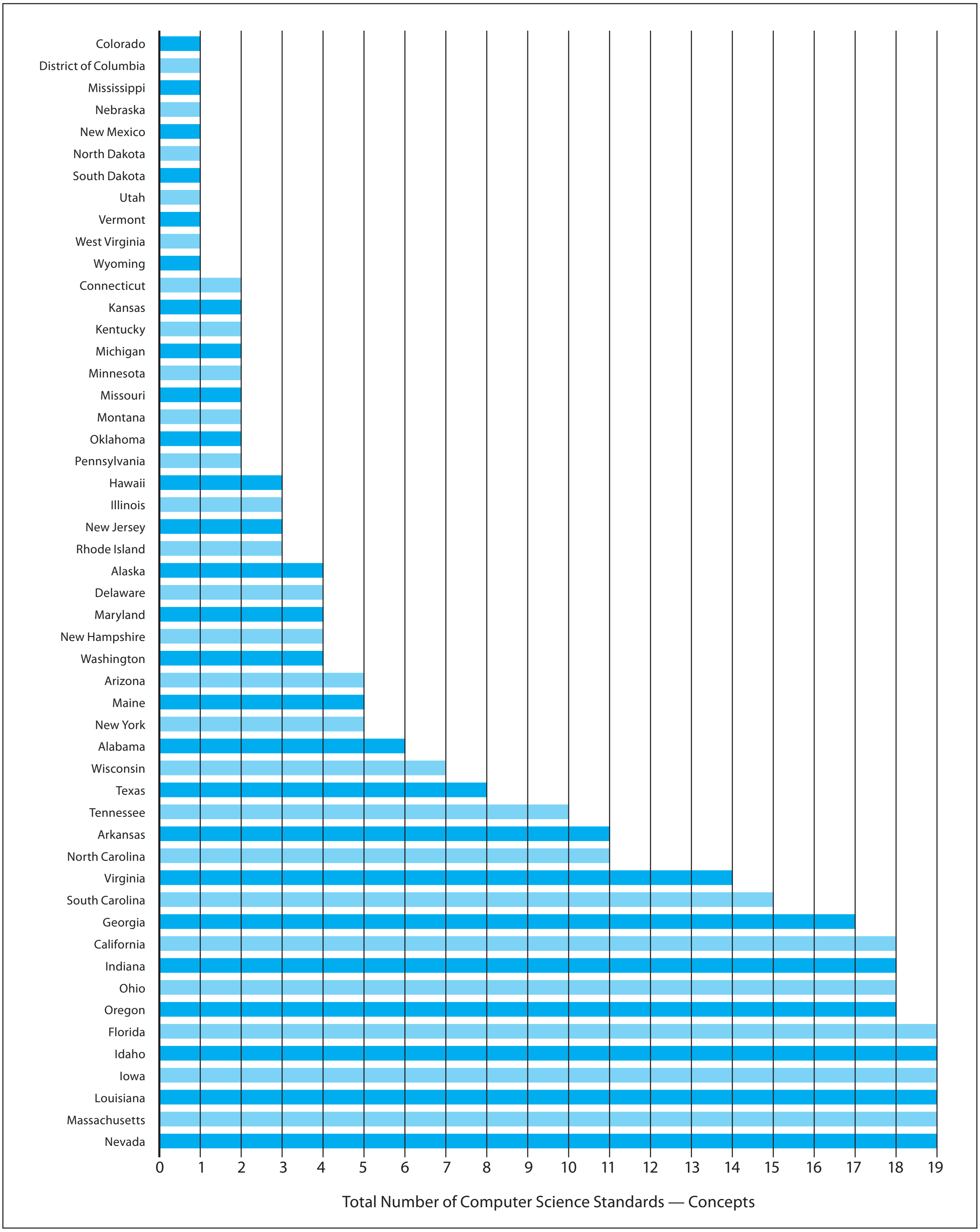


FIGURE 10 **State by State Adoption of Computer Science Standards — Capabilities**

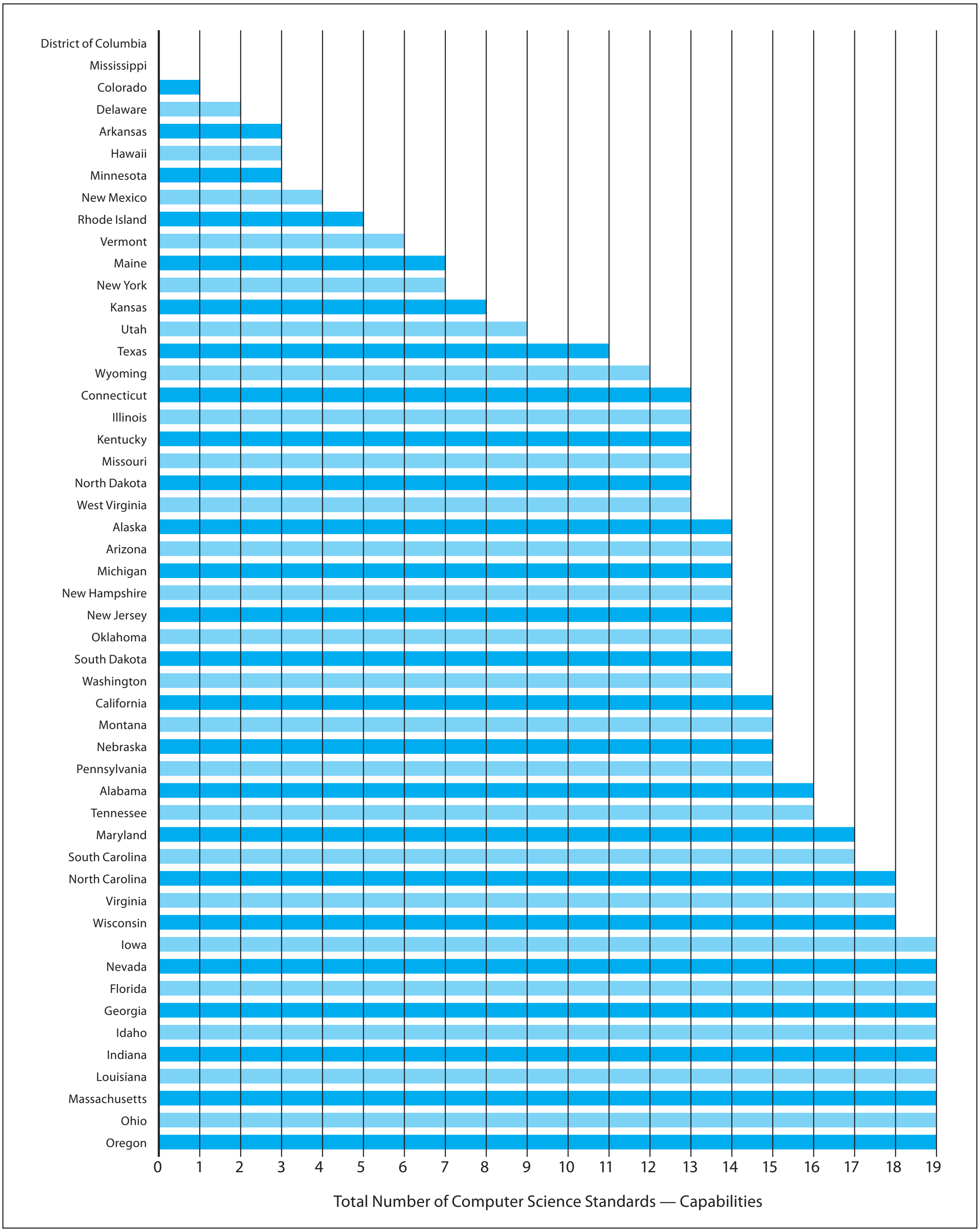
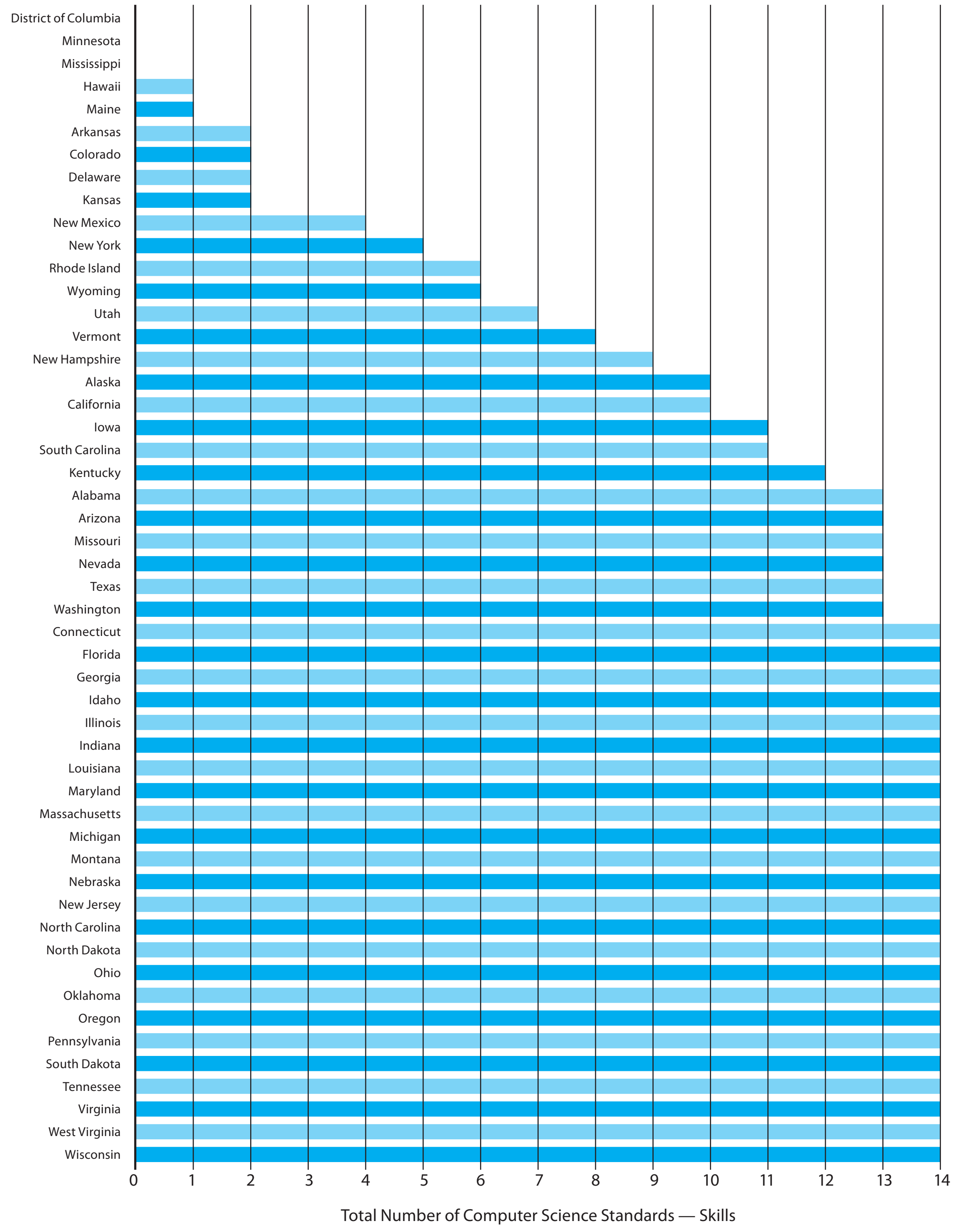


FIGURE 11 **State by State Adoption of Computer Science Standards — Skills**



information; and use instructional materials to learn about new applications or features.

Total Skills Standard Adoption: 73%

Level 1: (70.9%)

- Use Standard I/O (69%)
- Use a computer for learning (71%)
- Communicate about technology (63%)
- Use multimedia to support learning (65%)
- Create/develop appropriate multimedia with support (65%)
- Use I/O and keyboarding skills (61%)
- Use productivity tools (75%)
- Use technology tools (digital cameras, presentation, web, scanner, multimedia) (69%)
- Use telecommunications to get remote information (69%)
- Use online resources for problem solving (78%)
- Use technology tools (calculators, digital probes, videos) for problem solving (82%)
- Use content-specific technology tools (calculators, digital probes, web) to support learning (92%)
- Apply productivity and multimedia tools (69%)
- Make products including video, web pages, etc., using technology (67%)

Findings on Graduation Requirements

The researchers also surveyed the secondary school graduation requirements of all 50 states and the District of Columbia. Two particular areas of interest were examined. First, the study examined how a rigorous computer science course (such as an ACM/CSTA Level III course or the Advanced Placement Computer Science course) would be counted in graduation requirements. Second, they investigated whether the state had a Technology Literacy/Computer Applications requirement for graduation. Some states had a specific Technology Literacy requirement, some states had it as one item on a menu of constrained electives, and some states did not require it at all.

Forty states count an upper-level computer science as an unconstrained general elective credit. This means that computer science courses satisfy no particular category and thus fall into the "general credit toward graduation," as would a course in any random subject. Nine states allow computer science to count as a mathematics credit. Seven as an advanced mathematics credits.

Only one state (Georgia) counts computer science as a science credit. Six states defer the mapping of course credits to individual school districts whose resulting categorization of computer science courses ranges widely.

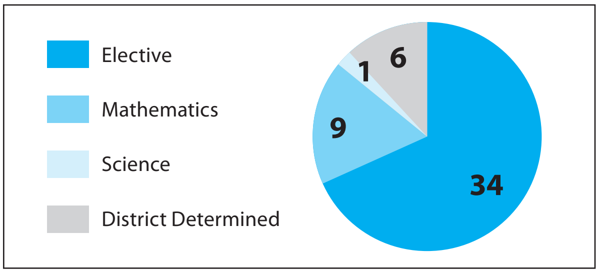
Determining how computer science factors into state graduation requirements yielded fairly consistent (and distressing) results (See Figure 12). The "credit" picture is clouded by differing state authorities and policies regarding who controls what each student must, at a minimum, learn. While the researchers discovered that it was fairly easy to find the state graduation requirements, determining precisely if or how computer science courses mapped into those requirements was often exceedingly challenging. For example, the researchers had to dig down through several levels of information to determine that New York allows Advanced Placement Computer Science to count as an advanced math elective. This lack of clarity and accessibility with regard to graduation requirement information is further compounded in states that defer binding specific courses to graduation requirements to local school districts. The researchers also learned that in some states (such as Kansas) the State Board of Regents (which control requirements for access to institutions of higher education in the state) have different requirements than the state boards of education that control the schools. (It is outside the scope of this study to review instances such as this in all 50 states.)

Limits of this Study

Education standards exist at the state policy level. This study seeks to paint a picture of the states' policies with regard to computer science as reflected in those standards. This is a fluid space with many proposals and draft standards always in flux and not necessarily well advertised. The findings in this document represent the researchers best efforts to accurately reflect state standards and requirements as of February, 2010.

The results and recommendations reflect the policies and practices that exist at the state level outside schools and classrooms. Standards can be attached to a specific course or state educational goals that allow individual schools to shape classes to their individual needs. This

FIGURE 12 How Computer Science Courses Count Toward Graduation Requirements



means that simply because a set of standards exists that does not imply that courses exist to carry them out or that they are being adequately taught. In many states, standards are promulgated without specific course definitions leaving it up to individual schools to implement those standards across the courses in their curriculum.

Technology standards exist in three forms. Some states embed them inside other disciplines, as in the case of calculator use in mathematics or science courses. Other states define generic technology standards unattached to a particular discipline, implying all teachers should be creating content addressing these standards within their own discipline. Still other states, in addition to one of the previous models, set standards for specialty career readiness (formerly vo-tech) programs that are often implemented at select centers or pullout programs that may not exist inside the school. In the case of computer science, the mapping of applicable state standards to computer science programs in schools is widely varied. We know there are schools with excellent computer science programs and, in the same state, schools with little or no computer science offerings and that this is consistent with state standards.

Conclusion

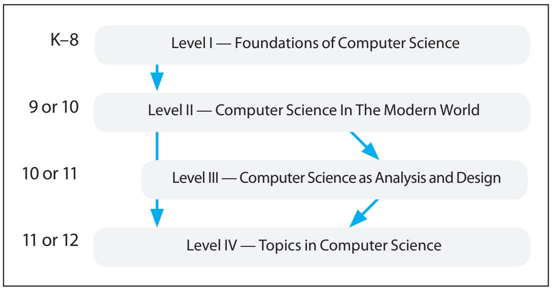
Computer science education in K–12 is vital, but without specific intervention at all levels of government to make it stand on its own within the K-12 education landscape it will continue to fade from our schools. This will hurt not only the field of computing but also all the fields that depend on innovations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

Description of the State Report Cards

On each state report card you will find two charts summarizing our findings on the standards, which reflect the two different ways to organize the results noted early in the report.

The first graph shows the alignment of a state's standards in each of the three curriculum levels outlined in the ACM/CSTA Model Curriculum (*Level I* — grades K–8, *Level II* — grades 9–10, and *Level III* — grades 11–12). An analysis of gaps in *Level IV* standards was not included in this study because there are no standards listed, instead potential course topics are offered. Below is a chart on the ACM/CSTA Model Curriculum to serve as a guide:

FIGURE 1 Framework for ACM/CSTA Model Computer Science Standards


K–8
Level I — Foundations of Computer Science
9 or 10
Level II — Computer Science In The Modern World
10 or 11
Level III — Computer Science as Analysis and Design
11 or 12
Level IV — Topics in Computer Science

The second chart shows the implementation of standards organized along the information technology fluency categories (Concepts, Capabilities and Skills) created by in the National Academies study *Being Fluent with Information Technology.*

More information on the ACM/CSTA levels and the standards themselves can be found in Section III (What is Computer Science). More information on the *Being Fluent with Information Technology* categories can be found in Section IV (Methodology).

Below each chart are three characterizations of the data:

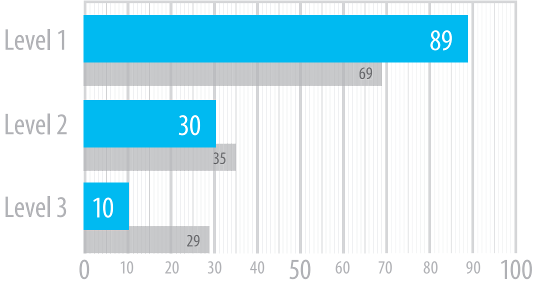
Standards: Shows the percentage of the *ACM/CSTA Model Curriculum* standards contained in the three standards areas (math, science, technology). For example, we found 15 standards, 10 (or 67%) of which were in technology, 5 (or 33%) of which were in mathematics. If there is only one percentage (for example, "84% of standards can be found in technology"), the remainder of the standards can be found in either science or mathematics.

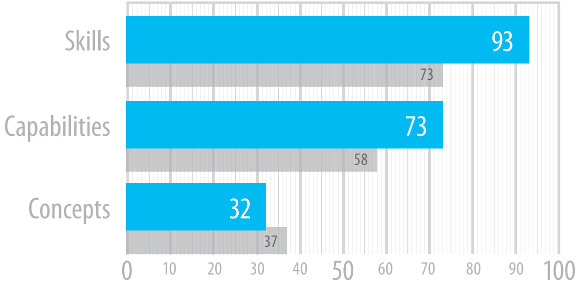
Content area: Describes some of the content areas in the state and the findings of the research related to where the standards are located for that state.

Graduation Requirements: Describes whether and how computer science can be counted as a graduation credit.

Alabama

Implements:





Standards

84% of standards can be found in the technology standards for the state.

Content Area

There are no course definitions for any computer science or technology courses at the high school level. There is a business document titled "Computer Essentials 6-7-8 Grades" which lays out a course description for middle school students. The introduction to the standards document states "Local school systems should develop local curriculum plans that incorporate these statements of what students should know and be able to do."

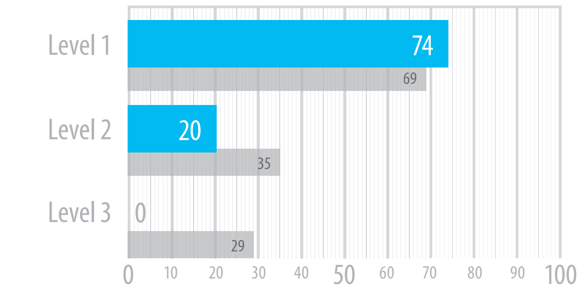
Graduation Requirements

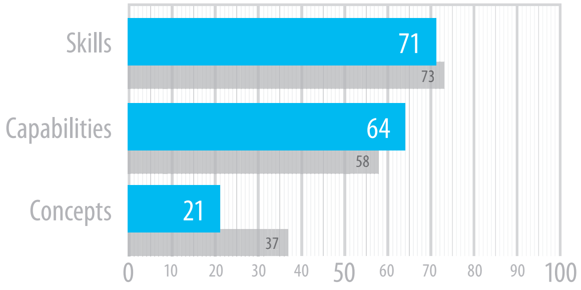
Alabama has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Alaska

Implements:





Standards

66% of standards can be found in the technology standards for the state, with 21% of standards found embedded in the math standards at the 9–12 level.

Content Area

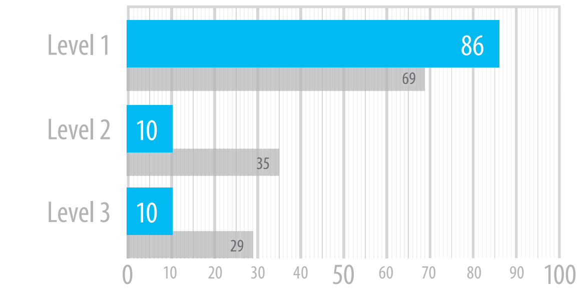
The state technology standards are explicitly designed to be integrated into all other disciplines. “Technology is a tool to be integrated across all content areas. The Alaska content standards in technology do not define technology as an isolated academic field.”

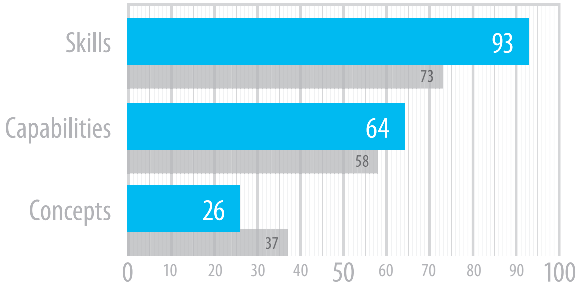
Graduation Requirements

Alaska does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation. There is some freedom at the district level for setting graduation requirements.

Arizona

Implements:





Standards

72% of standards can be found in the technology standards for the state, with 16% of standards found embedded in the science standards at the 9–12 level.

Content Area

The technology standards are assumed to be implemented by all disciplines. The rationale document for the standards states: “[We] reviewed technology that is currently integrated into other content area standards with the vision that as other standards are revised, technology will be seamlessly integrated.” In addition to providing standards, the state also provides a variety of resources and lesson plans that demonstrate technology integration in a variety of disciplines.

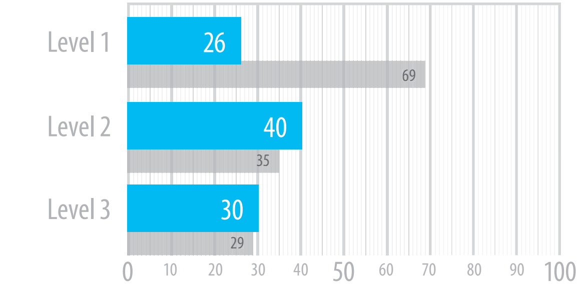
Graduation Requirements

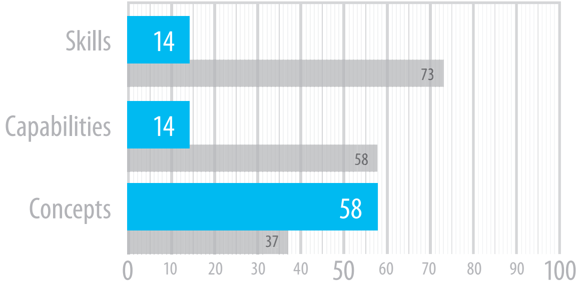
Arizona does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation. Some districts count Computer Science as a Business credit for their students.

🟦 State Results ▨ National Average

Arkansas

Implements:





Standards

93% of standards can be found in the math standards for the state.

Content Area

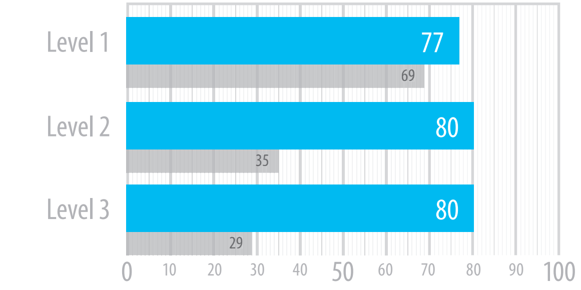
The high school math framework includes a computer mathematics course. There are business technology standards for teachers, however the state technology plan is explicit that local school districts will be responsible for creating their own plans and implementing the NETS standards locally. The plan also references the 21st Century Skills Partnership.

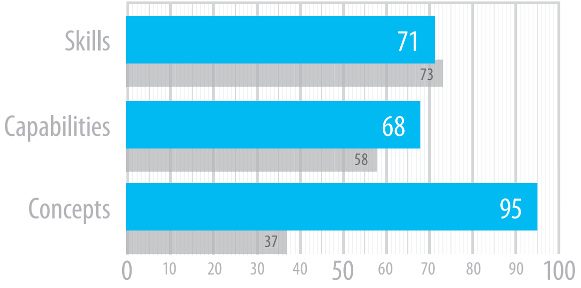
Graduation Requirements

Arkansas does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

California

Implements:





Standards

95% of standards can be found in the technology and business standards for the state.

Content Area

The mathematics framework document contains a chapter on the use of technology, while the career technology framework includes both engineering and an IT section with programming courses.

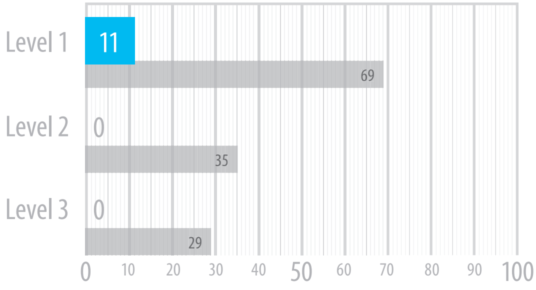
Graduation Requirements

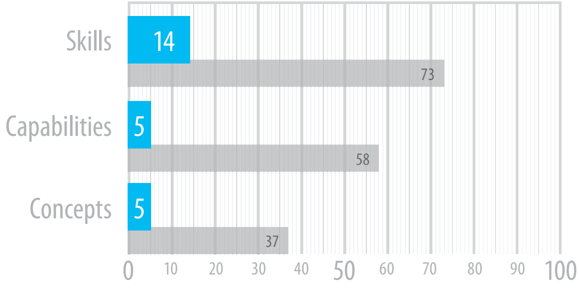
California does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation. Some districts count Computer Science as a Math credit for their students.

🟦 State Results ▨ National Average

Colorado

Implements:





Standards

100% of standards can be found in the math standards for the state. (There were only three standards identified)

Content Area

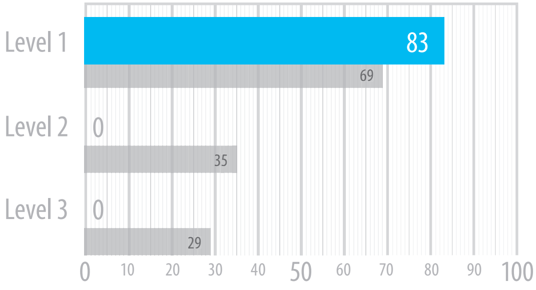
Colorado does not list any technology standards for students on their website, however it does have a department of educational technology that seems to be more about getting the technology into the schools as opposed to curricular uses of technology.

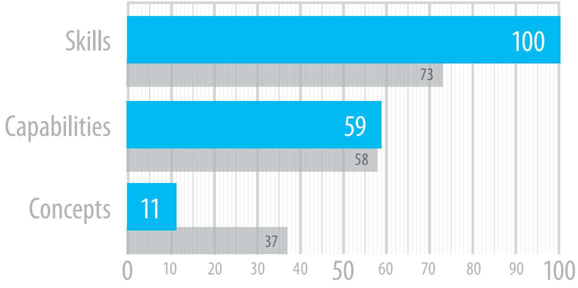
Graduation Requirements

Colorado has individual districts determine the graduation requirements. Individual districts also determine what kind of credit is awarded for Computer Science classes. Those teachers who responded to our survey reported that Computer Science counted as an elective credit.

Connecticut

Implements:





Standards

96% of standards can be found in the technology education standards for the state. (There were only three standards identified)

Content Area

There is a defined K–12 technology education standards framework for the state. There are no defined classes where these standards will be implemented and the state website does not include recommendations for integrative practices.

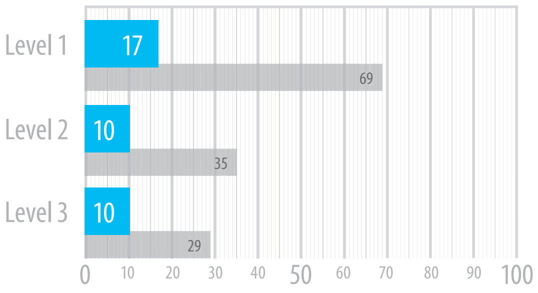
Graduation Requirements

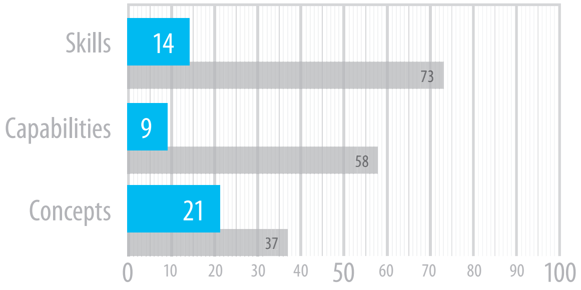
Connecticut has individual districts determine the graduation requirements. Individual districts also determine what kind of credit is awarded for Computer Science classes.

🟦 State Results ⬜ National Average

Delaware

Implements:





Standards

60% of standards can be found in the math standards for the state, with the remaining 40% found in science. (There were only five standards identified)

Content Area

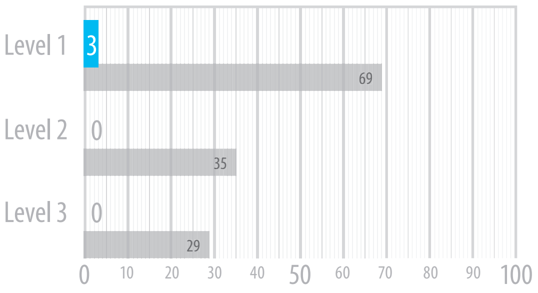
Delaware does have a technology strand embedded in the science content. From the science standards "The practice of science and the development of technology are critical pursuits of our society." Most of the standards themselves however talk about the use of technology to analyze results. "Use mathematics, reading, writing and technology when conducting scientific inquiries."

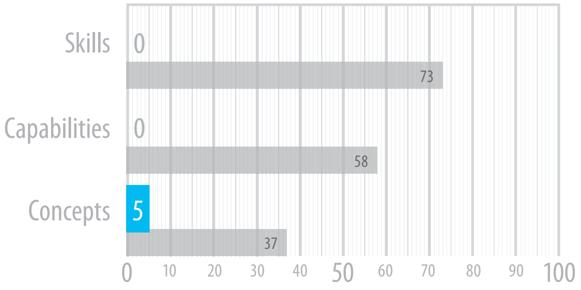
Graduation Requirements

Delaware has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

District of Columbia

Implements:





Standards

The one standard implemented can be found in the mathematics standards for the District.

Content Area

There are no technology standards for the District, and there is no statement of technology integration, requiring that technology would be taught in the regular subject classes. The science standards for 3–5th grade discuss that technology will be used in the learning of science for those grades.

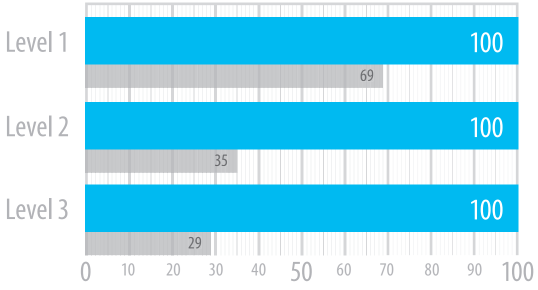
Graduation Requirements

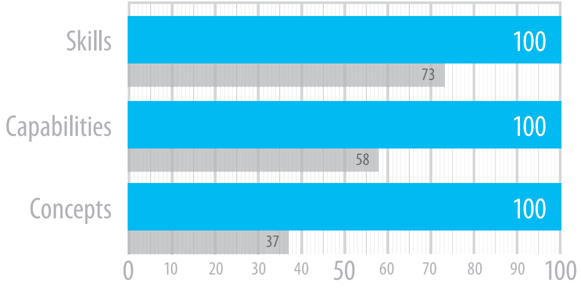
The District does not have a technology requirement for graduation, and computer science counts as an elective credit for students.

🟦 State Results ⬜ National Average

Florida

Implements:





Standards

96% of standards can be found in the technology standards for the state. There are information technology career frameworks that define standards for a variety of courses.

Content Area

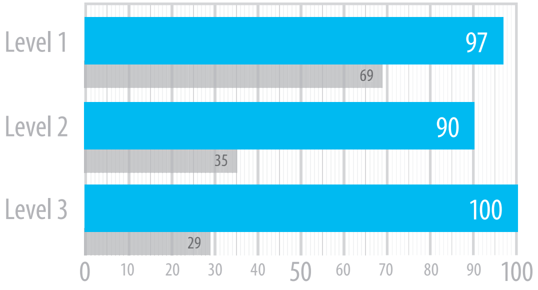
The Career and Technology Education Framework defines several pathways that contain sequences of courses for recommended concentrations. Outside of this specialized series of programs there are no general technology standards required for all students.

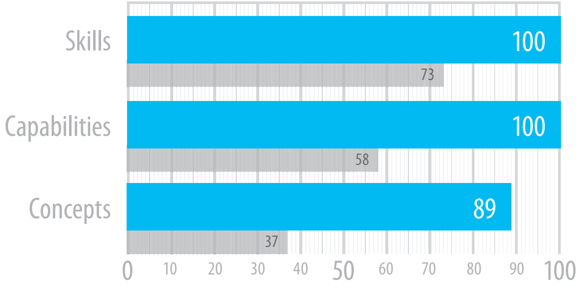
Graduation Requirements

Florida does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Georgia

Implements:





Standards

98% of standards can be found in the technology standards for the state. Georgia has a business and computer science strand in their standards that contains a variety of course descriptions for computer science.

Content Area

The business and computer science standards are part of the Career, Technical and Agricultural education section.

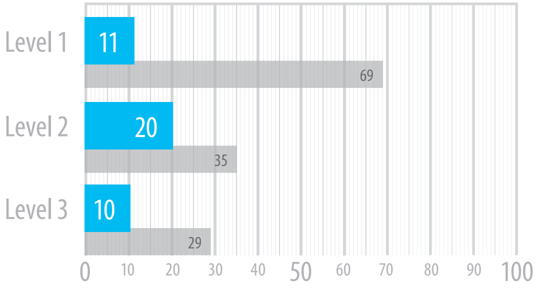
Graduation Requirements

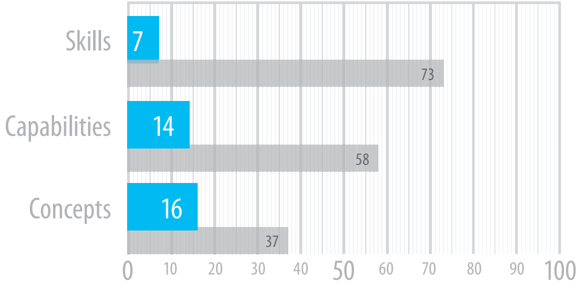
Georgia lists a technology literacy requirement as one of many potential credits to fill a mandated graduation requirement. Computer Science counts as a Science credit for graduation.

■ State Results ■ National Average

Hawaii

Implements:





Standards

57% of standards can be found in the math standards for the state. The remaining are in the career and technical standards.

Content Area

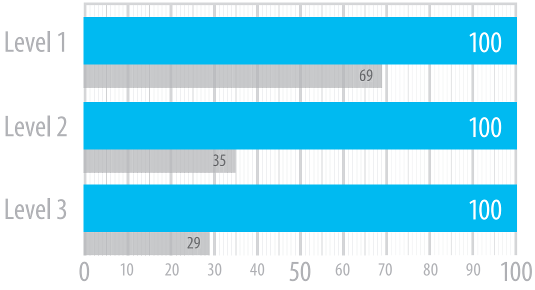
The career and technical standards contain recommendations for introducing students to career pathways and a strand to “design, modify, and apply technology effectively and efficiently solve problems.” There are no standards at the 9–12 level for this strand however.

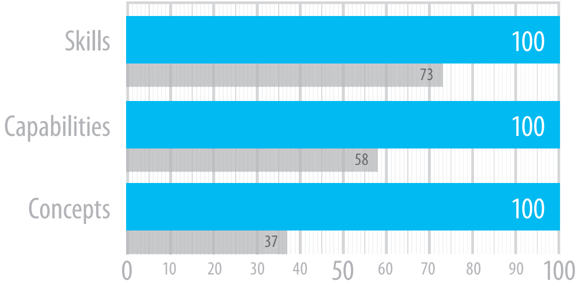
Graduation Requirements

Hawaii does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Idaho

Implements:





Standards

100% of standards can be found in the math standards for the state.

Content Area

The state has content standards for engineering and technical education with a STEM Pathway focus of Technology Education where these standards exist. While the course descriptions and titles are provided for access on the web site, schools must initiate such a program within their building.

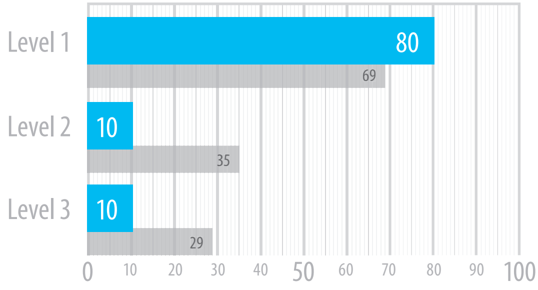
Graduation Requirements

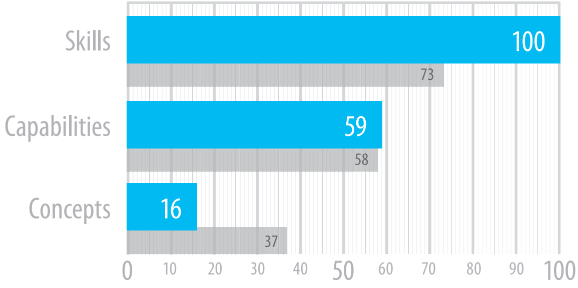
Idaho does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ▨ National Average

Illinois

Implements:





Standards

70% of standards can be found in the technology standards for the state, with 26% of the standards in the science standards.

Content Area

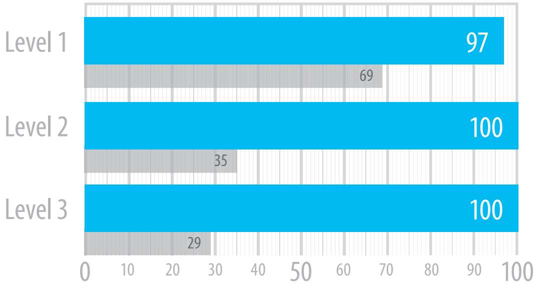
The science standards speak about integrating technology throughout the curriculum and talk about the design of technology through science. The science standards state "Technology is invented and improved by the use of scientific principles." The state technology standards provide a link to ISTE's NETS standards which align well with the ACM standards.

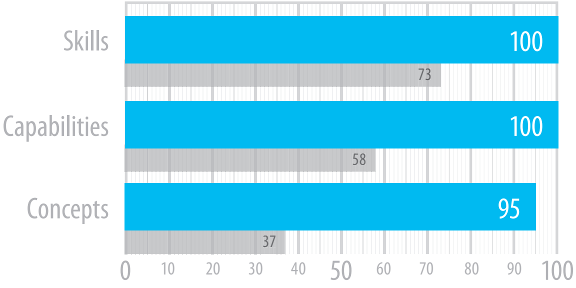
Graduation Requirements

Illinois does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation. Some districts count Computer Science as a Math credit for their students.

Indiana

Implements:





Standards

98% of standards can be found in the business and information technologies standards for the state.

Content Area

Computer Programming (and APCS) is listed with business and information technologies. There are information technology courses such as networking and troubleshooting listed as well.

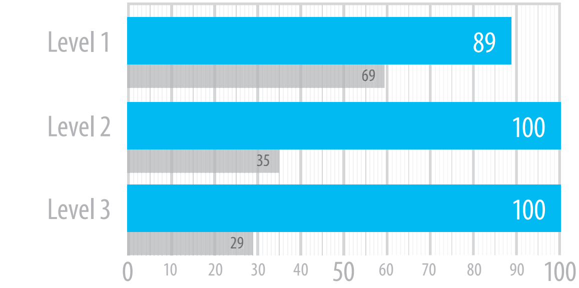
Graduation Requirements

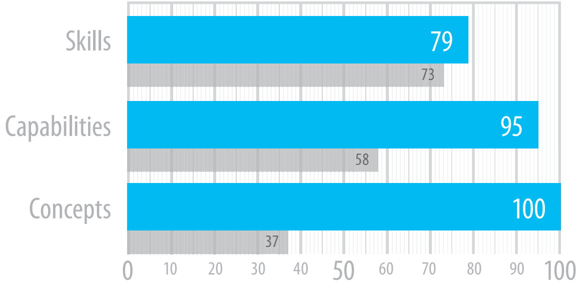
Indiana does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Iowa

Implements:





Standards

98% of standards can be found in the career and technical education standards for the state.

Content Area

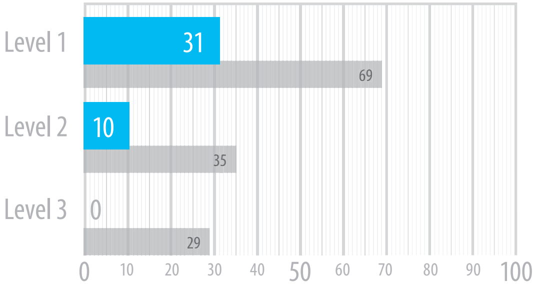
Project Lead the Way is listed under career and technical information and has links to computer programming courses. The technology standards cite the 21st Century skills documents that talk about problem solving and critical thinking.

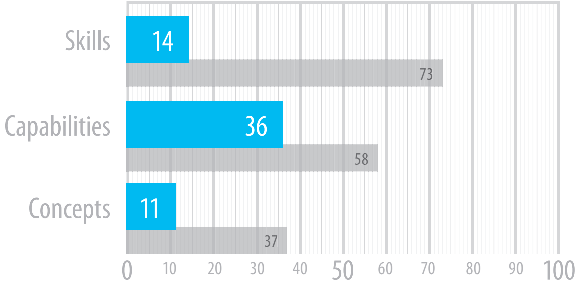
Graduation Requirements

Iowa does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Kansas

Implements:





Standards

100% of standards can be found in the science standards for the state.

Content Area

Kansas has a Science and Technology strand within the general science benchmarks for 8–12. There is also a business and computer technology course which is more of a business applications course than a computer science course.

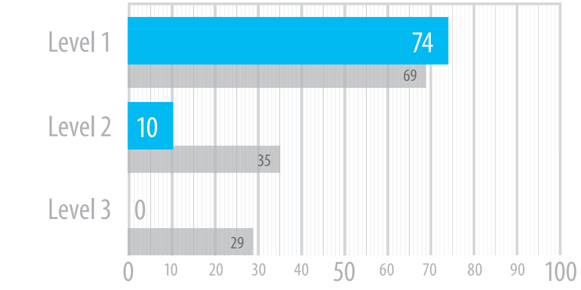
Graduation Requirements

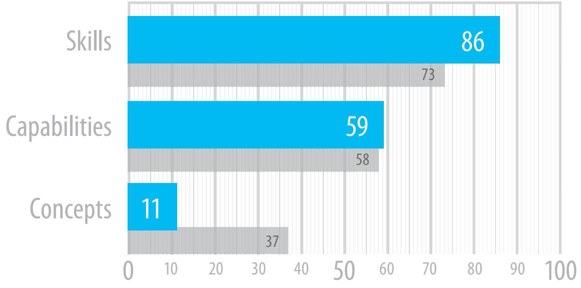
Kansas does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Kentucky

Implements:





Standards

96% of standards are found in the technology standards for the state.

Content Area

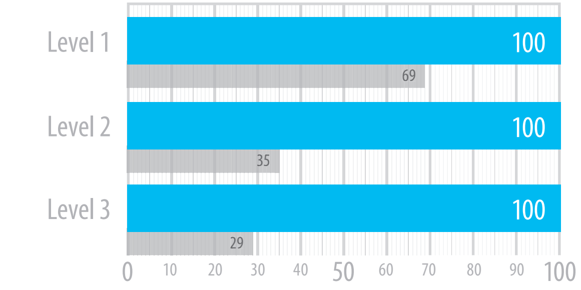
The technology standards for the state are about the application of technology and its support of other scientific and educational goals. There is no mention of programming or creating and modifying technology.

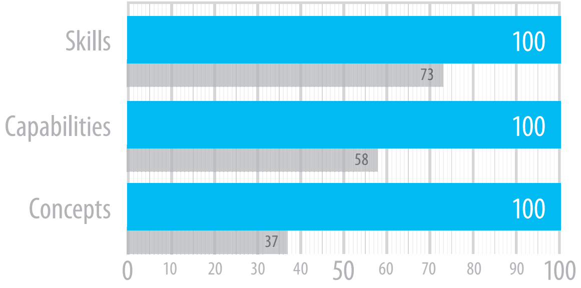
Graduation Requirements

Kentucky has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Louisiana

Implements:





Standards

94% of standards are located in the technology standards for the state.

Content Area

The technology standards for the state are based on the NETS standards. The technology standards document states that "these standards and the associated performance indicators are to be integrated in all aspects of the curriculum and not taught in isolation." There are, however, also technology education courses listed in the 1999 course titles and description document for Computer Education.

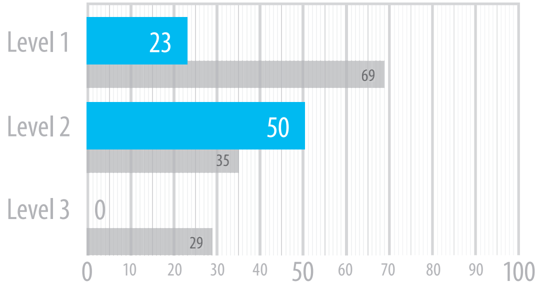
Graduation Requirements

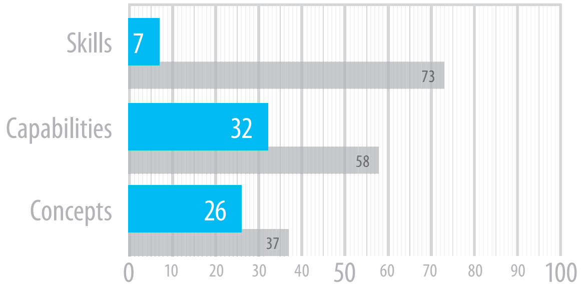
Louisiana does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation. Students have the opportunity to designate Computer Science as an Area of Concentration for their diploma.

🟦 State Results ⬜ National Average

Maine

Implements:





Standards

100% of all standards are in the science standards.

Content Area

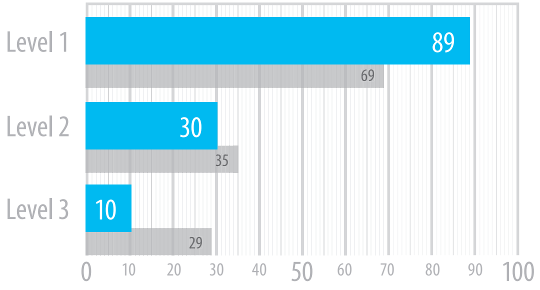
The science standards reference the 21st Century Skills framework as a crucial part of the curriculum. Technology and the process by which technology is created are emphasized as a part of the science standards.

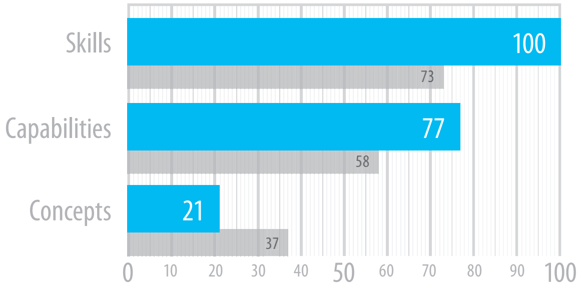
Graduation Requirements

Maine does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Maryland

Implements:





Standards

100% of all standards are in the technology standards for the state.

Content Area

Maryland has a voluntary technology education state curriculum with good technology alignment to the NETS standards. It is unknown how widely used those voluntary standards are. Maryland also has a set of educational technology standards that both overlap and expand upon the literacy standards.

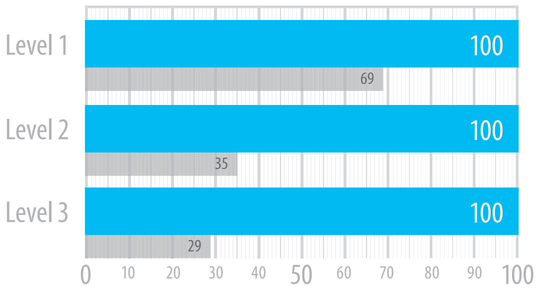
Graduation Requirements

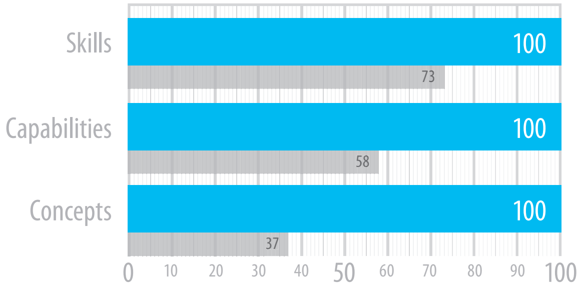
Maryland has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Massachusetts

Implements:





Standards

100% of all standards are found in the science and technology standards for the state.

Content Area

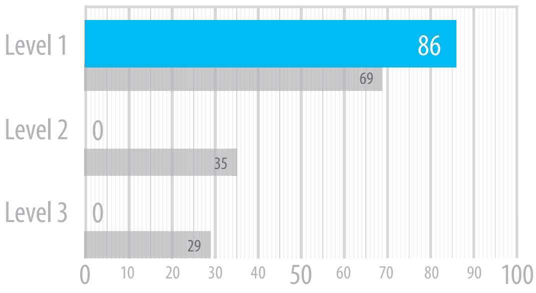
Massachusetts has excellent science standards that include a year of technology and engineering as a part of the high school curriculum. There are also several courses in the career and vo-tech education sector that include programming, robotics, etc. The more general literacy standards include integration scenarios for a variety of disciplines.

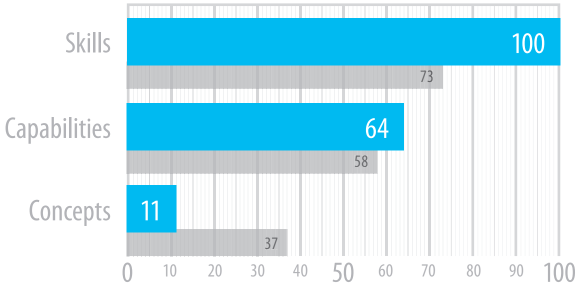
Graduation Requirements

Massachusetts has individual districts determine the graduation requirements. Individual districts also determine what kind of credit is awarded for Computer Science classes. Some districts require technology literacy as a graduation requirement for their students.

Michigan

Implements:





Standards

93% of standards can be found in the technology standards for the state.

Content Area

From the state DOE web page "To prepare students for future work sites, we must begin integrating technology into every aspect of education." The standards documents encourage integration into all disciplines, however the regular subject area standards are devoid of any technology standards.

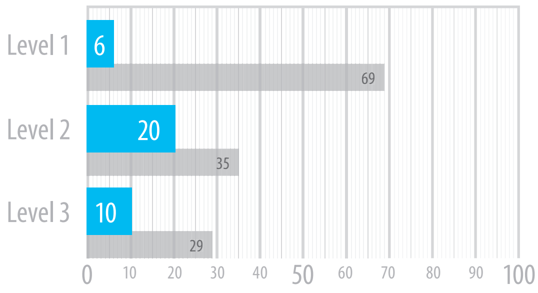
Graduation Requirements

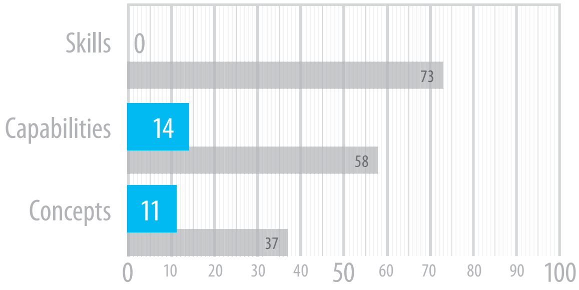
Michigan does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Minnesota

Implements:





Standards

There is only one matching standard for the state and it is in the math standards.

Content Area

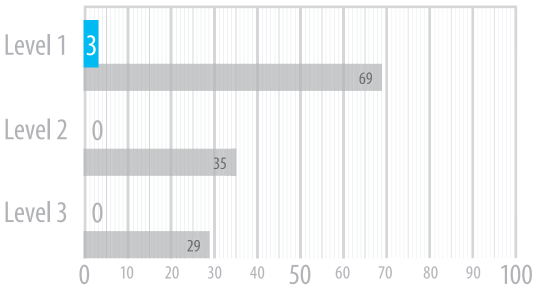
Minnesota has no standards listed.

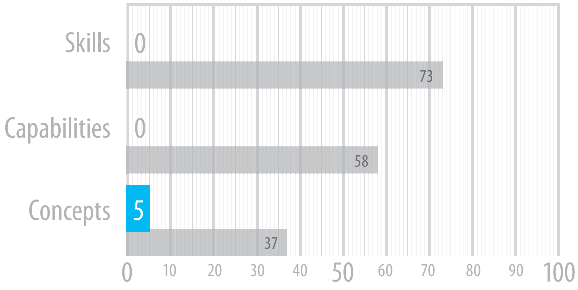
Graduation Requirements

Minnesota does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Mississippi

Implements:





Standards

Mississippi has only one standard listed and it is in the math standards.

Content Area

There are no technology standards listed.

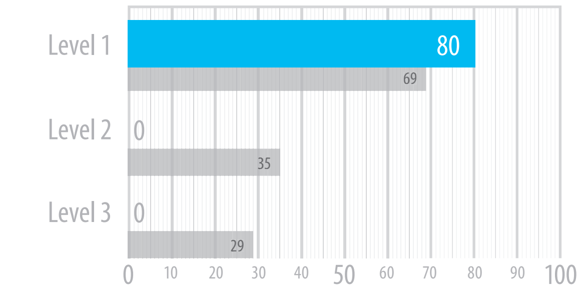
Graduation Requirements

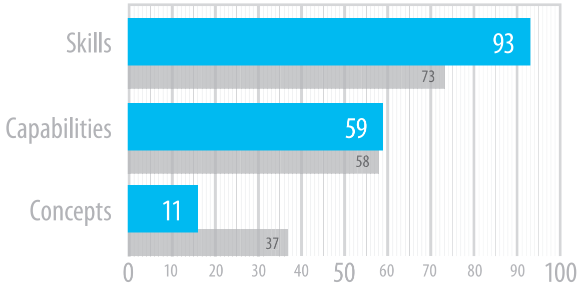
Mississippi has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Missouri

Implements:





Standards

100% of all standards are in the technology standards for the state.

Content Area

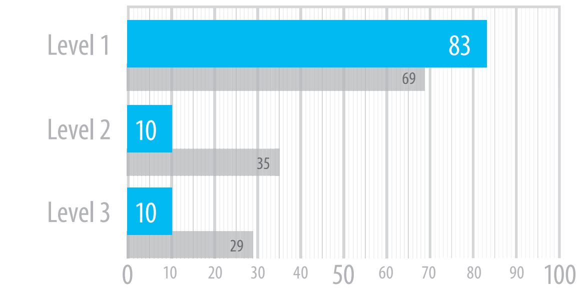
The technology standards are listed in the Division of Career Education, and are very focused on the application of technology.

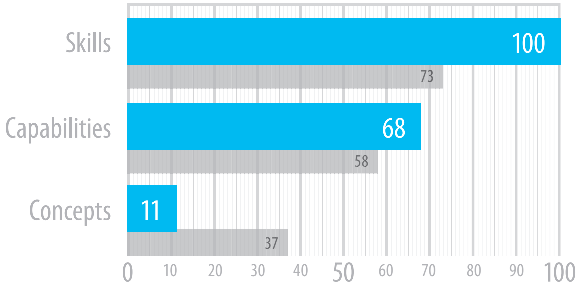
Graduation Requirements

Missouri does not have a technology literacy requirement for graduation. Computer Science counts as a Math credit for graduation.

Montana

Implements:





Standards

100% of all standards are in the technology standards for the state.

Content Area

From the Montana Standards on Technology "education needs to play an increasing role in empowering learners to be technologically literate and to integrate digital tools into their lives." The standards have been recently updated (June 2008) but are primarily application based.

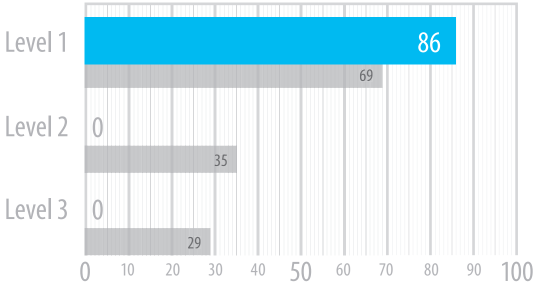
Graduation Requirements

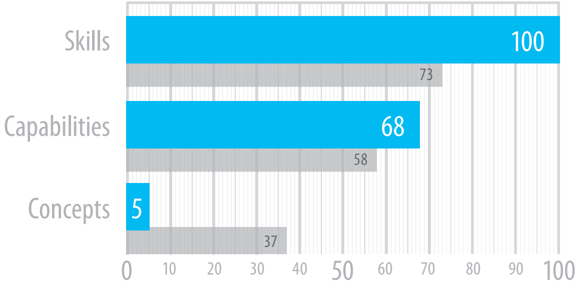
Montana does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ▨ National Average

Nebraska

Implements:





Standards

97% of standards are in the technology standards for the state.

Content Area

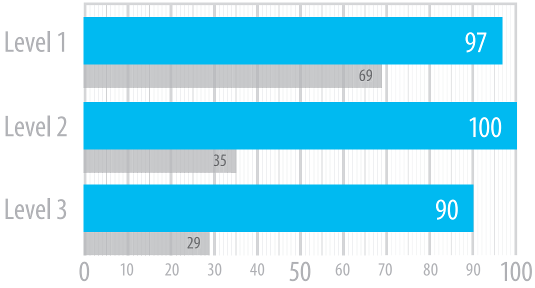
The technology standards are based on the NETS and are very heavily application based.

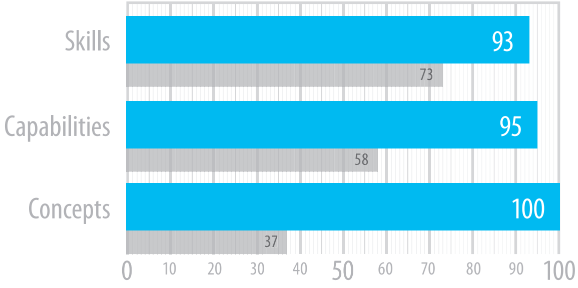
Graduation Requirements

Nebraska has individual districts determine the graduation requirements. Individual districts also determine what kind of credit is awarded for computer science classes. Some districts require technology literacy as a graduation requirement for their students.

Nevada

Implements:





Standards

75% of standards are found in the technology standards for the state.

Content Area

The technology standards are brief with just a few paragraphs about integration into other disciplines. The science and math standards do include technology as a part of the problem-solving process. From the standards website "Society needs individuals who are literate in technology and possess the skills that enable them to participate in a high-performance work force that adapts readily to constantly changing technology."

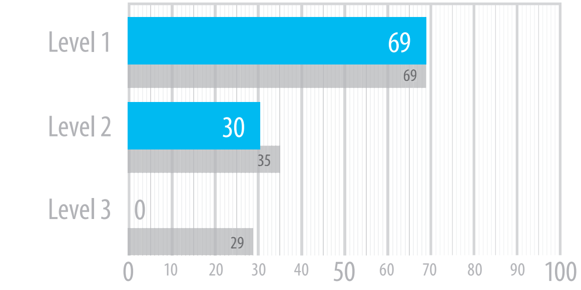
Graduation Requirements

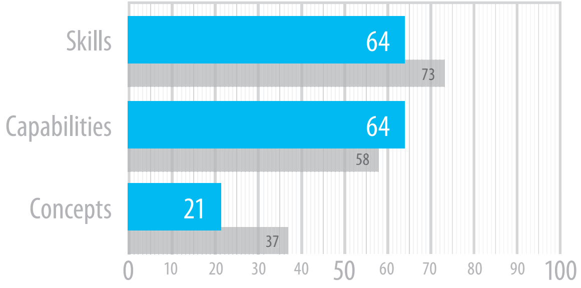
Nevada has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

New Hampshire

Implements:





Standards

100% of all standards are found in the Industrial Arts curriculum.

Content Area

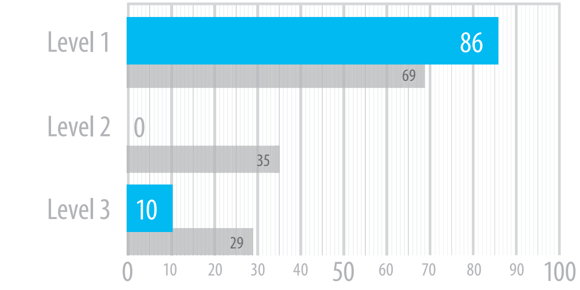
There are no technology standards listed with the regular curriculum areas, however there is a computer technology strand embedded in the industrial arts curriculum.

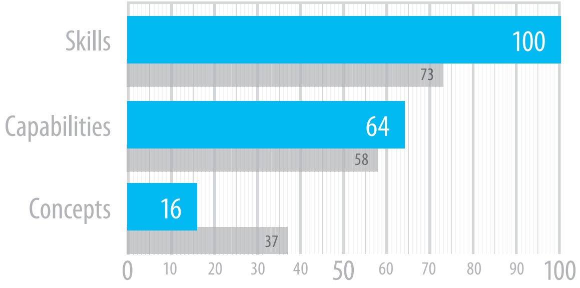
Graduation Requirements

New Hampshire has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

New Jersey

Implements:





Standards

76% of standards can be found in the technology standards for the state.

Content Area

The technology standards state “students should be able to select equipment and tools, apply technology to specific tasks, and maintain and troubleshoot equipment.” The standards were last updated in 2004, however there are indications on the website that they are currently being revisited.

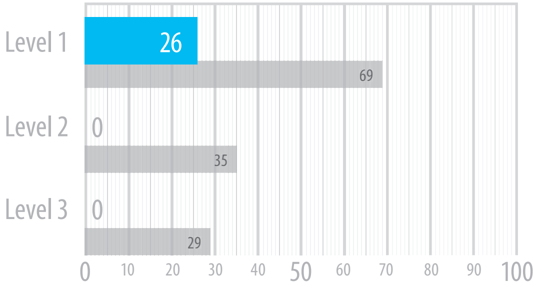
Graduation Requirements

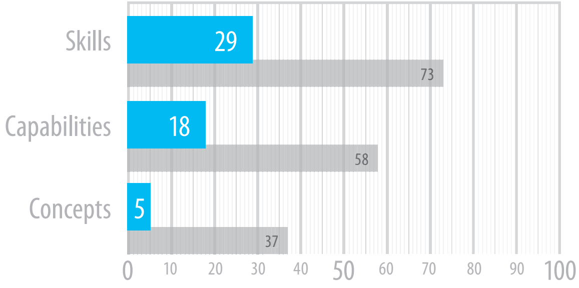
New Jersey does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

New Mexico

Implements:





Standards

78% of standards are located in the science standards for the state.

Content Area

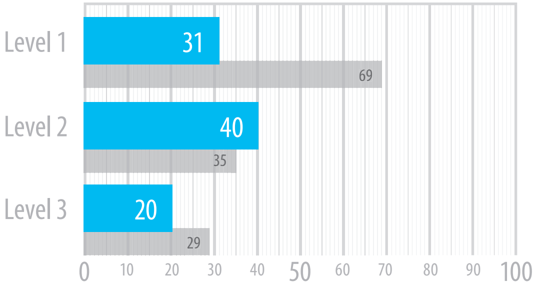
There are no technology standards for the state, however the science standards include a strand on “science and society” that includes a few of the ACM/CSTA standards.

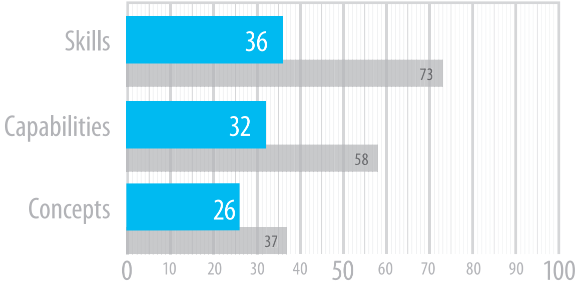
Graduation Requirements

New Mexico does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation

New York

Implements:





Standards

67% of standards can be found in the science standards for the state. The remainder are in the mathematics standards.

Content Area

New York has a Math, Science, and Technology framework that includes some technology standards. While some of these standards are computing in nature, most of them are based in engineering or manufacturing technology.

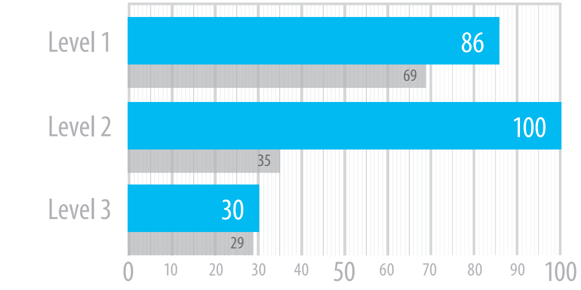
Graduation Requirements

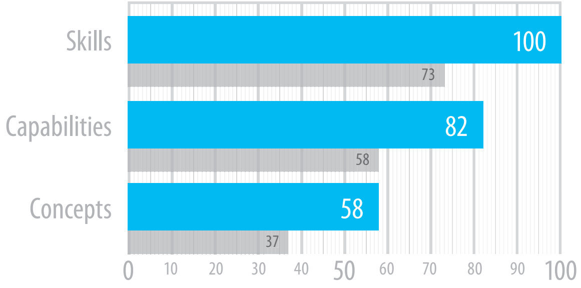
New York has a technology literacy requirement for graduation. Computer Science counts as a Math credit for graduation.

🟦 State Results ⬜ National Average

North Carolina

Implements:





Standards

67% of standards are located in the technology standards for the state.

Content Area

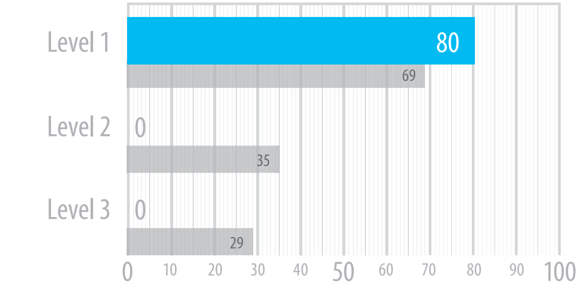
From the Computer Skills section of the NC Standard Course of Study, "The K–12 Computer/Technology Skills Standard Course of Study identifies the essential knowledge and skills that all students need to be active, lifelong learners in a technology intensive environment." The main technology standards are NETS based and indicate subject integration, and there are also programming and information technologies courses listed under the career and technical education sector.

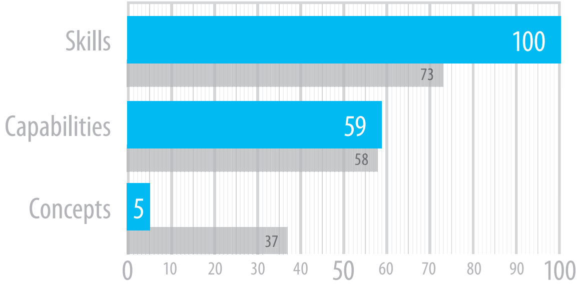
Graduation Requirements

North Carolina has a technology literacy requirement for graduation. Computer Science counts as a Math credit for graduation.

North Dakota

Implements:





Standards

92% of standards can be found in the Library and Technology Literacy standards for the state.

Content Area

The technology standards are mostly listed in the library and technology literacy standards for the state and are very similar to the NETS standards. The science standards for the state also have a strand devoted to science and technology.

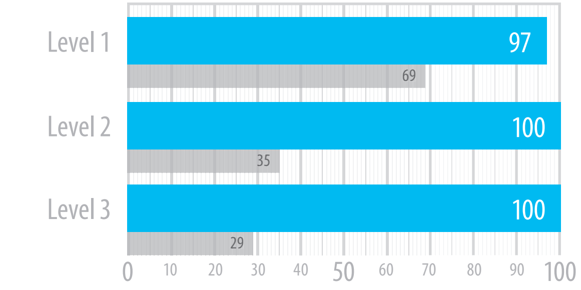
Graduation Requirements

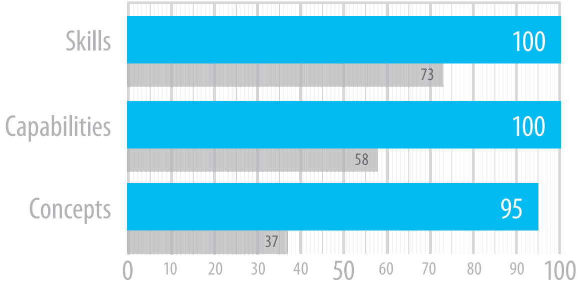
North Dakota does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Ohio

Implements:





Standards

98% of standards can be found in the technology standards for the state.

Content Area

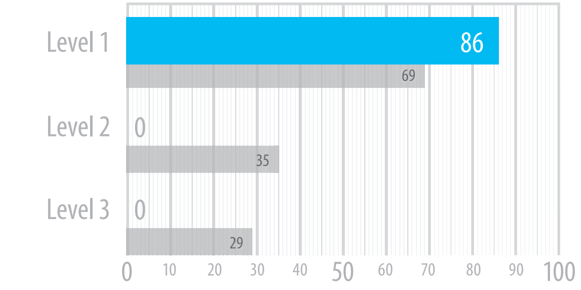
There is a NETS based integration curriculum, as well as an exceptional document in the Career Field Technical Content for Information Technology. The Career document includes courses such as programming and operating systems as potential offerings.

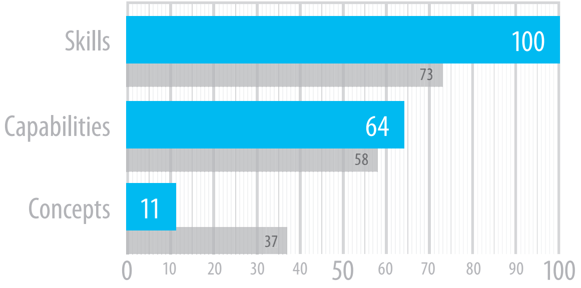
Graduation Requirements

Ohio does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Oklahoma

Implements:





Standards

100% of all standards can be found in the technology standards for the state.

Content Area

From the technology education standards document "Technology Education capitalizes on the individual's potential for reasoning and problem solving, for imagining and creating, and for constructing and expressing through the use of tools and materials related to technology." The standards listed in the document are NETS based, and there are no specific CS courses or standards listed.

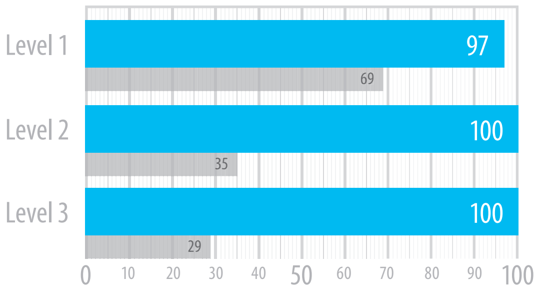
Graduation Requirements

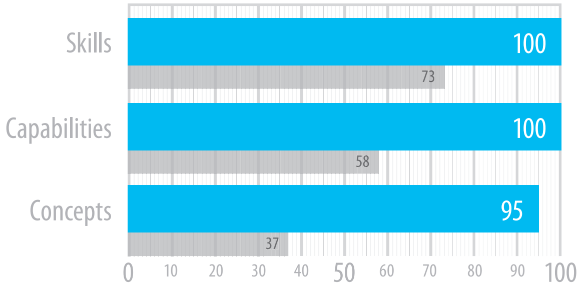
Oklahoma lists a technology literacy requirement as one of many potential credits to fill a mandated graduation requirement. Computer Science counts as a Math credit for graduation.

🟦 State Results ▨ National Average

Oregon

Implements:





Standards

100% of all standards can be found in the technology standards for the state.

Content Area

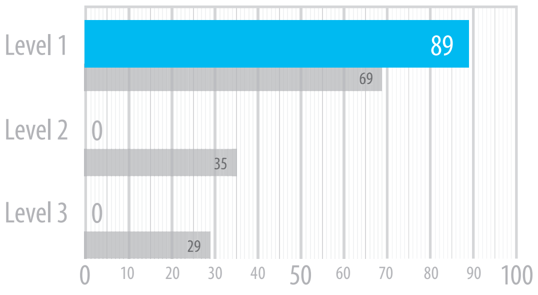
The standards were recently updated and are based upon the NETS standards. The Career Technology Education strands contain potential courses in computer science.

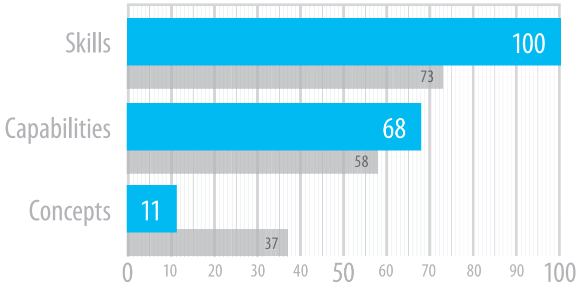
Graduation Requirements

Oregon lists a technology literacy requirement as one of many potential credits to fill a mandated graduation requirement. Computer Science counts as a Math credit for graduation.

Pennsylvania

Implements:





Standards

76% of standards are located in the technology standards for the state.

Content Area

The PA state standards have technology standards embedded in the Math and Science curriculum. The science strands devoted to technology deal with both the use of tools and technology's design and impact on society. There are also separate technology standards for grades 4, 8, and 12 with competencies that match the NETS standards.

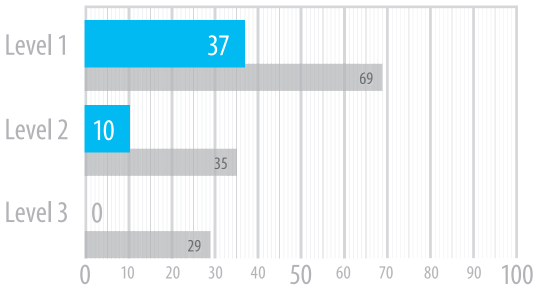
Graduation Requirements

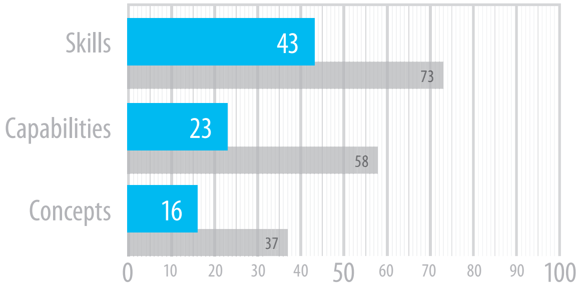
Pennsylvania has individual districts determine the graduation requirements. Individual districts also determine what kind of credit is awarded for computer science classes. In order to graduate, students must pass the PSSA exam; there is no technology or computer science PSSA.

🟦 State Results ▨ National Average

Rhode Island

Implements:





Standards

91% of standards are in the science standards for the state.

Content Area

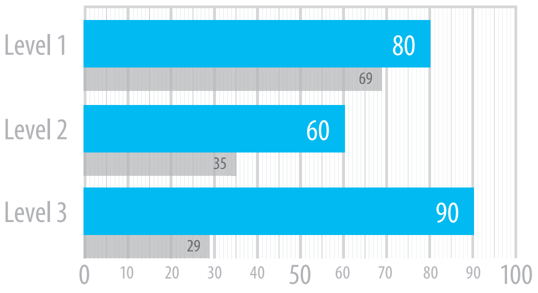
The engineering and technology standards for the state were published in draft form in January of 2007 and still have not been released in final form. The current draft is well written for the topics they cover.

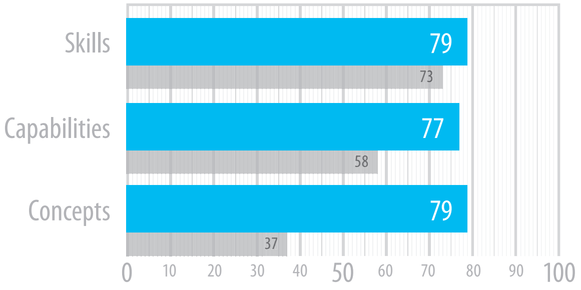
Graduation Requirements

Rhode Island does not have a technology literacy requirement for graduation. Computer Science counts as a Math credit for graduation.

South Carolina

Implements:





Standards

86% of standards can be found in the science standards for the state.

Content Area

The math standards have a brief mention of technology use, and the science standards have an interdisciplinary strand dealing with math and technology integration. There are no general technology standards listed for the state. The technology integration page of the state website has no links on it.

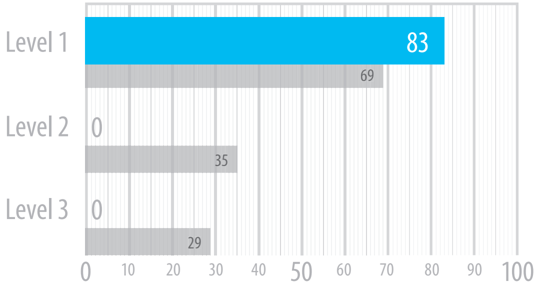
Graduation Requirements

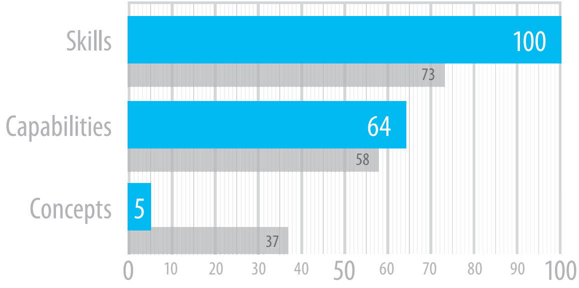
South Carolina has a technology literacy requirement for graduation. Computer Science counts as a Math credit for graduation.

🟦 State Results ▨ National Average

South Dakota

Implements:





Standards

81% of standards can be found in the technology standards for the state.

Content Area

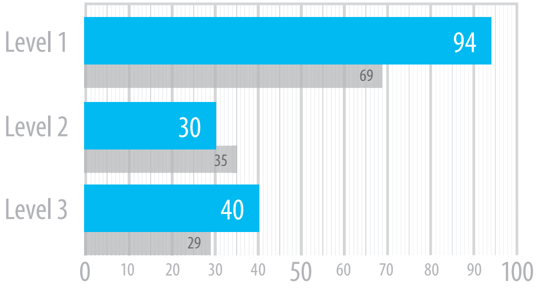
From the Office of Career and Technical Education: "Technology Education is a standards-based program that allows students to investigate and experience the means by which they meet their needs and wants, solve problems and extend their capabilities." There is no integration of technology into the math standards for the state, however the science standards include a technology strand. There is also a separate set of technology content strands matching the NETS.

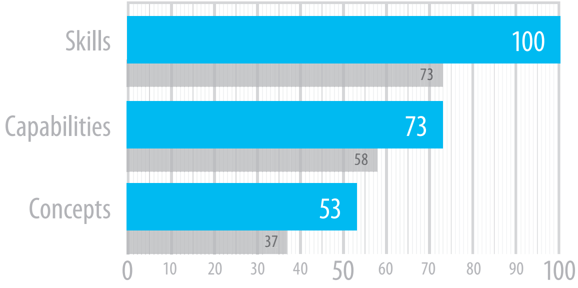
Graduation Requirements

South Dakota lists a technology literacy requirement as one of many potential credits to fill a mandated graduation requirement. Computer Science counts as an elective credit for graduation.

Tennessee

Implements:





Standards

83% of standards can be found in the technology standards for the state.

Content Area

The computer technology website lists various programming classes as well as applications and multimedia classes. Technology use is integrated into the other core content standards as well.

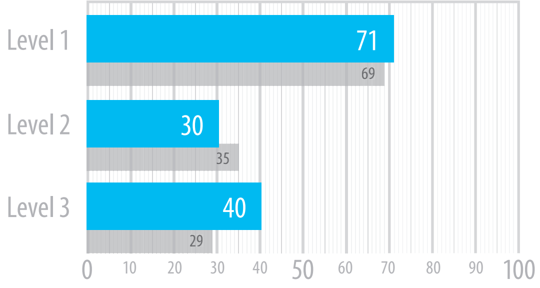
Graduation Requirements

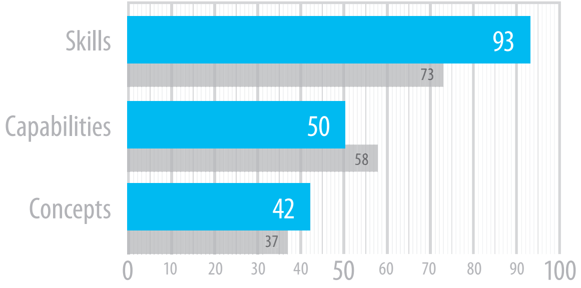
Tennessee has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Texas

Implements:





Standards

77% of standards can be found in the technology standards for the state.

Content Area

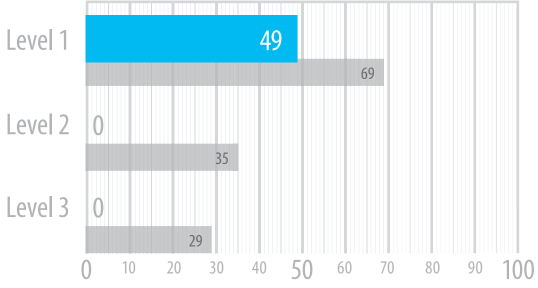
The technology standards themselves are short and ill-defined, a single paragraph gives 2–3 grades worth of information. There is a listing of potential courses including computer science, multimedia, and applications courses.

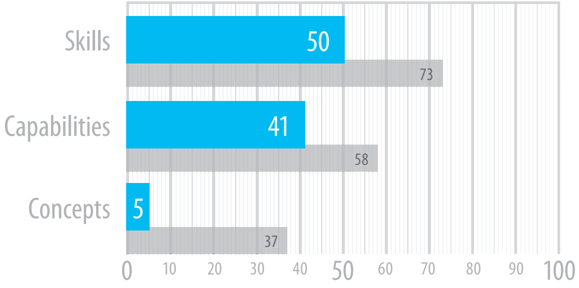
Graduation Requirements

Texas does not have a technology literacy requirement for graduation. Computer Science counts as a Math credit for graduation.

Utah

Implements:





Standards

67% of standards are located in the technology standards for the state.

Content Area

Many of the technology competencies appeared in the 7–12 library media standards for the state. All of the technology standards are about application of technology to other disciplines. There is a skill certificate program for career and technical education as well.

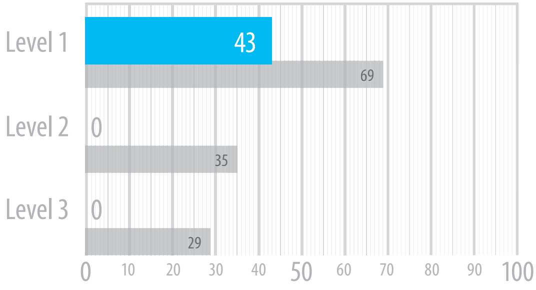
Graduation Requirements

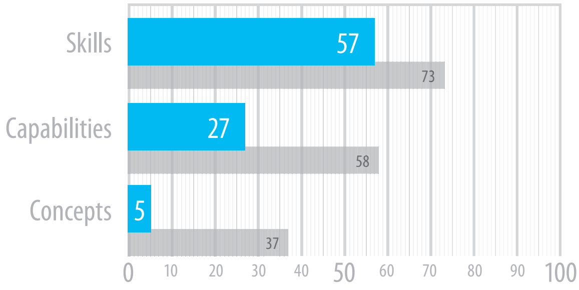
Utah has a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

■ State Results ■ National Average

Vermont

Implements:





Standards

100% of all standards can be found in the science standards for the state.

Content Area

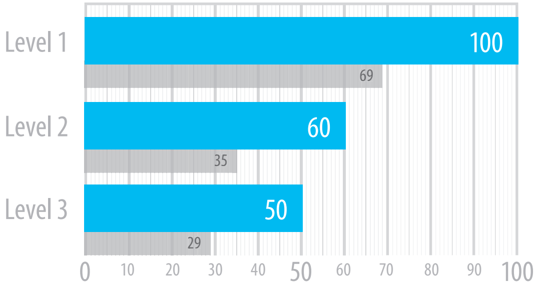
The technology plan is still under development. As recently as April there have been updates to the educational technology website. Most of the information technology plan talks about how to use technology on a more administrative or teacher level and not at a student level, although there is indication that curriculum work is being prepared for this coming summer.

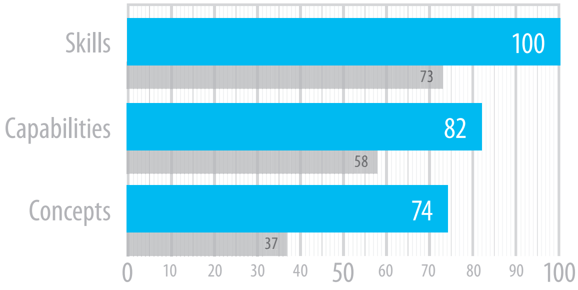
Graduation Requirements

Vermont does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Virginia

Implements:





Standards

87% of standards can be found in the math standards for the state.

Content Area

The math standards explicitly state that reasoning with technology is to be included.

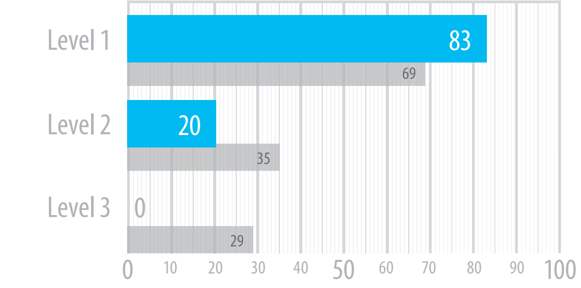
Graduation Requirements

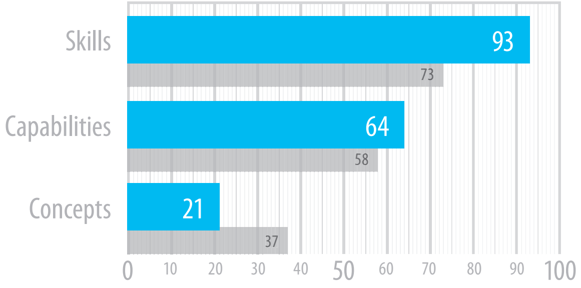
Virginia lists a technology literacy requirement as one of many potential credits to fill a mandated graduation requirement. Computer Science counts as a Math credit for graduation.

🟦 State Results ▨ National Average

Washington

Implements:





Standards

86% of standards can be found in the technology standards for the state.a

Content Area

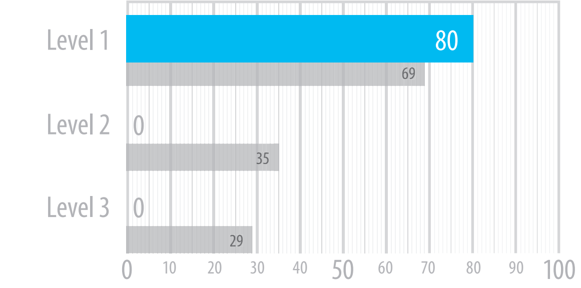
The technology standards define the difference between literacy and fluency and talk about personalizing technology. The technology standards include references to other domains where the standard is applicable.

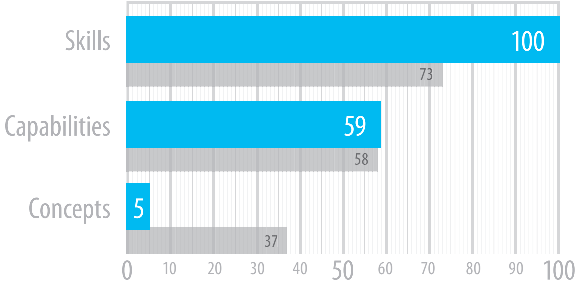
Graduation Requirements

Washington has individual districts determine the graduation requirements. Individual districts also determine what kind of credit is awarded for computer science classes.

West Virginia

Implements:





Standards

77% of standards can be found in the technology standards for the state.

Content Area

The state standards explicitly reference 21st Century Skills for each individual discipline. The technology standards are well defined and include both using and creating technology.

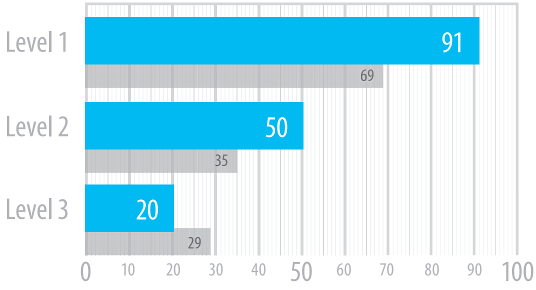
Graduation Requirements

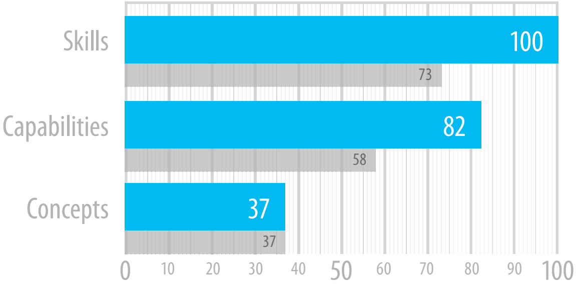
West Virginia does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ⬜ National Average

Wisconsin

Implements:





Standards

The standards are mostly divided between the math and science standards.

Content Area

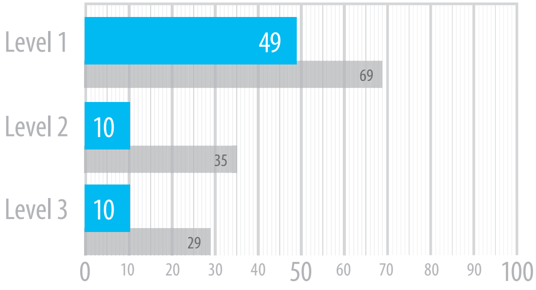
The science standards talk about the compliment of technology to science and its importance in scientific endeavor. The technology standards are focused on troubleshooting, support, and use of multimedia. They do not mention programming.

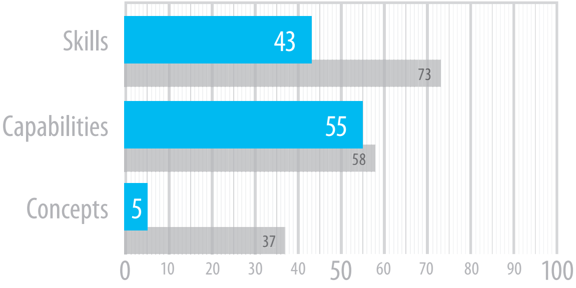
Graduation Requirements

Wisconsin does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

Wyoming

Implements:





Standards

The standards are evenly split between the math, science, and technology standards for the state.

Content Area

Technology is integrated into the basic math and science standards. Career vocational standards also cover technology integration. There are no explicit computer science standards.

Graduation Requirements

Wyoming does not have a technology literacy requirement for graduation. Computer Science counts as an elective credit for graduation.

🟦 State Results ▨ National Average

End Notes

[i] As defined in: *A Model Curriculum for K–12 Computer Science: Final Report of the ACM K–12 Task Force Curriculum Committee,* http://www.csta.acm.org/Curriculum/sub/ACMK12CSModel.html

[ii] *A Nation at Risk: The Imperative for Educational Reform. A Report by the National Commission on Excellence in Education* (Apr. 1983), recommendations section http://www2.ed.gov/pubs/NatAtRisk/recomm.html

[iii] Taken from The College Board public AP data sets available at http://professionals.collegeboard.com/data-reports-research/ap, for more detailed analysis see: http://www.acm.org/public-policy/AP%20Women%20Chart2009.jpg

[iv] Taken from The College Board public AP data sets available at http://professionals.collegeboard.com/data-reports-research/ap,

[v] Margolis, J., Estrella, R., Goode, J., Jellison Holme, J., Nao, K., *Stuck in the Shallow End*. MIT Press, Cambridge, MA, 2008.

[vi] Armoni, M., Ericson, B., Gal-Ezer, J., Seehorn, D., Stephenson, C., and Trees, F. *Ensuring Exemplary Teaching in an Essential Discipline.* ACM, New York, NY, 2008. http://www.csta.acm.org/ComputerScienceTeacherCertification/sub/CertificationStudyReport.html]

[vii] The outline for the course and supporting materials can be found at http://csprinciples.org/index.php, last accessed at Sept.13, 2010

[viii] Achieve.org and Dana Center. *Mathematics Benchmarks, Grades K–12, Examples of Alternative Capstone Courses*; http://www.utdanacenter.org/k12mathbenchmarks/resources/capstonecourses.php

[viv] See details on the Race to the Top fund, including guidance on state applications at http://www2.ed.gov/programs/racetothetop/index.html, last accessed at Sept.13, 2010

[x] A Blueprint for Reform The Reauthorization of the Elementary and Secondary Education Act.The Department of Education (Mar. 2010); http://www2.ed.gov/policy/elsec/leg/blueprint/blueprint.pdf

[xi] Margolis, J., Estrella, R., Goode, J., Jellison Holme, J., Nao, K. *Stuck in the Shallow End.* MIT Press, Cambridge, MA, 2008.

[xii] Taken from The College Board public AP data sets available at http://professionals.collegeboard.com/data-reports-research/ap, for more detailed analysis see: http://www.acm.org/public-policy/2009%20Minority%20Participation%20AP.jpg

[xiii] Ashcraft, C., Breitzman, A., Who Events It? An Analysis of Women's Participation in Information Technology Patenting.

[xiv] Armoni, M., Ericson, B., Gal-Ezer, J., Seehorn, D., Stephenson, C., and Trees, F. *Ensuring Exemplary Teaching in an Essential Discipline*. ACM, New York, NY, 2008. http://www.csta.acm.org/ComputerScienceTeacherCertification/sub/CertificationStudyReport.html

[xv] *Common Core Standards Initiative*. Council of Chief State School Officers and the National Governors Association; http://corestandards.org/

[xvi] A Framework for Science Education Preliminary Public Draft, National Research Council; http://www7.nationalacademies.org/bose/Standards_Framework_Public_Draft_Cover_Letter.html

[xvii] *A Model Curriculum for K–12 Computer Science: Final Report of the ACM K–12 Task Force Curriculum Committee*; http://www.csta.acm.org/Curriculum/sub/ACMK12CSModel.html

[xviii] Gal-Ezer, J., Vilner, T., and Zur, E. *Characteristics of students who failed (or succeeded) the introductory CS course*. Paper presented at the FIEE 2003 Conference, Boulder, CO. Retrieved July 26, 2010 from http://fie-conference.org/fie2003/index.htm

[xix] Sabin, M., Higgs, B., Riabov, V., and Moriera, A. Designing a pre-college computing course. *Journal of Computing In Small Colleges 20*, 5 (2005), 123-132.

[xx] Taylor, H.G., and Mountfield, L.C. The effect of high school computer science, gender, and work on success in college computer science. *ACM SIGCSE Bulletin 21*, 1 (1989), 195-198.

[xxi] Roberts, E. and Halopoff, G. *Computer Science Teachers Association analysis of high school survey data*. (2005) Retrieved July 27, 2010 from http://csta.acm.org/Research/sub/CSTAResearch.html

[xxii] Khoury, G. *Computer science state certification requirements CSTA Certification Committee report*. (2007) Retrieved July 27, 2010 from http://csta.acm.org/ComputerScienceTeacherCertification/sub/CertificationStudyReport.html

[xxiii] *Common Core Standards Initiative*. Council of Chief State School Officers and the National Governors Association; http://corestandards.org/

Association for
Computing Machinery
2 Penn Plaza, Suite 701
New York, NY 10121-0701

T: (212) 626 0686
F: (212) 869 0481
E: acmmediasales@acm.org

www.acm.org

acm Association for Computing Machinery

csta Computer Science Teachers Association

Globe photograph by Nick Aldwin