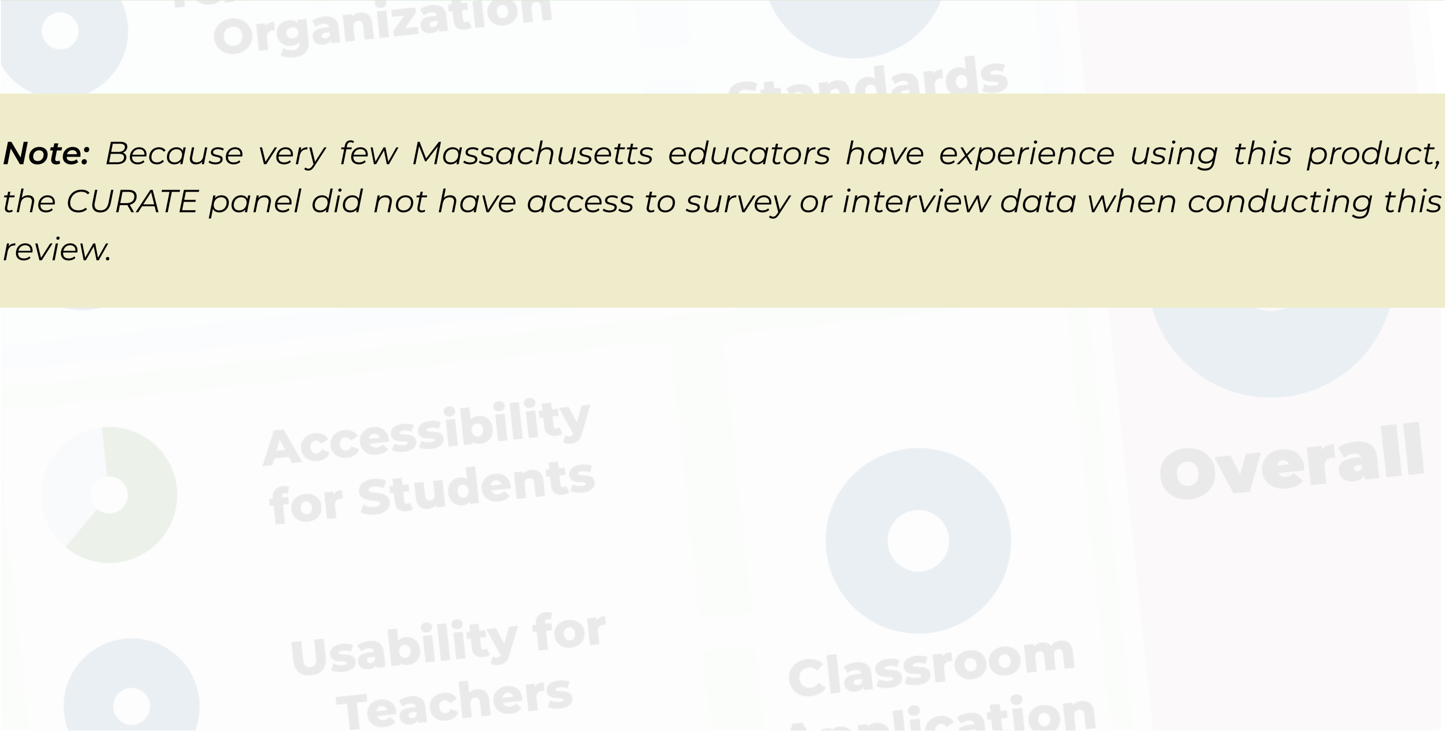
Computer Science Foundations

ellipsis education, 2022

PUBLICATION DATE: JANUARY 2025

Computer Science Foundations is a digital resource for Grades K-2. Please see the ellipsis education website and the publisher-provided information later in this report for product specifications.

Note: *Because very few Massachusetts educators have experience using this product, the CURATE panel did not have access to survey or interview data when conducting this review.*



Computer Science Foundations

ellipsis education, 2022

Digital Literacy & Computer Science, Grades K-2

Publication Date: January 2025

◕ **Scope and Progression**

◕ **Approach to Instruction**

◕ **Standards Alignment**

◕ **Accessibility for Students**

◕ **Usability for Teachers**

⏳ **Impact on Learning**

◕ **Classroom Application**

◕ **Overall**

● **Meets Expectations** - Most or all evidence indicates high quality; little to none indicates low quality. Materials may not be perfect, but Massachusetts teachers and students would be well served and strongly supported by them.

◕ **Partially Meets Expectations** - Some evidence indicates high quality, while some indicates low quality. Teachers in Massachusetts would benefit from having these materials but need to supplement or adapt them substantively to serve their students well.

◔ ***Does Not Meet Expectations*** - Little to no evidence indicates high quality; most or all evidence indicates low quality. Materials would not substantively help Massachusetts teachers and students meet the state's expectations for teaching and learning.

⏳ **No Rating** - Evidence is insufficient to generate rating.

Overall

The Bottom Line

Ellipsis *Computer Science Foundations* is a structured, engaging curriculum that addresses the majority of Massachusetts DLCS standards for students in grades K-2. However, materials do not consistently integrate DLCS practices with concepts. Tools provided within the materials for both teachers and students provide a foundation for learning computer science, with a focus on fundamentals that will position students for success in future grades. Teachers will need to supplement materials to affirm students' diverse identities and adequately support English learners (ELs).

Standards Alignment

Scope and Progression

Strengths

- Lessons are aligned with all strands of the DLCS standards. The publisher provides Massachusetts-specific standards alignment information, offering a map between each standard and the lesson that addresses the standard. For example, CS Foundations K, Day 6, Lesson 1 Learning Objectives contain language that is taken directly from the correlating DLCS standard. In addition, CS Foundations 1, Day 12, Lesson 3: “Who am I On-line” addresses DLCS K-2.CAS.b.1, K-2.CAS.c.1, and K-2. CAS.c.2.

Challenges

- Concepts do not build on one another across lessons. For example, the Engineering Design Process is taught in CS Foundations 2, Day 7, Lesson 3, but the process is applied only in the next lesson and then in the final project. Materials miss opportunities to meaningfully connect the process across a series of lessons.

- Materials include some above-grade level content without equipping students with the prior learning needed to grasp above-level concepts. For example, CS Foundations 2, Day 22, Lesson 3, designed for Grade 2, introduces the concepts of CSS, HTML, and Javascript. These text-based programming languages are not typically introduced until middle school or high school. The Massachusetts DLCS Curriculum Frameworks specify that programming in Grades K-2 should “not require a textual programming language” and should use, for example, “a blockbased programming language” (K-2.CT.d).

The Bottom Line

Materials are well-aligned with Massachusetts standards, with explicit alignment between the lessons and the standards. However, materials do not provide a clear progression of learning across lessons within a grade level, and they include some content that is well above grade level.

Approach to Instruction

Strengths

- The materials balance activities that provide information and engage students by offering a combination of context setting, skill development, practical application, and creative activities. This approach is conducive to engaging students in a well-rounded learning experience in digital literacy and computer science. For example, activities like "Drag and Drop Maze" (CS Foundations 1, Day 3, Lesson 1) focus on skill development by teaching students how to use block coding and algorithms. Lessons involving debugging, such as "Debug It," help students practice problem solving skills in real world scenarios (CS Foundations 2, Day 9, Lesson 1). Lessons on conditional statements and loops, like "Time for School" (CS Foundations 1, Day 8, Lesson 2) provide opportunities for students to apply coding concepts.

Challenges

- The materials only integrate some of the DLCS Practices with concepts. *Collaborating* is explicitly addressed and integrated with concepts in a number of lessons, such as CS Foundations 2, Day 4, Lesson 1, "Managing Frustration." Other practices, such as *Abstracting and Analyzing*, are not explicitly addressed. For example, CS Foundations K, Day 5, Lesson 1 includes instruction and an activity focused on connecting the brain to the computer. Though the lesson requires students to compare a computer to a human brain, it misses an opportunity to explicitly address the concept of *Abstracting*.

- Materials provide limited opportunities for students to explain solutions orally or to interact with other students to justify solutions. However, materials do encourage written explanations to problems as evidenced by exit tickets provided at the end of each lesson.

- The materials do not expose students to a sufficient range of representations and tools. Instead, they rely heavily on the use of ScratchJr to achieve the learning goals. While ScratchJr is an effective product to teach coding, students would benefit from learning through other tools as well.

Bottom Line

Materials support an approach to instruction that engages students in the development of skills, balanced with practical application and the opportunity to create computing artifacts. Teachers will need to supplement materials to cover all practices, provide students with opportunities to demonstrate their knowledge orally, justify their solutions, and expose students to a wider range of tools.

Classroom Application

Accessibility for Students

Strengths

- Lessons provide opportunities for students to access content in various ways. For example, the slide deck for CS Foundations 2, Day 21, Lesson 2 provides visual information on the day's lesson, including expectations for an assignment and supporting images (i.e., coding blocks). Visuals include a video to show coding blocks in action and a printed handout of criteria for a project. During the lesson, students also have opportunities to process content orally by brainstorming ideas and sharing with partners.

Challenges

- Materials do not adequately represent and affirm diverse cultures, backgrounds, and identities. Though materials incorporate photographs of individuals of various ethnicities, pictures are often outdated and appear out of context. Many of the "plugged-in" projects on ScratchJr do not take advantage of the app's wide-ranging character and background library and endless possibilities for customization. Additionally, the grading criteria for the creative computing experiences often limit student choice and agency rather than cultivate opportunities for self-expression.

- Supports for ELs are generic and are not consistently available across lessons. For example, EL guidance for CS Foundations 2, Day 23, Lesson 1 is based on a generic template suggesting that teachers use preferential seating, display the material and objective, translate select materials into the student's native language and encourage students to practice speaking. In some lessons, no specific EL recommendations are provided. For example, Grade 1, Day 23, Lesson 1 incorporates specialized computer vocabulary but there are no clearly identified supports provided to support ELs in accessing the lesson. Often EL recommendations are not available for non-coding lessons, which are typically more language intensive.

- Some lessons provide varied means for demonstrating learning but often lack support for students with disabilities or ELs. For example, in CS Foundations K, Day 26, Lesson 1, there are a couple of engaging ways for students to express their understanding of conditionals. Students turn and talk with peers and collaboratively brainstorm examples. Then students use physical blocks to create their own conditional, which serves as a formative assessment. Students also demonstrate learning via an exit ticket. However, these opportunities to demonstrate learning do not incorporate supports for students with disabilities or ELs to show what they have learned.

The Bottom Line

Materials provide a range of methods for students to access grade-appropriate content, as well as varied means for students to demonstrate learning. Teachers will need to supplement materials to affirm students' diverse identities within computer science instruction and provide supports that broaden access for ELs.

Usability for Teachers

Strengths

- Lessons and tasks advance student learning with clear purpose. For example, CS Foundations K, Day 13, Lesson 2 starts with a concise activity description which outlines the goals for the activity and learning objectives for the students. These are restated on the formative assessment form as well as the exit ticket to remind students and teachers of the lesson purpose.

- Pacing is reasonable and flexible; the curriculum can be effectively implemented in a range of scheduling models. Materials can be fully covered within one school year. For example, the CS Foundations K Pacing Guide suggests that lessons take 60 minutes per day and require 28 days of instruction to complete. The 60 minutes are broken into two 30-minute lesson topics which can be delivered over two days, offering teachers flexibility to implement the curriculum in a manner that fits within their schedule.

Challenges

- Materials include informal and formal assessments but lack guidance on how teachers should adjust instruction based on assessment findings. For example, CS Foundations 2, Day 1, Lesson 1 starts with a pre-assessment focused on student knowledge of overall subject matter including digital literacy, coding, etc. The pre-assessment also incorporates an exit ticket focused on how to access a ScratchJr project. Following the pre-assessment, materials lack detailed guidance on how teachers should use this information to adapt instruction to student needs.

- The materials include some guidance to build teachers' subject matter knowledge, although the guidance is inconsistent. For example, CS Foundations K, Day 3, Lesson 1 includes a callout box in the lesson plan that offers an activity tip to help increase student engagement. There is also a social-emotional learning (SEL) badge that indicates that the lesson can support SEL. However, the materials are inconsistent in the availability of these guides, thereby limiting their value toward building teacher subject matter knowledge.

- Although materials include some rubrics and exemplars to set high expectations for students, the availability of exemplars is inconsistent. For example, in CS Foundations 1, Day 30, Lesson 2, students are tasked with completing a final project. The materials include examples of high-quality work that students can use as a model for their own projects. The exemplars help students understand what a successful interactive game looks like and what they need to do to achieve that level of quality. The lesson also includes a rubric for students to understand how the project will be graded. Though these resources are strong, they are missing for many tasks.

- Materials include consistent routines and resources to support classroom activities but lack strategies to support implementation in the classroom. Each lesson incorporates “Activity Procedures” that provide structure for students to engage in discussion, followed by an activity, culminating in a “Challenge” for the students to complete. Lessons conclude with an exit ticket to gauge student comprehension. However, materials do not incorporate recommended grouping strategies and protocols. In addition, slides and handouts are not editable, making it difficult for teachers to modify activities based on the needs of their classes.

The Bottom Line

Ellipsis *Computer Science Foundations* provides well-structured and consistent lessons that advance student learning with clear purpose. Pacing is reasonable and materials can be completed within one school year. However, teachers will need to supplement materials to determine grouping strategies and adjust instruction based on assessment findings. Additionally, slides and handouts are not editable, limiting teachers’ ability to easily modify resources to meet student needs.

Impact on Learning

The Bottom Line

A DESE-commissioned policy brief found in 2018 that "research has yet to catch up to recent developments in curriculum materials." As with many comprehensive curriculum products currently in use, high-quality studies of student learning impacts are not yet available for *Computer Science Foundations*. This is a promising and important area for further study.

Looking for more information? Review the full 2022 DLCS Curriculum Guide.

What the Publisher Says....

We asked publishers for information on product specifications and technological requirements, professional learning opportunities for Massachusetts educators, and diversity of representation in their materials. See what Ellipsis Education had to say about *Computer Science Foundations* (2022).

Diverse Representation

Describe how you ensure that students of diverse races, ethnicities, nationalities, socioeconomic classes, family experiences, linguistic backgrounds, abilities, cultures, religions, genders, gender identities, sexual orientations, and other identities see themselves fully reflected and respected in your curriculum. For example, describe any bias or inclusivity review procedures you have in place and provide evidence of their efficacy.

Our curriculum leverages culturally relevant pedagogy (Ladson-Billings, 1995) as a central tenet of the content and focus so that teachers can empower all students to have access to CS curriculum, see themselves in CS, and engage in doing CS and using it as a tool to make their community and the world around them a better place. [Ellipsis Education] includes attention to the three broad characteristics of teacher-enacted culturally relevant pedagogy (CRP) in the lens of CS, including teacher conceptions of self and others (confidence to deliver the curriculum and break down barriers of access to all students from various underrepresented backgrounds), teacher structured social relations (teacher as a facilitator who pulls in student experience and ideas – funds of knowledge – as real-world examples), and teacher conceptions of CS knowledge (Ladson-Billings, 1995).

We ensure that the principles of CRP are upheld through our curriculum development process, which includes several measures to check for and mitigate against potential bias. First, all writers are trained on our writing and style guidelines, so that they can consistently apply the mechanical elements of inclusion. Second, curriculum topics that lend themselves to issues related to representation are brought to a weekly writer's meeting so that the writers can collaboratively plan the lesson. Members of our writing team reflect diverse backgrounds, so writers are able to weigh these topics against their own experiences to identify potential biases and marginalization. Next, all lessons at [Ellipsis Education] go through a formal review process which includes thoughtful attention to our inclusive practices. If improvement is needed, the lesson will be sent back to the original writer for revisions. Finally, we are, at all times, open to feedback from our customers which allows us to continually improve the quality of our curriculum.

[Ellipsis Education] includes images, descriptions, names, and examples that are racially diverse, gender-inclusive, culturally varied, and reflective of the varying levels of physical ability to ensure every student can see themselves in the various aspects of STEM covered in the curriculum.

Consider the grade-level lesson examples below as evidence of our approach to represent diverse learners.

- K - What is a Computer Scientist?
  - Students examine images of a diverse group of computer scientists that have influenced the field over time and consider how computer scientists solve problems.
- 5 - Seeking Diverse Perspectives
  - Students explore what it means to seek the perspective of others and gain an understanding of the value of diverse perspectives in collaborative environments.

See below for additional evidence of effectiveness in engaging diverse learners.

- ELL Guides
  - Based on WIDA language mastery levels, this documentation uses SIOP approaches to develop language goals. This allows educators to ensure students have access to a quality computer science curriculum regardless of English language proficiency.
- Social Emotional Learning (SEL)
  - Infused SEL tips emphasize interpersonal and social skills. These lesson tips focus on decision making, self-management, and relationship skills and work to engage diverse learners throughout their computer science journey and beyond.

Professional Learning

Describe any professional learning opportunities (materials or experiences, publisher-provided or otherwise) available for Massachusetts educators that are designed to support high-quality implementation of your curriculum.

[Ellipsis Education] is honored to support high-quality computer science instruction by providing teachers with dynamic and engaging professional learning opportunities. Our professional learning is usually facilitated virtually but can be delivered on-site if that works better for a district or school. Most districts choose from the below offerings to ensure teachers are poised for success with the curriculum.

1. **[Ellipsis Education] Implementation Training**: Implementation trainings will focus on two primary instructional needs:
   - Accessing [Ellipsis Education] Curriculum and Instructional Materials
     - The [Ellipsis Education] Customer Experience team ensures all training attendees have access to and are able to navigate the content and use of the instructional materials within their [Ellipsis Education] course(s).
   - Navigating the Software/Applications Used in the Curriculum
     - Educators will be oriented to and have a chance to explore the coding platform(s) they will use when teaching students.
2. **Computer Science Professional Learning Sessions:** The [Ellipsis Education] Professional Learning team can deliver additional professional development on a variety of topics of interest to CS educators. Topics are usually presented in 3-hour virtual sessions, with time set aside for educators to work independently on applying new learning to their unique contexts. Session titles include:
   - **Algorithm Basics for Educators** - Algorithms are one of the most fundamental concepts in Computer Science so it's important to get it "right" with your students. In this session, you will deepen your own understanding of algorithms through modeling, real-life examples, and semantic wave processes. Then, you'll refine your algorithm development skills through coding activities. Finally, you will learn a variety of strategies to help students succeed when working with algorithms.
   - **Computational Thinking in the Classroom** - Computational thinking is an essential skill that benefits students across content areas, grade levels, and applications. In this session, you will explore the everyday uses of computational thinking and identify opportunities to use computational thinking in your subject area. You will learn the four steps of the computational thinking process, apply your new learning in a variety of activities, and consider how to support students through those same steps. Finally, you will develop a plan for integrating at least one computational thinking concept into an upcoming lesson with students.
   - **Block Coding and Programming Basics** - Block coding programs allow for a powerful and effective introduction to coding. In this session, you will directly engage in the ScratchJr and Scratch block coding environments to develop your understanding of three core programming concepts: data structures, abstraction, and data flow. You will break these three broad concepts down

into sub-concepts and create a concept map showing how each of these are applied in block coding. Finally, you will explore the progressions of these concepts in computer science standards to better understand the targets your students will need to meet.

- **Meaningful Assessment for Computer Science** - Computer science provides wonderful opportunities to measure students' learning across a variety of outcomes, including grade-level standards and post-secondary readiness skills. In this session, you will experience a variety of assessment approaches and consider how each contributes to have a holistic view of students' learning. You will explore the purposes of assessment, as well as learn best practices for designing and implementing assessments with students. The session culminates in you articulating your personal assessment philosophy, which will include your action plan for a more meaningful assessment practice in your classroom.

Product Specifications

Describe what a school or district needs to implement your curriculum successfully, including instructional hours and technological infrastructure. Provide basic information about what products are associated with the curriculum (e.g., what texts a typical purchase includes, what tools are openly available online)

Curriculum Package

[Ellipsis Education] courses for grades K-5 take approximately 27 hours to complete in full, while grades 6-8 take approximately 66 hours to complete in full. Courses can be divided into 30, 45, or 60 minutes class periods in order to support a school's schedule. In situations where customers need more than the available hours, we often combine content from multiple courses to create a customized computer science pathway that fills the available instructional time.

[Ellipsis Education] Curriculum is delivered to teachers through Canvas LMS and integrates well with the tools districts are using to navigate in-classroom, hybrid, and distance learning. Each course includes: full-year, step-by-step curriculum with pacing guides, standards alignment documents, comprehensive teacher and student resources, English Language Learner Guides, teacher and student instructional slide decks for all coding lessons, full-year customer support, assessments, and educator onboard training.

Technology Alignment

[Ellipsis Education] Curriculum is delivered to teachers through Canvas LMS and integrates well with the tools districts are using to navigate classroom instruction. Teachers will be able to access their curriculum using any device with internet access.

Unplugged, Digital Citizenship, and STEM Career lessons may occasionally require the use of technology but are often very intentionally designed to give students a break from technology. As such, compatibility for each operating system and browser is dependent on each course. Coding lessons included in our curriculum can be completed on a variety of device types, including Chromebooks, iPads, and Windows computers. Specific device compatibility is listed in the next section.

Device Compatibility

[Ellipsis Education] recommends the use of the following devices for students in grades K-8:

- CS Foundations K-2
  - Online and Offline: Chromebook, iPad, Android Tablet
- CS Fundamentals 3-5:
  - Online and Offline: Windows, MAC, Chromebook, Android Tablet
  - Online Only: iPad
- Intro to CS Applications, CS Applications JavaScript, CS Applications Java:
  - Online and Offline: Windows, MAC, Chromebook,
  - Online Only: iPad, Android

Third Party Developer Environments
Students will need to be able to access the following coding environments in order to fully participate in the Coding activities included in the curriculum:

- CS Fundations K-2
  - ScratchJr
- CS Fundamentals 3-5
  - Scratch (online and offline access available)
- Intro to CS Applications, CS Applications JavaScript, HS JavaScript
  - Rep.lit (online access) and Brackets (offline access)

Response to Report

None submitted