High School Computer Science Teacher Questionnaire

High School Computer Science Teacher Questionnaire Tables

2018 NSSME+
High School Computer Science Teacher Questionnaire

Teacher Background and Opinions

1. How many years have you taught prior to this school year: [Enter each response as a whole number (for example: 15).]

a.	any subject at the K–12 level?	
b.	computer science at the K–12 level?	
c.	at this school, any subject?	

2. At what grade levels do you currently teach computer science? [Select all that apply.]

□	K–5
□	6–8
□	9–12
□	I do not currently teach computer science. *[Teacher ineligible, exit survey]*

3. Omitted – Used only for survey routing.

4. In a typical week, how many different computer science classes (sections) are you currently teaching?
   - If you meet with the *same class of students* multiple times per week, count that class only once.
   - If you teach the *same computer science course* to multiple classes of students, count each class separately.

1	2	3	4	5	6	7	8	9	10
○	○	○	○	○	○	○	○	○	○

5. For each computer science class you currently teach, select the course type and enter the number of students enrolled. Enter the classes in the order that you teach them. For teachers on an alternating day block schedule, please order your classes starting with the first class you teach this week. [Select one course type on each row and enter the number of students as a whole number (for example: 25).]

GRADES 9–12 COURSE TYPE	EXAMPLE COURSES
Computer technology courses that do not include programming	Computer literacy; Keyboarding; Media technology (digital video/audio, multimedia presentations, digital arts); Desktop publishing; Computer applications (word processing, spreadsheets, slide presentations); Computer repair and computer networking; Web design; Computer-aided design (architectural drawing, fashion design)
Introductory high school computer science courses that include programming	Computer Science Discoveries such as code.org; Exploring computer science; Computer Science Essentials such as PLTW; Introductory Programming; IB Computer Science Standard Level
Computer science courses that might qualify for college credit	AP Computer Science A; AP Computer Science Principles; IB Computer Science Higher Level
Specialized/elective computer science courses with programming as a prerequisite	Advanced Computer science electives such as Robotics; Game or mobile app development; or other advanced computer science elective with programming as a prerequisite

CLASS	COURSE TYPE	NUMBER OF STUDENTS ENROLLED
Your 1st computer science class:		
Your 2nd computer science class:		
…		
Your 10th computer science class:		

COURSE TYPE LIST	
1	Computer technology courses that do not include programming
2	Introductory high school computer science courses that include programming
3	Computer science courses that might qualify for college credit
4	Specialized/elective computer science courses with programming as a prerequisite

6. Later in this questionnaire, we will ask you questions about your *[[xth]]* computer science class, which you indicated was *[[course type indicated in Q5]]*. What is your school's title for this course? _______________

7. Have you been awarded one or more bachelor's and/or graduate degrees in the following fields? (With regard to bachelor's degrees, count only areas in which you majored. Do not include endorsements or certificates.) [Select one on each row.]

		YES	NO
a.	Business	○	○
b.	Computer science	○	○
c.	Education (general or subject specific such as computer science education)	○	○
d.	Information science	○	○
e.	Mathematics	○	○
f.	Natural sciences (for example: Biology, Chemistry, Physics, Earth Sciences)	○	○
g.	Computer engineering	○	○
h.	Electrical engineering	○	○
i.	Other engineering	○	○
j.	Other, please specify.____________	○	○

8. *[Presented only to teachers that selected "Yes" for Q7c]*
What type of education degree do you have? (With regard to bachelor's degrees, count only areas in which you majored.) [Select all that apply.]

□	Computer Science Education
□	Elementary Education
□	Mathematics Education
□	Science Education
□	Other education, please specify. ____________

9. Did you complete one or more computer science courses in each of the following areas at the undergraduate or graduate level? [Select one on each row.]

		YES	NO
a.	Introduction to computer science	○	○
b.	Introduction to programming	○	○
c.	Algorithms (for example: sorting; search trees, heaps, and hashing; divide-and-conquer)	○	○
d.	Artificial intelligence (for example: machine learning, robotics, computer vision)	○	○
e.	Computer graphics (for example: ray tracing, the graphics pipeline, transformations, texture mapping)	○	○
f.	Computer networks (for example: application layer protocols, Internet protocols, network interfaces)	○	○
g.	Database systems (for example: the relational model, relational algebra, SQL)	○	○
h.	Human-computer interaction (for example: human information processing subsystems; libraries of standard graphical user interface objects; methodologies to measure the usability of software)	○	○
i.	Operating systems/computer systems	○	○
j.	Software design/engineering	○	○
k.	Other upper division computer science	○	○

10. Did you complete the following mathematics courses at the undergraduate or graduate level? [Select one on each row.]

		YES	NO
a.	Linear algebra	○	○
b.	Probability	○	○
c.	Statistics	○	○
d.	Number theory (for example: divisibility theorems, properties of prime numbers)	○	○
e.	Discrete mathematics (for example: combinatorics, graph theory, game theory)	○	○

11. Did you complete courses in each of the following areas at the undergraduate or graduate level? [Select one on each row.]

		YES	NO
a.	Computer engineering	○	○
b.	Electrical/Electronics engineering	○	○
c.	Other types of engineering courses	○	○

12. Which of the following best describes the program you completed to earn your teaching credential (sometimes called certification or license)?

○	An undergraduate program leading to a bachelor's degree and a teaching credential
○	A post-baccalaureate credentialing program (no master's degree awarded)
○	A master's program that also led to a teaching credential
○	I have not completed a program to earn a teaching credential. ***[Skip to Q14]***

13. In which of the following areas are you certified (have a credential or endorsement) to teach at the high school level? [Select all that apply.]

□	Business
□	Computer science
□	Engineering
□	Mathematics
□	Science (any area)
□	Other

14. After completing your undergraduate degree and prior to becoming a teacher, did you have a full-time job that included computer programming or computer/software engineering?

○	Yes
○	No

Professional Development

The questions in this section ask about your participation in professional development focused on computer science or computer science teaching. When answering these questions, please include:

- face-to-face and/or online courses;
- professional meetings/conferences;
- workshops;
- professional learning communities/lesson studies/teacher study groups; and
- coaching and mentoring.

Do not include:

- courses you took prior to becoming a teacher; and
- time spent providing professional development (including coaching and mentoring) for other teachers.

15. When did you **last participate** in professional development focused on computer science or computer science teaching?

○	In the last 12 months	
○	1–3 years ago	*[Skip to Q20]*
○	4–6 years ago	
○	7–10 years ago	
○	More than 10 years ago	
○	Never	

16. **In the last 3 years,** which of the following types of professional development related to computer science or computer science teaching have you had? [Select one on each row.]

		YES	NO
a.	I attended a professional development program/workshop.	○	○
b.	I attended a national, state, or regional computer science teacher association meeting.	○	○
c.	I completed an online course/webinar.	○	○
d.	I participated in a professional learning community/lesson study/teacher study group.	○	○
e.	I received assistance or feedback from a formally designated coach/mentor.	○	○
f.	I took a formal course for college credit.	○	○

17. What is the **total** amount of time you have spent on professional development related to computer science or computer science teaching **in the last 3 years**?

○	Less than 6 hours
○	6–15 hours
○	16–35 hours
○	36–80 hours
○	More than 80 hours

18. Considering all of your computer science-related professional development **in the last 3 years**, to what extent does each of the following describe your experiences? [Select one on each row.]

		NOT AT ALL		SOMEWHAT		TO A GREAT EXTENT
a.	I had opportunities to engage in activities to learn computer science content.	①	②	③	④	⑤
b.	I had opportunities to experience lessons, as my students would, from the textbook/units I use in my classroom.	①	②	③	④	⑤
c.	I had opportunities to examine classroom artifacts (for example: student work samples, e-portfolios, videos of classroom instruction).	①	②	③	④	⑤
d.	I had opportunities to rehearse instructional practices during the professional development (meaning: try out, receive feedback, and reflect on those practices).	①	②	③	④	⑤
e.	I had opportunities to apply what I learned to my classroom and then come back and talk about it as part of the professional development.	①	②	③	④	⑤
f.	I worked closely with other teachers from my school.	①	②	③	④	⑤
g.	I worked closely with other teachers who taught the same grade and/or subject whether or not they were from my school.	①	②	③	④	⑤

19. Thinking about all of your computer science-related professional development **in the last 3 years**, to what extent was each of the following emphasized? [Select one on each row.]

		NOT AT ALL		SOMEWHAT		TO A GREAT EXTENT
a.	Deepening your own computer science content knowledge, including programming	①	②	③	④	⑤
b.	Deepening your understanding of how computer science is done (for example: breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)	①	②	③	④	⑤
c.	Implementing the computer science textbook/online course to be used in your classroom	①	②	③	④	⑤
d.	Learning how to use programming activities that require a computer	①	②	③	④	⑤
e.	Learning about difficulties that students may have with particular computer science ideas and/or practices	①	②	③	④	⑤
f.	Monitoring student understanding during computer science instruction					
g.	Differentiating computer science instruction to meet the needs of diverse learners	①	②	③	④	⑤
h.	Incorporating students' cultural backgrounds into computer science instruction	①	②	③	④	⑤
i.	Learning how to provide computer science instruction that integrates engineering, mathematics, and/or science	①	②	③	④	⑤

Preparedness to Teach Computer Science

20. Within computer science, many teachers feel better prepared to teach some topics than others. How prepared do you feel to teach each of the following topics **at the grade level(s) you teach,** whether or not they are currently included in your teaching responsibilities? [Select one on each row.]

	NOT ADEQUATELY PREPARED	SOMEWHAT PREPARED	FAIRLY WELL PREPARED	VERY WELL PREPARED
a. Computing systems	①	②	③	④
b. Networks and the Internet	①	②	③	④
c. Data and analysis	①	②	③	④
d. Algorithms and programming	①	②	③	④
e. Impacts of computing	①	②	③	④

21. How well prepared do you feel to do each of the following in your computer science instruction? [Select one on each row.]

	NOT ADEQUATELY PREPARED	SOMEWHAT PREPARED	FAIRLY WELL PREPARED	VERY WELL PREPARED
a. Develop students' conceptual understanding of the computer science ideas you teach	①	②	③	④
b. Develop students' abilities to do computer science (for example: breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)	①	②	③	④
c. Develop students' awareness of STEM careers	①	②	③	④
d. Provide computer science instruction that is based on students' ideas (whether completely correct or not) about the topics you teach	①	②	③	④
e. Use formative assessment to monitor student learning	①	②	③	④
f. Differentiate computer science instruction to meet the needs of diverse learners	①	②	③	④
g. Incorporate students' cultural backgrounds into computer science instruction	①	②	③	④
h. Encourage students' interest in computer science	①	②	③	④
i. Encourage participation of all students in computer science	①	②	③	④

Opinions about Computer Science Instruction

22. Please provide your opinion about each of the following statements. [Select one on each row.]

		STRONGLY DISAGREE	DISAGREE	NO OPINION	AGREE	STRONGLY AGREE
a.	Students learn computer science best in classes with students of similar abilities.	①	②	③	④	⑤
b.	It is better for computer science instruction to focus on ideas in depth, even if that means covering fewer topics.	①	②	③	④	⑤
c.	At the beginning of instruction on a computer science idea, students should be provided with definitions for new vocabulary that will be used.	①	②	③	④	⑤
d.	Most class periods should provide opportunities for students to share their thinking and reasoning.	①	②	③	④	⑤
e.	Hands-on/manipulatives/programming activities should be used primarily to reinforce a computer science idea that the students have already learned.	①	②	③	④	⑤
f.	Teachers should ask students to justify their solutions to a computational problem.	①	②	③	④	⑤
g.	Students learn best when instruction is connected to their everyday lives.	①	②	③	④	⑤
h.	Most class periods should provide opportunities for students to apply computer science ideas to real-world contexts.	①	②	③	④	⑤
i.	Students should learn computer science by doing computer science (for example: breaking problems into smaller parts, considering the needs of a user, creating computational artifacts).	①	②	③	④	⑤

Leadership Experiences

23. **In the last 3 years** have you… [Select one on each row.]

		YES	NO
a.	Served as a lead teacher or department chair?	○	○
b.	Served as a ***formal*** mentor or coach for a computer science teacher? (Do not include supervision of student teachers.)	○	○
c.	Supervised a student teacher in your classroom?	○	○
d.	Served on a school or district/diocese-wide computer science committee (for example: developing curriculum, developing pacing guides, selecting instructional materials)?	○	○
e.	Led or co-led a workshop or professional learning community (for example: teacher study group, lesson study) for other teachers focused on computer science or computer science teaching?	○	○
f.	Taught a computer science lesson for other teachers to observe?	○	○
g.	Observed another teacher's computer science lesson for the purpose of giving him/her feedback?	○	○

Your Computer Science Instruction

The rest of this questionnaire is about your *[[xth]]*computer science class, which you indicated was *[[type indicated in Q5]]* and is titled *[[title provided in Q6]]*.

24. On average, how many minutes per week does this class meet? [Enter your response as a whole number (for example: 300).] __________

25. Enter the number of students for each grade represented in this class. [Enter each response as a whole number (for example: 15).]

9th grade	
10th grade	
11th grade	
12th grade	
Other	

26. For the students in this class, indicate the number of males and females in each of the following categories of race/ethnicity. [Enter each response as a whole number (for example: 15).]

		MALES	FEMALES
a.	American Indian or Alaskan Native		
b.	Asian		
c.	Black or African American		
d.	Hispanic or Latino		
e.	Native Hawaiian or Other Pacific Islander		
f.	White		
g.	Two or more races		

27. Which of the following best describes the prior achievement levels of the students in this class relative to other students in this school?

○	Mostly low achievers
○	Mostly average achievers
○	Mostly high achievers
○	A mixture of levels

28. How much control do you have over each of the following for computer science instruction in this class? [Select one on each row.]

		NO CONTROL		MODERATE CONTROL		STRONG CONTROL
a.	Determining course goals and objectives	①	②	③	④	⑤
b.	Selecting curriculum materials (for example: textbooks/online courses)	①	②	③	④	⑤
c.	Selecting content, topics, and skills to be taught	①	②	③	④	⑤
d.	Selecting programming languages to use	①	②	③	④	⑤
e.	Selecting the sequence in which topics are covered	①	②	③	④	⑤
f.	Determining the amount of instructional time to spend on each topic	①	②	③	④	⑤
g.	Selecting teaching techniques	①	②	③	④	⑤
h.	Determining the amount of homework to be assigned	①	②	③	④	⑤
i.	Choosing criteria for grading student performance	①	②	③	④	⑤

29. Think about your plans for this class for the entire course. By the end of the course, how much emphasis will each of the following student objectives receive? [Select one on each row.]

		NONE	MINIMAL EMPHASIS	MODERATE EMPHASIS	HEAVY EMPHASIS
a.	Learning computer science vocabulary and/or program syntax	①	②	③	④
b.	Understanding computer science concepts	①	②	③	④
c.	Learning how to do computer science (for example: breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)	①	②	③	④
d.	Learning how to develop computational solutions	①	②	③	④
e.	Learning about real-life applications of computer science	①	②	③	④
f.	Increasing students’ interest in computer science	①	②	③	④
g.	Developing students’ confidence that they can successfully pursue careers in computer science	①	②	③	④

30. How often do **you** do each of the following in your computer science instruction in this class? [Select one on each row.]

	NEVER	RARELY (FOR EXAMPLE: A FEW TIMES A YEAR)	SOMETIMES (FOR EXAMPLE: ONCE OR TWICE A MONTH)	OFTEN (FOR EXAMPLE: ONCE OR TWICE A WEEK)	ALL OR ALMOST ALL COMPUTER SCIENCE LESSONS
a. Explain computer science ideas to the whole class	①	②	③	④	⑤
b. Engage the whole class in discussions	①	②	③	④	⑤
c. Have students work in small groups	①	②	③	④	⑤
d. Have students do hands-on/manipulative programming activities that do not require a computer	①	②	③	④	⑤
e. Have students work on programming activities using a computer	①	②	③	④	⑤
f. Use flipped instruction (have students watch lectures/demonstrations outside of class to prepare for in-class activities)	①	②	③	④	⑤
g. Have students read from a textbook/online course in class, either aloud or to themselves	①	②	③	④	⑤
h. Have students explain and justify their method for solving a problem	①	②	③	④	⑤
i. Have students present their solution strategies to the rest of the class	①	②	③	④	⑤
j. Have students compare and contrast different methods for solving a problem	①	②	③	④	⑤
k. Have students write their reflections (for example: in their journals, on exit tickets) in class or for homework	①	②	③	④	⑤
l. Focus on literacy skills (for example: informational reading or writing strategies)	①	②	③	④	⑤

31. How often do you have **students** do each of the following in this class? [Select one on each row.]

		NEVER	RARELY (FOR EXAMPLE: A FEW TIMES A YEAR)	SOMETIMES (FOR EXAMPLE: ONCE OR TWICE A MONTH)	OFTEN (FOR EXAMPLE: ONCE OR TWICE A WEEK)	ALL OR ALMOST ALL COMPUTER SCIENCE LESSONS
a.	Create computational artifacts (for example: programs, simulations, visualizations, digital animations, robotic systems, or apps)	①	②	③	④	⑤
b.	Create a computational artifact designed to be used by someone outside the class or other students	①	②	③	④	⑤
c.	Provide feedback on other students' computational products or designs	①	②	③	④	⑤
d.	Get input on computational products or designs from people with different perspectives (do not include feedback that you give students)	①	②	③	④	⑤
e.	Systematically use test cases to verify program performance and/or identify problems	①	②	③	④	⑤
f.	Identify real-world problems that might be solved computationally	①	②	③	④	⑤
g.	Consider how a program they are creating can be separated into modules/procedures/objects	①	②	③	④	⑤
h.	Identify and adapt existing code to solve a new computational problem	①	②	③	④	⑤
i.	Use computational methods to simulate events or processes (for example: rolling dice, supply and demand)	①	②	③	④	⑤
j.	Analyze datasets using a computer to detect patterns	①	②	③	④	⑤
k.	Write comments within code to document purposes or features	①	②	③	④	⑤
l.	Create instructions for an end-user explaining how to use a computational artifact	①	②	③	④	⑤
m.	Explain computational solution strategies verbally or in writing	①	②	③	④	⑤
n.	Compare and contrast the strengths and limitations of different representations such as flow charts, tables, code, or pictures	①	②	③	④	⑤

32. Which best describes how each of the following devices (if required) is provided for this computer science class? [Select one on each row.]

	NOT REQUIRED FOR THIS CLASS	PROVIDED BY THE SCHOOL, AND STUDENTS ARE NOT ALLOWED TO USE THEIR OWN	PROVIDED BY THE SCHOOL, BUT STUDENTS ARE ALLOWED TO USE THEIR OWN	STUDENTS ARE EXPECTED TO PROVIDE THEIR OWN, BUT THE SCHOOL HAS SOME AVAILABLE FOR USE	STUDENTS ARE REQUIRED TO PROVIDE THEIR OWN
a. Computers (desktops or laptops)	○	○	○	○	○
b. Mobile computing devices (tablets or smartphones)	○	○	○	○	○
c. Data storage devices	○	○	○	○	○

33. Please indicate the availability of each of the following for your computer science instruction in this class. [Select one on each row.]

	ALWAYS AVAILABLE IN YOUR CLASSROOM	AVAILABLE UPON REQUEST	NOT AVAILABLE
a. Probes for collecting data (for example: motion sensors, temperature probes)	○	○	○
b. Projection devices (for example: Smartboard, document camera, LCD projector)	○	○	○
c. Robotics equipment	○	○	○

34. In a typical week, how much time outside of this class are students expected to spend on computer science assignments?

○	None
○	1–15 minutes per week
○	16–30 minutes per week
○	31–60 minutes per week
○	61–90 minutes per week
○	91–120 minutes per week
○	More than 2 hours per week

This next item asks about different types of instructional materials; please read the entire list of materials before answering

35. Thinking about your instruction in this class over the entire year, about how often is instruction based on materials from each of the following sources? [Select one on each row.]

	NEVER	RARELY (FOR EXAMPLE: A FEW TIMES A YEAR)	SOMETIMES (FOR EXAMPLE: ONCE OR TWICE A MONTH)	OFTEN (FOR EXAMPLE: ONCE OR TWICE A WEEK)	ALL OR ALMOST ALL COMPUTER SCIENCE LESSONS
a. Commercially published textbooks (printed or electronic), including the supplementary materials (for example: worksheets) that accompany the textbooks	①	②	③	④	⑤
b. State, county, or district/diocese-developed units or lessons	①	②	③	④	⑤
c. Online units or courses that students work through at their own pace (for example: MOOCs, EdX, IMACS)	①	②	③	④	⑤
d. Lessons or resources from websites that have a subscription fee or per lesson cost (for example: BrainPOP, Discovery Ed, Teachers Pay Teachers)	①	②	③	④	⑤
e. Lessons or resources from websites that are free (for example: Khan Academy, code.org)	①	②	③	④	⑤
f. Units or lessons you created (either by yourself or with others)	①	②	③	④	⑤
g. Units or lessons you collected from any other source (for example: conferences, journals, colleagues, university or museum partners)	①	②	③	④	⑤

36. Does your school/district/diocese designate instructional materials (textbooks, units, or lessons) to be used in this class?

○	Yes
○	No *[Skip to 39]*

37. Which of the following types of instructional materials does your school/district/diocese designate to be used in this class? [Select all that apply.]

□	Commercially published textbooks (printed or electronic), including the supplementary materials (for example: worksheets) that accompany the textbooks
□	State, county, or district/diocese-developed instructional materials
□	Online units or courses that students work through at their own pace (for example: MOOCs, EdX, IMACS)
□	Lessons or resources from websites that have a subscription fee or per lesson cost (for example: BrainPOP, Discovery Ed, Teachers Pay Teachers)
□	Lessons or resources from websites that are free (for example: Khan Academy, code.org)

38. Omitted – Used only for survey routing.

39. ***[Presented only to teachers who selected "Sometimes" "Often" or "All" for Q35a or c]***
[Version for teachers who indicate using a commercial textbook most often] Please indicate the title, author, most recent copyright year, and ISBN code of the commercially published textbook (printed or electronic) used most often by the students in this class.

- The 10- or 13-character ISBN code can be found on the copyright page and/or the back cover of the textbook.
- Do not include the dashes when entering the ISBN.
- Example ISBN:

[Version for teachers who indicate using an online course most often] Please indicate the title and URL of the online units or courses used most often by the students in this class.

Title:	
First Author: [for teachers who indicate using a commercial textbook most often]	
Year: [for teachers who indicate using a commercial textbook most often]	
ISBN: [for teachers who indicate using a commercial textbook most often]	
URL: [for teachers who indicate using an online program most often]	

40. ***[Presented only to teachers who did not select "Never" for Q35d or e]***
Please indicate up to 3 online sources of lessons/activities that you use most frequently in this class. Enter only the host/domain name, for example: www.myfavoriteCSsite.net

URL:	
URL:	
URL:	

41. Please rate how each of the following affects your computer science instruction in this class. [Select one on each row.]

		INHIBITS EFFECTIVE INSTRUCTION		NEUTRAL OR MIXED		PROMOTES EFFECTIVE INSTRUCTION	N/A
a.	Current state standards	①	②	③	④	⑤	○
b.	Textbook selection policies	①	②	③	④	⑤	○
c.	Teacher evaluation policies	①	②	③	④	⑤	○
d.	College entrance requirements	①	②	③	④	⑤	○
e.	Students' prior knowledge and skills	①	②	③	④	⑤	○
f.	Students' motivation, interest, and effort in computer science	①	②	③	④	⑤	○
g.	Parent/guardian expectations and involvement	①	②	③	④	⑤	○
h.	Principal support	①	②	③	④	⑤	○
i.	Amount of time for you to plan, individually and with colleagues	①	②	③	④	⑤	○
j.	Amount of time available for your professional development	①	②	③	④	⑤	○

42. In your opinion, how great a problem is each of the following for your computer science instruction in this class? [Select one on each row.]

		NOT A SIGNIFICANT PROBLEM	SOMEWHAT OF A PROBLEM	SERIOUS PROBLEM
a.	Lack of reliable access to the Internet	①	②	③
b.	Lack of functioning computing devices (for example: desktop computers, laptop computers, tablets, smartphones)	①	②	③
c.	Insufficient power sources for devices (for example: electrical outlets, charging stations)	①	②	③
d.	Lack of support to maintain technology (for example: repair broken devices, install software)	①	②	③
e.	School restrictions on Internet content that is allowed	①	②	③

Your Most Recently Completed Computer Science Unit in this Class

The questions in this section are about the most recently completed computer science unit in this class which you indicated is *[[type indicated in Q5]]* and is titled *[[title provided in Q6]]*.

- Depending on the structure of your class and the instructional materials you use, a unit may range from a few to many class periods.
- Do not be concerned if this unit was not typical of your instruction.

43. Which of the following best describes the content focus of this unit?

○	Computing systems
○	Networks and the Internet
○	Data and analysis
○	Algorithms and programming
○	Impacts of computing

44. *[Presented only to teachers who selected "Sometimes" "Often" or "All" for Q35a or b]* Was this unit based primarily on a commercially published textbook/online course or state, county, or district/diocese-developed materials?

○	Yes
○	No *[Skip to Q47]*

This next set of items is about the textbook or state, county, or district/diocese-developed lessons you used in this unit.

45. Please indicate the extent to which you did each of the following while teaching this unit. [Select one on each row.]

		NOT AT ALL		SOMEWHAT		TO A GREAT EXTENT
a.	I used these materials to guide the structure and content emphasis of the unit.	①	②	③	④	⑤
b.	I picked what is important from these materials and skipped the rest.	①	②	③	④	⑤
c.	I incorporated activities (for example: problems, investigations, readings) from other sources to supplement what these materials were lacking.	①	②	③	④	⑤
d.	I modified activities from these materials.	①	②	③	④	⑤

46. *[Presented only to teachers who did not select "Not at all" for Q45b]* During this unit, when you skipped activities (for example: problems, programming activities, readings) in these materials, how much was each of the following a factor in your decisions? [Select one on each row.]

		NOT A FACTOR	A MINOR FACTOR	A MAJOR FACTOR
a.	The computer science ideas addressed in the activities I skipped are not included in my pacing guide/standards.	①	②	③
b.	I did not have the materials needed to implement the activities I skipped.	①	②	③
c.	I did not have the knowledge needed to implement the activities I skipped.			
d.	The activities I skipped were too difficult for my students.	①	②	③
e.	My students already knew the computer science ideas or were able to learn them without the activities I skipped.	①	②	③
f.	I have different activities for those computer science ideas that work better than the ones I skipped.	①	②	③
g.	I did not have enough instructional time for the activities I skipped.	①	②	③

47. *[Presented only to teachers who did not select “Not at all” for Q45c]* During this unit, when you supplemented these materials with additional activities, how much was each of the following a factor in your decisions? [Select one on each row.]

		NOT A FACTOR	A MINOR FACTOR	A MAJOR FACTOR
a.	My pacing guide indicated that I should use supplemental activities.	①	②	③
b.	Supplemental activities were needed to prepare students for standardized tests.	①	②	③
c.	Supplemental activities were needed to provide students with additional practice.	①	②	③
d.	Supplemental activities were needed so students at different levels of achievement could increase their understanding of the ideas targeted in each activity.	①	②	③
e.	I had additional activities that I liked.	①	②	③

48. *[Presented only to teachers who did not select “Not at all” for Q45d]* During this unit, when you modified activities from these materials, how much was each of the following a factor in your decisions? [Select one on each row.]

		NOT A FACTOR	A MINOR FACTOR	A MAJOR FACTOR
a.	I did not have the necessary materials/supplies for the original activities.	①	②	③
b.	The original activities were too difficult conceptually for my students.	①	②	③
c.	The original activities were too easy conceptually for my students.	①	②	③
d.	I did not have enough instructional time to implement the activities as designed.	①	②	③
e.	The original activities were too structured for my students.	①	②	③
f.	The original activities were not structured enough for my students.	①	②	③

49. How well prepared did you feel to do each of the following as part of your instruction on this particular unit? [Select one on each row.]

		NOT ADEQUATELY PREPARED	SOMEWHAT PREPARED	FAIRLY WELL PREPARED	VERY WELL PREPARED
a.	Anticipate difficulties that students may have with particular computer science ideas and procedures in this unit	①	②	③	④
b.	Find out what students thought or already knew about the key computer science ideas	①	②	③	④
c.	Implement the instructional materials (for example: textbook, online course) to be used during this unit	①	②	③	④
d.	Monitor student understanding during this unit	①	②	③	④
e.	Assess student understanding at the conclusion of this unit	①	②	③	④

Your Most Recent Computer Science Lesson in this Class

The next three questions refer to the most recent computer science lesson in this class, which you indicated is *[[type indicated in Q5]]* and is titled *[[title provided in Q6]]*, even if it included activities and/or interruptions that are not typical (for example: a test, students working on projects, a fire drill). If the lesson spanned multiple days, please answer for the most recent day.

50. How many minutes was that day's computer science lesson? Answer for the entire length of the class period, even if there were interruptions. [Enter your response as a non-zero whole number (for example: 50).] ____________________

51. Of these *[[answer to Q50]]* minutes, how many were spent on the following: [Enter each response as a whole number (for example: 15).]

a.	Non-instructional activities (for example: attendance taking, interruptions)	
b.	Whole class activities (for example: lectures, explanations, discussions)	
c.	Small group work	
d.	Students working individually (for example: reading textbooks, programming, taking a test or quiz)	

52. Which of the following activities took place during that day's computer science lesson? [Select all that apply.]

□	Teacher explaining a computer science idea to the whole class
□	Teacher conducting a demonstration while students watched
□	Whole class discussion
□	Students working in small groups
□	Students completing textbook/worksheet problems
□	Students doing hands-on/manipulative programming activities not using a computer
□	Students working on programming tasks using a computer
□	Students reading about computer science
□	Students writing about computer science (do not include students taking notes)
□	Test or quiz
□	None of the above

Demographic Information

53. Are you:

○	Female
○	Male
○	Other

54. Are you of Hispanic or Latino origin?

○	Yes
○	No

55. What is your race? [Select all that apply.]

□	American Indian or Alaskan Native
□	Asian
□	Black or African American
□	Native Hawaiian or Other Pacific Islander
□	White

56. In what year were you born? [Enter your response as a whole number (for example: 1969).]

____

Thank you!

High School Computer Science Teacher Questionnaire Tables

Table CTQ 1
Number of Years High School Computer Science Teachers Spent Teaching Prior to This School Year

	MEAN NUMBER OF YEARS
Any subject at the K–12 level	12 (0.6)
Computer science at the K–12 level	6 (0.5)
At this school, any subject	8 (0.5)

There is no table for CTQ 2.

There is no table for CTQ 3.

Table CTQ 4
Number of Sections of Computer Science Classes Taught Per Week by High School Teachers

	PERCENT OF TEACHERS
1 Section	26 (3.2)
2 Sections	25 (3.4)
3 Sections	20 (3.4)
4 Sections	10 (2.2)
5 Sections	10 (2.8)
6 Sections	6 (1.4)
7 Sections	1 (0.6)
8 Sections	1 (0.7)
9 Sections	0 ---†
10 Sections	0 (0.2)

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

There is no table for CTQ 5.

There is no table for CTQ 6.

Table CTQ 7
Subjects of High School Computer Science Teachers' Degrees

	PERCENT OF TEACHERS
Business	23 (2.8)
Computer science	19 (3.1)
Education (general or subject specific such as computer science education)	51 (4.1)
Information science	4 (1.7)
Mathematics	27 (3.2)
Natural sciences (e.g., Biology, Chemistry, Physics, Earth Sciences)	10 (2.5)
Computer engineering	1 (0.5)
Electrical engineering	5 (2.0)
Other engineering	5 (1.9)
Other subject	26 (3.7)

Table CTQ 8
High School Computer Science Teachers With Education Degrees

	PERCENT OF TEACHERS
Computer Science Education	4 (2.1)
Elementary Education	4 (1.3)
Mathematics Education	16 (2.6)
Science Education	4 (1.6)
Other Education	29 (3.4)

Table CTQ 9
Computer Science College Courses Completed by High School Computer Science Teachers

	PERCENT OF TEACHERS
Introduction to computer science	76 (3.0)
Introduction to programming	80 (2.8)
Algorithms (e.g., sorting; search trees, heaps, and hashing; divide-and-conquer)	50 (3.8)
Artificial intelligence (e.g., machine learning, robotics, computer vision)	14 (2.7)
Computer graphics (e.g., ray tracing, the graphics pipeline, transformations, texture mapping)	22 (3.6)
Computer networks (e.g., application layer protocols, Internet protocols, network interfaces)	32 (3.7)
Database systems (e.g., the relational model, relational algebra, SQL)	38 (3.7)
Human-computer interaction (e.g., human information processing subsystems; libraries of standard graphical user interface objects; methodologies to measure the usability of software)	17 (3.2)
Operating systems/Computer systems	45 (3.5)
Software design/engineering	35 (3.1)
Other upper division computer science	39 (3.9)

Table CTQ 10
Mathematics College Courses
Completed by High School Computer Science Teachers

	PERCENT OF TEACHERS
Linear algebra	72 (3.0)
Probability	59 (3.3)
Statistics	84 (2.7)
Number theory (e.g., divisibility theorems, properties of prime numbers)	44 (3.6)
Discrete mathematics (e.g., combinatorics, graph theory, game theory)	44 (4.1)

Table CTQ 11
Engineering College Courses
Completed by High School Computer Science Teachers

	PERCENT OF TEACHERS
Computer engineering	19 (2.9)
Electrical/Electronics engineering	19 (3.3)
Other types of engineering courses	23 (3.6)

Table CTQ 12
High School Computer Science Teachers' Paths to Certification

	PERCENT OF TEACHERS
An undergraduate program leading to a bachelor's degree and a teaching credential	38 (3.7)
A post-baccalaureate credentialing program (no master's degree awarded)	24 (3.2)
A master's program that also led to a teaching credential	22 (2.8)
I have not completed a program to earn a teaching credential.	16 (2.7)

Table CTQ 13
High School Computer Science Teachers' Areas of Certification

	PERCENT OF TEACHERS
Business	28 (2.4)
Computer science	44 (3.6)
Engineering	10 (2.4)
Mathematics	34 (3.4)
Science (any area)	9 (2.3)
Other	23 (3.0)

Table CTQ 14
High School Computer Science Teachers With Full-Time Job Experience in Computer Programming or Computer/Software Engineering Prior to Teaching

	PERCENT OF TEACHERS
Full-time job experience in their designated field prior to teaching	35 (4.3)

Table CTQ 15
High School Computer Science Teachers' Most Recent Participation in Computer Science-Focused Professional Development

	PERCENT OF TEACHERS
In the last 12 months	64 (3.8)
1–3 years ago	18 (2.7)
4–6 years ago	4 (1.2)
7–10 years ago	2 (1.4)
More than 10 years ago	1 (0.6)
Never	11 (2.7)

Table CTQ 16
High School Computer Science Teachers Participating in Various Computer Science-Focused Professional Development Activities in the Last Three Years

	PERCENT OF TEACHERS†
I attended a professional development program/workshop.	88 (2.4)
I attended a national, state, or regional computer science teacher association meeting.	35 (3.9)
I completed an online course/webinar.	59 (4.7)
I participated in a professional learning community/lesson study/teacher study group.	62 (3.8)
I received assistance or feedback from a formally designated coach/mentor.	29 (3.7)
I took a formal course for college credit.	20 (3.1)

† Includes only high school computer science teachers indicating in Q15 that they participated in computer science-focused professional development in the last three years.

Table CTQ 17
Time Spent by High School Computer Science Teachers on Computer Science-Focused Professional Development in the Last Three Years

	PERCENT OF TEACHERS†
Less than 6 hours	4 (1.3)
6–15 hours	10 (2.4)
16–35 hours	20 (2.6)
36–80 hours	29 (3.5)
More than 80 hours	36 (3.7)

† Includes only high school computer science teachers indicating in Q15 that they participated in computer science-focused professional development in the last three years.

Table CTQ 18

High School Computer Science Teachers' Descriptions of Computer Science-Focused Professional Development in the Last Three Years

	PERCENT OF TEACHERS†				
	NOT AT ALL		SOMEWHAT		TO A GREAT EXTENT
	1	2	3	4	5
I had opportunities to engage in activities to learn computer science content.	3 (2.0)	2 (0.9)	18 (2.6)	29 (3.5)	47 (3.7)
I had opportunities to experience lessons, as my students would, from the textbook/units I use in my classroom.	9 (2.7)	9 (2.0)	20 (3.0)	19 (2.9)	43 (4.1)
I had opportunities to examine classroom artifacts (e.g., student work samples, e-portfolios, videos of classroom instruction).	14 (2.4)	12 (2.7)	28 (4.4)	27 (3.5)	19 (3.1)
I had opportunities to rehearse instructional practices during the professional development (i.e., try out, receive feedback, and reflect on those practices).	28 (3.8)	16 (2.5)	25 (2.6)	14 (2.6)	18 (3.1)
I had opportunities to apply what I learned to my classroom and then come back and talk about it as part of the professional development.	25 (3.8)	16 (2.9)	20 (3.4)	18 (2.8)	22 (3.3)
I worked closely with other teachers from my school.	42 (4.8)	17 (2.4)	15 (3.4)	13 (3.1)	13 (2.6)
I worked closely with other teachers who taught the same grade and/or subject whether or not they were from my school.	15 (2.8)	15 (3.2)	19 (3.5)	23 (3.1)	28 (3.8)

† Includes only high school teachers indicating in Q15 that they participated in computer science-focused professional development in the last three years.

Table CTQ 19
High School Computer Science Teachers' Perceptions of Topics Emphasized During Professional Development in the Last Three Years

	PERCENT OF TEACHERS†				
	NOT AT ALL 1	2	SOMEWHAT 3	4	TO A GREAT EXTENT 5
Deepening your own computer science content knowledge, including programming	4 (2.1)	5 (1.4)	21 (3.0)	28 (3.1)	42 (3.9)
Deepening your understanding of how computer science is done (e.g., breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)	5 (1.3)	10 (2.0)	22 (3.1)	30 (3.0)	33 (3.7)
Implementing the computer science textbook/online course to be used in your classroom	15 (2.9)	15 (3.2)	20 (3.3)	22 (2.9)	28 (3.9)
Learning how to use programming activities that require a computer	5 (1.4)	10 (2.7)	22 (3.2)	27 (3.8)	36 (3.9)
Learning about difficulties that students may have with particular computer science ideas and/or practices	13 (3.0)	15 (2.7)	24 (2.8)	26 (3.5)	22 (3.8)
Monitoring student understanding during computer science instruction	14 (2.7)	20 (3.0)	26 (4.2)	23 (3.3)	17 (3.5)
Differentiating computer science instruction to meet the needs of diverse learners	15 (2.6)	29 (4.0)	27 (3.1)	18 (3.1)	11 (3.0)
Incorporating students' cultural backgrounds into computer science instruction	31 (3.6)	18 (3.0)	26 (3.7)	13 (2.4)	12 (3.1)
Learning how to provide computer science instruction that integrates engineering, mathematics, and/or science	14 (2.4)	18 (2.8)	32 (3.8)	21 (3.1)	14 (2.7)

† Includes only high school teachers indicating in Q15 that they participated in computer science-focused professional development in the last three years.

Table CTQ 20
High School Computer Science Teachers' Perceptions of Their Preparedness to Teach Various Topics

	PERCENT OF TEACHERS			
	NOT ADEQUATELY PREPARED	SOMEWHAT PREPARED	FAIRLY WELL PREPARED	VERY WELL PREPARED
Computing systems	7 (1.4)	28 (3.2)	35 (3.5)	31 (3.9)
Networks and the Internet	11 (2.1)	35 (4.1)	31 (3.6)	23 (3.4)
Data and analysis	9 (1.9)	24 (2.5)	39 (3.6)	27 (4.1)
Algorithms and programming	5 (1.6)	14 (2.0)	34 (4.1)	47 (4.0)
Impacts of computing	6 (1.7)	19 (2.5)	40 (3.7)	35 (3.4)

Table CTQ 21
High School Computer Science Teachers' Perceptions of Their Preparedness for Each of a Number of Tasks

	PERCENT OF TEACHERS			
	NOT ADEQUATELY PREPARED	SOMEWHAT PREPARED	FAIRLY WELL PREPARED	VERY WELL PREPARED
Develop students' conceptual understanding of the computer science ideas you teach	2 (0.8)	11 (2.1)	45 (3.9)	42 (3.6)
Develop students' abilities to do computer science (e.g., breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)	1 (0.7)	15 (2.5)	36 (3.4)	48 (3.7)
Develop students' awareness of STEM careers	3 (1.8)	21 (2.9)	40 (4.7)	36 (4.2)
Provide computer science instruction that is based on students' ideas (whether completely correct or not) about the topics you teach	7 (2.2)	22 (3.6)	43 (4.7)	28 (3.9)
Use formative assessment to monitor student learning	2 (0.6)	19 (2.7)	44 (3.7)	35 (3.4)
Differentiate computer science instruction to meet the needs of diverse learners	11 (2.4)	35 (3.8)	32 (3.5)	21 (3.3)
Incorporate students' cultural backgrounds into computer science instruction	20 (2.9)	39 (3.6)	25 (3.6)	16 (3.1)
Encourage students' interest in computer science	2 (1.2)	8 (1.6)	41 (3.6)	49 (3.6)
Encourage participation of all students in computer science	3 (1.5)	15 (3.1)	36 (3.8)	45 (3.8)

Table CTQ 22
High School Computer Science
Teachers' Opinions About Teaching and Learning

	PERCENT OF TEACHERS				
	STRONGLY DISAGREE	DISAGREE	NO OPINION	AGREE	STRONGLY AGREE
Students learn computer science best in classes with students of similar abilities.	3 (1.3)	34 (3.3)	12 (2.1)	35 (3.1)	15 (2.5)
It is better for computer science instruction to focus on ideas in depth, even if that means covering fewer topics.	1 (1.0)	20 (2.6)	20 (3.4)	45 (4.1)	13 (2.9)
At the beginning of instruction on a computer science idea, students should be provided with definitions for new vocabulary that will be used.	1 (0.4)	11 (2.0)	13 (2.4)	52 (3.6)	23 (3.2)
Most class periods should provide opportunities for students to share their thinking and reasoning.	0 ---†	3 (0.8)	7 (2.5)	54 (4.2)	36 (4.2)
Hands-on/manipulatives/programming activities should be used primarily to reinforce a computer science idea that the students have already learned.	2 (0.9)	15 (2.8)	11 (2.8)	35 (3.2)	36 (3.6)
Teachers should ask students to justify their solutions to a computational problem.	0 (0.3)	2 (0.7)	6 (1.4)	60 (3.9)	32 (3.7)
Students learn best when instruction is connected to their everyday lives.	0 ---†	2 (0.8)	8 (1.8)	49 (3.9)	40 (4.1)
Most class periods should provide opportunities for students to apply computer science ideas to real-world contexts.	0 ---†	4 (1.3)	17 (2.7)	46 (4.1)	33 (4.3)
Students should learn computer science by doing computer science (e.g., breaking problems into smaller parts, considering the needs of a user, creating computational artifacts).	0 ---†	0 ---†	3 (1.2)	34 (4.0)	63 (4.2)

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 23
High School Computer Science Teachers Having
Various Leadership Responsibilities Within the Last Three Years

	PERCENT OF TEACHERS
Served as a lead teacher or department chair	36 (3.6)
Served as a formal mentor or coach for a computer science teacher	10 (2.2)
Supervised a student teacher in your classroom	15 (2.6)
Served on a school district-wide/diocese-wide computer science committee (e.g., developing curriculum, developing pacing guides, selecting instructional materials)	39 (4.0)
Led or co-led a workshop or professional learning community (e.g., teacher study group, lesson study) for other teachers focused on computer science or computer science teaching	22 (3.1)
Taught a computer science lesson for other teachers to observe	36 (3.7)
Observed another teacher's computer science lesson for the purpose of giving him/her feedback	17 (2.7)

Table CTQ 24
Average Minutes Per Week High School Computer Science Classes Meet

	AVERAGE NUMBER OF MINUTES
Instructional time per week	241 (5.2)

Table CTQ 25
Average Number of Students in High School Computer Science Classes

	AVERAGE NUMBER OF STUDENTS
High school computer science classes	17 (0.8)

Table CTQ 26
Race/Ethnicity of Students in High School Computer Science Classes

	AVERAGE PERCENT OF STUDENTS
American Indian or Alaskan Native	0 (0.1)
Asian	13 (1.9)
Black or African American	8 (1.3)
Hispanic or Latino	16 (2.8)
Native Hawaiian or Other Pacific Islander	1 (0.5)
White	59 (2.9)
Two or more races	3 (0.8)

Table CTQ 27
Prior Achievement Level of Students in High School Computer Science Classes

	PERCENT OF STUDENTS
Mostly low achievers	0 (0.4)
Mostly average achievers	23 (2.8)
Mostly high achievers	36 (4.4)
A mixture of levels	41 (4.4)

Table CTQ 28
High School Computer Science Classes in Which Teachers Report Having Control Over Various Curricular and Instructional Decisions

	PERCENT OF CLASSES				
	NO CONTROL		MODERATE CONTROL		STRONG CONTROL
	1	2	3	4	5
Determining course goals and objectives	5 (1.5)	6 (1.6)	15 (2.6)	17 (4.2)	57 (4.3)
Selecting curriculum materials (e.g., textbooks/online courses)	4 (1.3)	5 (1.4)	15 (2.4)	19 (4.1)	58 (4.7)
Selecting content, topics, and skills to be taught	4 (1.3)	8 (1.7)	13 (1.9)	21 (4.2)	53 (4.2)
Selecting programming languages to use	13 (2.2)	10 (2.2)	11 (2.5)	17 (3.9)	49 (4.3)
Selecting the sequence in which topics are covered	2 (1.0)	5 (1.8)	10 (1.7)	19 (3.3)	63 (4.2)
Determining the amount of instructional time to spend on each topic	1 (0.9)	2 (1.0)	11 (2.4)	23 (3.4)	63 (4.4)
Selecting teaching techniques	0 (0.4)	2 (1.6)	8 (1.9)	22 (4.0)	68 (4.5)
Determining the amount of homework to be assigned	0 (0.3)	2 (1.0)	6 (1.5)	15 (3.2)	77 (3.6)
Choosing criteria for grading student performance	1 (0.6)	5 (2.7)	6 (1.5)	17 (3.4)	71 (4.1)

Table CTQ 29
Emphasis Given in High School Computer Science Classes to Various Instructional Objectives

	PERCENT OF CLASSES			
	NONE	MINIMAL EMPHASIS	MODERATE EMPHASIS	HEAVY EMPHASIS
Learning computer science vocabulary and/or program syntax	0 ---†	12 (2.3)	56 (4.2)	33 (3.9)
Understanding computer science concepts	0 ---†	5 (1.8)	40 (3.4)	55 (3.6)
Learning how to do computer science (e.g., breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)	0 (0.2)	3 (1.0)	37 (3.7)	60 (3.5)
Learning how to develop computational solutions	0 (0.2)	10 (2.2)	47 (4.1)	43 (4.1)
Learning about real-life applications of computer science	0 (0.1)	15 (2.5)	46 (4.0)	39 (4.3)
Increasing students' interest in computer science	0 (0.1)	8 (1.8)	43 (3.9)	50 (3.6)
Developing students' confidence that they can successfully pursue careers in computer science	0 ---†	4 (1.3)	43 (3.7)	52 (3.9)

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 30
High School Computer Science Classes in Which Teachers Report Using Various Activities in Their Classrooms

	PERCENT OF CLASSES				
	NEVER	RARELY (E.G., A FEW TIMES A YEAR)	SOMETIMES (E.G., ONCE OR TWICE A MONTH)	OFTEN (E.G., ONCE OR TWICE A WEEK)	ALL OR ALMOST ALL COMPUTER SCIENCE LESSONS
Explain computer science ideas to the whole class	0 (0.1)	3 (1.1)	13 (2.7)	57 (4.0)	27 (3.4)
Engage the whole class in discussions	0 (0.2)	5 (1.8)	23 (3.2)	44 (3.6)	27 (3.4)
Have students work in small groups	1 (0.3)	11 (2.5)	22 (3.5)	36 (4.2)	30 (2.8)
Have students do hands-on/manipulative programming activities that do not require a computer	8 (2.2)	39 (4.2)	32 (4.0)	13 (2.9)	8 (2.3)
Have students work on programming activities using a computer	0 ---†	1 (0.5)	3 (1.3)	27 (3.7)	69 (3.7)
Use flipped instruction (have students watch lectures/demonstrations outside of class to prepare for in-class activities)	28 (3.5)	31 (4.2)	17 (2.8)	16 (2.5)	8 (2.4)
Have students read from a textbook/online course in class, either aloud or to themselves	22 (3.8)	26 (3.4)	21 (3.5)	25 (3.7)	6 (2.1)
Have students explain and justify their method for solving a problem	1 (0.6)	8 (1.8)	28 (3.4)	44 (4.5)	19 (4.2)
Have students present their solution strategies to the rest of the class	6 (1.6)	22 (3.0)	37 (3.8)	29 (3.7)	6 (2.2)
Have students compare and contrast different methods for solving a problem	4 (1.5)	22 (3.6)	33 (3.6)	33 (3.4)	8 (2.4)
Have students write their reflections (e.g., in their journals, on exit tickets) in class or for homework	22 (3.8)	28 (3.5)	18 (3.0)	19 (3.5)	13 (3.4)
Focus on literacy skills (e.g., informational reading or writing strategies)	19 (2.4)	32 (4.0)	29 (3.9)	16 (2.8)	4 (2.0)

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 31
High School Computer Science Classes in Which Teachers Report Students Engaging in Various Aspects of Computer Science Practices

	PERCENT OF CLASSES				
	NEVER	RARELY (E.G., A FEW TIMES A YEAR)	SOMETIMES (E.G., ONCE OR TWICE A MONTH)	OFTEN (E.G., ONCE OR TWICE A WEEK)	ALL OR ALMOST ALL COMPUTER SCIENCE LESSONS
Create computational artifacts (e.g., programs, simulations, visualizations, digital animations, robotic systems, or apps)	3 (1.0)	8 (2.2)	14 (2.5)	26 (3.3)	50 (4.1)
Create a computational artifact designed to be used by someone outside the class or other students	14 (2.7)	38 (3.9)	26 (3.3)	15 (3.1)	6 (2.2)
Provide feedback on other students' computational products or designs	3 (1.6)	20 (3.5)	29 (3.5)	37 (4.5)	10 (1.9)
Get input on computational products or designs from people with different perspectives (do not include feedback that you give students)	16 (3.1)	31 (3.8)	32 (3.6)	16 (3.6)	5 (2.1)
Systematically use test cases to verify program performance and/or identify problems	11 (2.7)	23 (4.1)	21 (2.7)	33 (4.2)	13 (3.0)
Identify real-world problems that might be solved computationally	1 (0.6)	16 (2.9)	37 (4.7)	29 (3.9)	17 (3.7)
Consider how a program they are creating can be separated into modules/procedures/objects	2 (0.9)	10 (1.7)	26 (3.5)	37 (3.7)	25 (3.4)
Identify and adapt existing code to solve a new computational problem	2 (0.9)	9 (1.8)	30 (3.4)	45 (3.7)	14 (2.4)
Use computational methods to simulate events or processes (e.g., rolling dice, supply and demand)	7 (2.0)	12 (2.8)	36 (3.9)	36 (3.5)	10 (1.9)
Analyze datasets using a computer to detect patterns	25 (3.7)	24 (3.1)	32 (3.5)	15 (3.3)	5 (2.1)
Write comments within code to document purposes or features	0 (0.2)	7 (1.9)	21 (2.7)	39 (3.8)	33 (4.3)
Create instructions for an end-user explaining how to use a computational artifact	17 (3.2)	23 (3.0)	31 (3.9)	21 (3.4)	9 (2.9)
Explain computational solution strategies verbally or in writing	4 (1.1)	15 (2.4)	39 (4.0)	32 (3.9)	11 (3.3)
Compare and contrast the strengths and limitations of different representations such as flow charts, tables, code, or pictures	19 (2.8)	32 (3.7)	28 (3.0)	17 (3.3)	6 (2.2)

Table CTQ 32
Provision of Technologies in High School Computer Science Classes

	PERCENT OF CLASSES		
	COMPUTERS (DESKTOPS OR LAPTOPS)	MOBILE COMPUTING DEVICES (TABLETS OR SMARTPHONES)	DATA STORAGE DEVICES
Not required for this class	0 ---†	57 (4.2)	46 (3.3)
Provided by the school, and students are not allowed to use their own	35 (4.5)	9 (2.2)	9 (2.8)
Provided by the school, but students are allowed to use their own	58 (4.5)	15 (2.3)	26 (3.4)
Students are expected to provide their own, but the school has some available for use	2 (0.7)	10 (2.9)	7 (2.2)
Students are required to provide their own	5 (1.6)	8 (3.4)	13 (2.4)

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 33
Availability of Instructional Technology in High School Computer Science Classes

	PERCENT OF CLASSES		
	NOT AVAILABLE	AVAILABLE UPON REQUEST	ALWAYS AVAILABLE IN YOUR CLASSROOM
Probes for collecting data (e.g., motion sensors, temperature probes)	60 (3.9)	25 (3.2)	16 (3.1)
Projection devices (e.g., Smartboard, document camera, LCD projector)	1 (0.5)	12 (2.8)	87 (2.9)
Robotics equipment	43 (3.3)	30 (3.7)	26 (3.6)

Table CTQ 34
Amount of Homework Assigned in High School Computer Science Classes Per Week

	PERCENT OF CLASSES
None	16 (2.6)
1–15 minutes per week	13 (2.9)
16–30 minutes per week	22 (4.4)
31–60 minutes per week	29 (3.9)
61–90 minutes per week	12 (2.5)
91–120 minutes per week	4 (1.0)
More than 2 hours per week	4 (1.2)

Table CTQ 35
Frequency of Use of Various Instructional Resources in High School Computer Science Classes

	PERCENT OF CLASSES				
	NEVER	RARELY (E.G., A FEW TIMES A YEAR)	SOMETIMES (E.G., ONCE OR TWICE A MONTH)	OFTEN (E.G., ONCE OR TWICE A WEEK)	ALL OR ALMOST ALL COMPUTER SCIENCE LESSONS
Commercially published textbooks (printed or electronic), including the supplementary materials (e.g., worksheets) that accompany the textbooks	36 (3.6)	21 (3.4)	17 (2.9)	17 (3.2)	9 (1.8)
State, county, or district/diocese-developed units or lessons	69 (4.4)	12 (2.3)	12 (2.9)	4 (1.2)	3 (1.9)
Online units or courses that students work through at their own pace (e.g., MOOCS, EdX, IMACS)	33 (3.2)	15 (2.9)	19 (3.3)	18 (3.3)	15 (3.3)
Lessons or resources from websites that have a subscription fee or per lesson cost (e.g., BrainPOP, Discovery Ed, Teachers Pay Teachers)	63 (4.0)	11 (2.0)	17 (3.3)	6 (1.7)	3 (1.4)
Lessons or resources from websites that are free (e.g., Khan Academy, code.org)	14 (2.8)	15 (2.5)	28 (3.6)	22 (3.9)	21 (4.0)
Units or lessons you created (either by yourself or with others)	6 (2.2)	7 (1.4)	23 (3.2)	35 (3.9)	28 (4.2)
Units or lessons you collected from any other source (e.g., conferences, journals, colleagues, university or museum partners)	14 (2.9)	17 (3.4)	41 (4.2)	22 (3.4)	6 (1.4)

Table CTQ 36
High School Computer Science Classes for Which the District/Diocese Designates Instructional Materials to Be Used

	PERCENT OF CLASSES
Instructional materials designated by district/diocese	26 (3.7)

Table CTQ 37
High School Computer Science Classes for Which Various Types of Instructional Materials Are Designated

	PERCENT OF CLASSES
Commercially published textbooks (printed or electronic), including the supplementary materials (e.g., worksheets) that accompany the textbooks	14 (3.0)
State county/district/diocese-developed instructional materials	2 (1.0)
Online units or courses that students work through at their own pace (e.g., MOOCs, EdX, IMACS)	4 (1.0)
Lessons or resources from websites that have a subscription fee or per lesson cost (e.g., BrainPOP, Discovery Ed, Teachers Pay Teachers)	9 (3.2)
Lessons or resources from websites that are free (e.g., Khan Academy, code.org)	15 (4.0)

There is no table for CTQ 38.

Table CTQ 39a
Copyright Year of Instructional Materials Used in High School Computer Science Classes

	PERCENT OF CLASSES†
2018	0 ---‡
2017	8 (5.2)
2016	9 (3.6)
2015	12 (4.6)
2014	4 (2.1)
2013	9 (5.2)
2012 or earlier	59 (7.0)

† Includes only high school computer science classes for which teachers indicated in Q37 that they use commercially published textbooks.

‡ No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 39b
Publishers of Textbooks Used in High School Computer Science Classes

	PERCENT OF CLASSES†
Pearson	24 (5.6)
Cengage	23 (5.9)
Skylight	12 (4.6)
Wiley	8 (3.8)
Project Lead The Way	6 (2.5)
Jones & Bartlett Learning	5 (3.2)
D&S Marketing Systems	3 (2.9)
Goodheart-Wilcox	3 (2.0)
Stacey Armstrong	3 (2.2)
Apple Inc. Education	2 (1.6)
Emc Publishing	2 (2.1)
Microsoft Press	2 (1.6)
O'Reilly Media	2 (1.4)
Virtualbookworm.com Publishing	2 (1.4)
Barron's Educational Series	1 (1.3)
McGraw-Hill Education	1 (0.5)
Oracle	1 (0.8)
Oxford University Press	1 (1.0)
Springer Nature	1 (0.9)
STEM Fuse	0 (0.5)

† Includes only high school computer science classes for which teachers indicated in Q37 that they use commercially published textbooks.

There is no table for CTQ 40.

Table CTQ 41
High School Computer Science Classes in Which Teachers Report the Effect Various Factors Have on Computer Science Instruction

	PERCENT OF CLASSES					
	INHIBITS EFFECTIVE INSTRUCTION		NEUTRAL OR MIXED		PROMOTES EFFECTIVE INSTRUCTION	N/A
	1	2	3	4	5	
Current state standards	2 (1.0)	5 (1.8)	34 (3.7)	10 (2.3)	18 (3.3)	30 (4.5)
Textbook selection policies	4 (1.3)	4 (1.0)	37 (4.2)	5 (1.2)	12 (2.9)	39 (4.7)
Teacher evaluation policies	3 (1.0)	5 (1.5)	40 (4.4)	17 (3.6)	21 (3.8)	14 (2.4)
College entrance requirements	1 (0.5)	3 (0.9)	40 (4.5)	16 (2.9)	22 (3.4)	18 (3.0)
Students' prior knowledge and skills	5 (1.4)	10 (2.7)	24 (3.4)	31 (3.4)	25 (2.9)	6 (2.7)
Students' motivation, interest, and effort in computer science	2 (1.0)	8 (2.3)	14 (3.2)	25 (3.3)	50 (3.7)	1 (0.4)
Parent/guardian expectations and involvement	1 (0.6)	7 (1.9)	43 (3.6)	20 (3.6)	19 (3.5)	10 (2.8)
Principal support	1 (0.6)	1 (0.8)	17 (2.6)	25 (3.8)	48 (4.3)	7 (2.4)
Amount of time for you to plan, individually and with colleagues	6 (1.7)	4 (1.2)	17 (3.3)	23 (3.4)	43 (3.5)	6 (1.4)
Amount of time available for your professional development	4 (1.1)	8 (1.8)	21 (3.4)	24 (3.9)	40 (3.8)	4 (1.0)

Table CTQ 42
High School Computer Science Classes in Which Teachers Report Technology Problems

	PERCENT OF CLASSES		
	NOT A SIGNIFICANT PROBLEM	SOMEWHAT OF A PROBLEM	SERIOUS PROBLEM
Lack of reliable access to the Internet	81 (4.4)	15 (4.2)	5 (1.6)
Lack of functioning computing devices (e.g., desktop computers, laptop computers, tablets, smartphones)	73 (4.5)	19 (4.2)	8 (2.2)
Insufficient power sources for devices (e.g., electrical outlets, charging stations)	86 (3.1)	10 (2.8)	4 (1.2)
Lack of support to maintain technology (e.g., repair broken devices, install software)	66 (4.4)	25 (3.4)	9 (2.7)
School restrictions on Internet content that is allowed	63 (4.3)	29 (3.3)	9 (2.4)

Table CTQ 43
Focus of the Most Recently Completed High School Computer Science Unit

	PERCENT OF CLASSES
Computing systems	7 (1.8)
Networks and the Internet	5 (1.8)
Data and analysis	1 (0.3)
Algorithms and programming	81 (3.0)
Impacts of computing	6 (1.9)

Table CTQ 44
Most Recent High School Computer Science Unit Based Primarily on Any Commercially Published Textbook or State/County/District-Developed Materials

	PERCENT OF CLASSES†
Most recent unit based on commercially published textbook or state/county/district-developed materials	63 (5.4)

† Includes only high school computer science classes for which teachers indicated in Q35 that they use commercially published textbooks or state/county/district/diocese-developed units or lessons more than once a month.

Table CTQ 45
Ways Instructional Materials Were Used in the Most Recently Completed Unit in High School Computer Science Classes

	PERCENT OF CLASSES†				
	NOT AT ALL 1	2	SOMEWHAT 3	4	TO A GREAT EXTENT 5
I used these materials to guide the structure and content emphasis of the unit.	0 (0.2)	0 ---‡	16 (3.6)	38 (6.6)	46 (7.1)
I picked what is important from these materials and skipped the rest.	5 (2.1)	15 (4.2)	31 (6.3)	27 (6.0)	22 (6.5)
I incorporated activities (e.g., problems, investigations, readings) from other sources to supplement what these materials were lacking.	1 (0.9)	6 (3.8)	22 (5.2)	48 (6.5)	22 (6.5)
I modified activities from these materials.	6 (3.1)	10 (3.6)	28 (5.4)	35 (6.4)	21 (5.9)

† Includes only high school computer science classes for which teachers responded yes in Q44.

‡ No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 46
Reasons Parts of the Instructional Materials Were Skipped in High School Computer Science Classes

	PERCENT OF CLASSES†		
	NOT A FACTOR	A MINOR FACTOR	A MAJOR FACTOR
The computer science ideas addressed in the activities I skipped are not included in my pacing guide/standards.	51 (6.7)	39 (6.8)	11 (4.0)
I did not have the materials needed to implement the activities I skipped.	72 (7.0)	19 (6.1)	8 (4.0)
I did not have the knowledge needed to implement the activities I skipped.	65 (7.5)	30 (7.3)	6 (2.6)
The activities I skipped were too difficult for my students.	49 (7.2)	44 (6.8)	8 (2.9)
My students already knew the computer science ideas or were able to learn them without the activities I skipped.	56 (6.2)	36 (6.1)	8 (3.0)
I have different activities for those computer science ideas that work better than the ones I skipped.	32 (5.6)	48 (6.6)	20 (4.9)
I did not have enough instructional time for the activities I skipped.	40 (5.8)	44 (7.0)	16 (4.2)

† Includes only high school computer science classes for which teachers responded yes in Q44 and indicated in Q45 that they “picked what was important from these materials and skipped the rest” to any extent.

Table CTQ 47
Reasons Why the Instructional Materials Were Supplemented in High School Computer Science Classes

	PERCENT OF CLASSES†		
	NOT A FACTOR	A MINOR FACTOR	A MAJOR FACTOR
My pacing guide indicated that I should use supplemental activities.	66 (6.3)	30 (6.3)	4 (1.8)
Supplemental activities were needed to prepare students for standardized tests.	48 (6.9)	39 (7.5)	13 (3.4)
Supplemental activities were needed to provide students with additional practice.	21 (5.0)	43 (6.8)	36 (6.0)
Supplemental activities were needed so students at different levels of achievement could increase their understanding of the ideas targeted in each activity.	27 (5.6)	42 (6.5)	31 (5.7)
I had additional activities that I liked.	21 (5.7)	49 (7.5)	29 (6.1)

† Includes only high school computer science classes for which teachers responded yes in Q44 and indicated in Q45 that they “incorporated activities (e.g., problems, investigations, readings) from other sources to supplement what these materials were lacking” to any extent.

Table CTQ 48
Reasons Why the Instructional Materials Were Modified in High School Computer Science Classes

	PERCENT OF CLASSES†		
	NOT A FACTOR	A MINOR FACTOR	A MAJOR FACTOR
I did not have the necessary materials/supplies for the original activities.	68 (7.1)	29 (7.2)	3 (1.5)
The original activities were too difficult conceptually for my students.	57 (6.5)	35 (6.6)	8 (2.7)
The original activities were too easy conceptually for my students.	67 (6.3)	32 (6.3)	1 (0.8)
I did not have enough instructional time to implement the activities as designed.	46 (6.5)	47 (6.7)	7 (2.6)
The original activities were too structured for my students.	69 (6.6)	29 (6.6)	2 (1.3)
The original activities were not structured enough for my students.	63 (7.3)	35 (7.3)	2 (1.5)

† Includes only high school computer science classes for which teachers responded yes in Q44 and indicated in Q45 that they “modified activities from these materials” to any extent.

Table CTQ 49
High School Computer Science Classes Taught by Teachers Feeling Prepared for Each of a Number of Tasks in the Most Recent Unit

	PERCENT OF CLASSES			
	NOT ADEQUATELY PREPARED	SOMEWHAT PREPARED	FAIRLY WELL PREPARED	VERY WELL PREPARED
Anticipate difficulties that students may have with particular computer science ideas and procedures in this unit	3 (0.9)	18 (3.2)	53 (4.5)	26 (3.9)
Find out what students thought or already knew about the key computer science ideas	1 (0.8)	20 (3.2)	50 (3.8)	29 (4.6)
Implement the instructional materials (e.g., textbook, online course) to be used during this unit	2 (0.7)	15 (2.8)	42 (4.2)	41 (4.2)
Monitor student understanding during this unit	1 (0.5)	14 (3.1)	42 (4.2)	43 (4.6)
Assess student understanding at the conclusion of this unit	2 (0.8)	12 (2.8)	45 (3.9)	41 (4.0)

Table CTQ 50
Duration of the Most Recent High School Computer Science Lesson

	AVERAGE NUMBER OF MINUTES
Duration of lesson	61 (1.9)

Table CTQ 51
Average Percentage of Time Spent on Different Activities in the Most Recent High School Computer Science Lesson

	AVERAGE PERCENT OF CLASS TIME
Non-instructional activities (e.g., attendance taking, interruptions)	9 (0.5)
Whole class activities (e.g., lectures, explanations, discussions)	29 (2.3)
Small group work	22 (2.1)
Students working individually (e.g., reading textbooks, programming, taking a test or quiz)	40 (2.1)

Table CTQ 52
High School Computer Science Classes Participating in Various Activities in the Most Recent Lesson

	PERCENT OF CLASSES
Teacher explaining a computer science idea to the whole class	70 (3.7)
Teacher conducting a demonstration while students watched	46 (3.6)
Whole class discussion	49 (4.1)
Students working in small groups	57 (4.2)
Students completing textbook/worksheet problems	16 (3.0)
Students doing hands-on/manipulative programming activities not using a computer	19 (2.9)
Students working on programming tasks using a computer	84 (2.8)
Students reading about computer science	20 (2.8)
Students writing about computer science (do not include students taking notes)	13 (3.0)
Test or quiz	9 (1.6)
None of the above	0 ---†

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 53
Sex of High School Computer Science Teachers

	PERCENT OF TEACHERS
Female	40 (3.6)
Male	60 (3.6)
Other	0 ---†

† No high school computer science teachers in the sample selected this response option. Thus, it is not possible to calculate the standard error of this estimate.

Table CTQ 54
High School Computer Science Teachers of Hispanic or Latino Origin

	PERCENT OF TEACHERS
Hispanic or Latino	8 (2.2)

Table CTQ 55
Race of High School Computer Science Teachers

	PERCENT OF TEACHERS
American Indian or Alaskan Native	2 (0.5)
Asian	4 (1.4)
Black or African American	3 (1.3)
Native Hawaiian or Other Pacific Islander	1 (0.6)
White	94 (1.7)

Table CTQ 56
Age of High School Computer Science Teachers

	AVERAGE TEACHER AGE
Age	44 (1.0)