Table of Contents:

1 How should I use this guide?

2 What is computational thinking?
   a. Computational thinking defined
   b. Frequently asked questions (FAQs)
   c. Computational thinking myths debunked
   d. The computational thinking modules
      - What is Computational Thinking?
      - Collecting Data
      - Analyzing Data
      - Finding Patterns
      - Abstraction
      - Decomposition
      - Building Models
      - Developing Algorithms

3 Why does this matter for students?
   a. Outcomes
   b. Careers
   c. Real-world problem solving skills
   d. Meeting students where they are: cultural responsiveness in the classroom
   e. Computational thinking for students with special needs

4 Who should teach computational thinking?
   a. Computational thinking in the humanities
      - English/Language Arts
      - History and Social Studies
   b. Computational thinking in STEM
      - Math
      - Science
   c. Computational thinking in elective courses
      - Music
      - Health and Physical Education
      - Art
      - World Languages
   d. Additional standards and guidelines

5 Where should I begin?
   a. Helpful resources
   b. Gain confidence with computational thinking
1. How should I use this guide?

Dear educator,

Congratulations! You’re about to embark on a journey with computational thinking that will have you (and your students) viewing the world in a whole new way. Computational thinking is a modern problem-solving approach that has roots in every imaginable discipline, from art to science and everything in between.

This guide and the accompanying self-paced module will orient you to computational thinking, dive deeply into the seven computational thinking strategies, explain how computational thinking fits into the work you are already doing and show you the richness that computational thinking adds to your instruction. It will provide practical tips for teaching computational thinking and give you resources along the way. At the end of the module, you will emerge with a sharp sense of how computational thinking equips your students for 21st century careers and you’ll have a toolbox full of techniques to increase engagement, expand cultural competency and diversify instruction in your classroom.

Be sure to take the modules at your own pace, and know that whether you are a computational novice or guru, this guide was made with you in mind. As an educator, you know that a spark of curiosity can light a fire of learning in your students. This guide and self-paced module will help you ignite the future of your students through computational thinking.

2. What is computational thinking?

Computational thinking isn’t about using computers: it’s about thinking like computers in order to solve problems. Humans and computers speak fundamentally different languages. In order to get a computer to complete a specific task, computer engineers have to explain the task to the computer in a language it understands. This is what we commonly refer to as “computer programming”.

Computational thinking is a logic-based approach that uses computer programming strategies to tackle real-world problems. In this guide and accompanying module, we’ll explore seven computational thinking strategies that computer engineers use every day: collecting data, analyzing data, finding patterns, abstracting, decomposing, building models and developing algorithms. We’ll take these strategies out of the computer lab and into the real world where we’ll see how approaches like decomposing problems into smaller sub-problems and finding patterns in data extend far beyond the world of computer science and apply to every subject.
2. What is computational thinking?

Frequently asked questions (FAQs)

Q: How did computational thinking originate?
A: Computational thinking is as old as computing itself. As soon as computers were developed, scientists needed ways to instruct them to carry out tasks. Jeanette Wing’s 2006 piece on computational thinking catapulted the thinking process out of the computer lab and into the classroom, where it has continued to gain traction. Many universities (such as Carnegie Mellon) have developed computational thinking centers that infuse a wide variety of disciplines with these problem-solving approaches.

Q: Is computational thinking really for me?
A: Yes! Computational thinking is all about developing skills that helps us solve problems. That means that computational thinking is important for everyone.

Q: Why is this important for my students?
A: Students will need computational thinking skills in order to succeed in the modern workforce. Strategies like abstracting out a common solution to many problems and testing out a hypothesis are useful in every profession. Computational thinking helps students live robust and productive lives by giving them tools that help them solve problems and showing them how they have the agency to make an impact on issues that matter to them.

Q: Do I have to make big changes to my curriculum in order to integrate computational thinking into my classroom?
A: No need to reinvent the wheel here! Chances are you’re already using computational thinking strategies in your classroom. The Ignite My Future in School computational thinking lessons provide you with a variety of real-life scenarios designed to pique students’ curiosity. Use the simulations and real-world connections in these lessons to make it clear to students that the material they’re learning has real-world implications.

Q: Does computational thinking align with national standards?

Q: Can I teach computational thinking if I don’t have computers in my classroom?
A: You sure can! Each Ignite My Future in School lesson plan has been designed for classrooms without access to digital devices. However, each lesson also includes fun ways to integrate technology if it’s available to you.

Q: My students have a very diverse set of needs. How can I diversify instruction of computational thinking strategies in order to meet students where they are?
A: Computational thinking is a great way to provide students with organizational techniques and problem-solving skills. Many Ignite My Future in School lessons utilize graphic organizers and are designed to address many learning styles. Computational thinking encourages collaboration and allows students to go at their own pace, making it a great fit for classrooms with diverse learners.

Q: I’m a new teacher. Why should I become proficient in computational thinking?
A: Computational thinking will help you develop powerful skills as an educator. By familiarizing yourself with the seven computational thinking strategies, you will have failsafe approaches that will help you teach any curriculum. The Ignite My Future in School curriculum incorporates many proven teaching techniques, such as checking for understanding and using collaborative structures.
2. What is computational thinking?

b) Frequently asked questions continued... (FAQs)
Q: I’m a seasoned teacher. Why should I change up what I’m doing to incorporate computational thinking?
A: It’s very likely that as a seasoned teacher you’re already using computational thinking approaches in your classroom. Computational thinking will help you connect your lessons to real-world problems and current events and provide you with direct connections to the careers and skills of tomorrow.

Q: Computational thinking sounds kind of technical. Does it relate directly to students’ lives?
A: Computational thinking may sound technical, but it’s fundamentally a human ability. Computational thinking is all about giving students tools they can use to solve the problems they encounter in their lives and empowering them to become changemakers in their communities.

Q: What’s the best way to learn how to teach computational thinking?
A: Collaborating with other educators is a great way to get started in computational thinking. This guide provides many professional development approaches that you might consider as you begin your journey. Be sure to visit the Ignite My Future in School resources page for a variety of materials that can help you get started.

c) Computational thinking myths debunked

Myth: Computational thinking is all about coding.
Reality: Computational thinking is a great way to teach coding and computer science, but it’s really a set of problem-solving strategies that apply to every discipline.

Myth: Computational thinking is only useful for teachers who specialize in STEM subjects.
Reality: Computational thinking is applicable to every classroom, from language arts to physical education and everything in between. Computational thinking strategies like abstraction, decomposition and finding patterns are foundational skills that help students dig deeply into subject matter and make connections between different curricular threads.

Myth: Computational thinking is a passing fad. These skills will be outdated by the time my students are in the workforce.
Reality: Computational thinking has grown in popularity since 1950s and occupational outlook forecasts estimate that jobs requiring computational thinking skills will only grow.

Myth: Teaching computational thinking will take valuable time and focus away from the liberal arts.
Reality: Computational thinking is a great way to explore the liberal arts. Check out some of the lesson plans from Ignite My Future in School to see how computational thinking applies to political science, poetry, music, theater and many other subjects.

Myth: You need computers in your classroom in order to teach computational thinking.
Reality: You can teach computational thinking effectively without using even one computer in your classroom! Computational thinking is, by nature, adaptable to any teaching environment.

Myth: I’m no good with computers, so I won’t be able to understand computational thinking.
Reality: Computational thinking strategies are simply logical problem-solving approaches. In fact, by learning computational thinking, you will likely become more proficient with computers because you will understand the fundamental logic behind computer programming.
2. What is computational thinking?

The computational thinking modules

This section of the professional learning guide provides overviews, pre- and post- activities, discussion questions, and supporting resources that accompany each of the eight self-paced modules. This flexible model allows educators to align computational thinking strategies with school or district goals, such as literacy, culturally responsive instruction, career and college readiness, or cooperative and collaborative classroom structures across disciplines. The introductory module will orient you to computational thinking and then dive deeply into the seven computational thinking strategies with the additional modules. Let’s get started!

Module: What Is Computational Thinking?

1. Overview: Computational thinking isn’t as much about what computers do as how they do it. This module provides an introduction to the seven core problem-solving strategies that comprise computational thinking and gives you the tools you need to bring this 21st century skill into your classroom, no matter what curriculum you’re teaching.

2. Pre-activity: Two Truths and One Lie: As you think about the definition of computational thinking, consider the following:
   a. Computational thinking involves thinking like a computer in order to solve real-world problems
   b. Computational thinking is focused on learning about computers
   c. Computational thinking is applicable to every subject
Which is the lie? You guessed it! Computational thinking is not just about computers.

3. Post-activity: Consider engaging with your subject or grade-alike colleagues during lunchtime or a conference period and brainstorm how everyone can use computational thinking in their classes.

4. Discussion questions:
   a. What are some examples of computational thinking strategies?
   b. How is computational thinking different than computer science?
   c. How does computational thinking directly relate to our lives and the lives of our students?
   d. How can we differentiate instruction on computational thinking so that it relates to students from every background?

5. Resources:
   a. Ignite My Future in School Career Vignettes
   b. Drop the Beat and Speak My Language are two computational thinking lessons that demonstrate that how computational thinking applies to every subject and can incorporate students from every background.
   c. Computational Thinking Curriculum Connectors
2. What is computational thinking?

The computational thinking modules continued...

Module: Collecting Data

1. Overview: This module digs into what data is, how to find it and the best ways to mitigate bias when collecting data. You’ll get some examples of how the computational thinking strategy of collecting data helps solve problems in the real world and we’ll discuss how to incorporate collecting data into your classroom.

2. Pre-activity: Set up a simple survey using SurveyMonkey.com and email it to a few of your colleagues. Here are some sample questions you could use:
   a. What is your favorite flavor of ice cream?
   b. What is your favorite season of the year?
   c. What is your favorite sports team?
   d. What is your favorite novel?
   e. What is your favorite flower?

Collect responses and devise of at least two ways to display this data.

3. Post-activity: Reflect on your experience with your survey and the information in the module with a group discussion amongst colleagues.

4. Discussion questions:
   a. Do you collect data in your everyday life? When, why, and how?
   b. How much time did it take to build your survey and receive answers?
   c. Do you see any potential for using survey tools in your classroom?
   d. What are some other ways to collect data?
   e. What are some engaging ways we can teach students the difference between qualitative and quantitative data?
   f. Why is it important that students understand the concept of bias and know how to identify it?

5. Resources:
   a. The family activity Score! and the lesson Keep My Interest provide two different perspectives on collecting data. Saving Species and Can iAccess? are other great activities that focus on the computational thinking strategy of collecting data.
   b. Explore the U.S. Census Interactive Population Map and the Martian Impact Craters Project created by the Smithsonian Institute.
2. What is computational thinking?

The computational thinking modules continued...
Module: Analyzing Data

1. **Overview:** By analyzing data, we can identify connections among things, people and ideas. In this module, we’ll talk about analyzing different types of data across every academic discipline and how this computational thinking strategy helps our students understand, make meaning of, and even predict real-world outcomes. Predicting future trends is only possible with a thorough understanding of the present data.

2. **Pre-activity:** Identify a group of colleagues or friends that have a mutual shared interest, for example, a sports team, social issue or real estate. Use the website FiveThirtyEight to identify one trend or statistic related to this shared interest that you can track over time. Create a spreadsheet that you can share amongst this group so that each member has the ability to record statistics at different moments. After doing this over a number of weeks, you’ll have a dataset on the topic of interest.

3. **Post-activity:** Convene a meeting of your friends and colleagues in order to review the data. Determine the best way to represent this data in order to quickly spot trends.

4. **Discussion questions:**
   a. How does analyzing data relate to finding patterns?
   b. Sometimes the analysis of data seems slanted. Why do you think this happens?
   c. How could you use the computational thinking strategy of analyzing data in an English/Language Arts Classroom? What about a Physical Education class?

5. **Resources:**
   a. Build a Movement, Find Your Focus, and Outbreak are all lessons that emphasize analyzing data.
2. What is computational thinking?

The computational thinking modules continued...

Module: Finding Patterns

1. Overview: From natural science to computer science, we can find examples of patterns everywhere we look. In this module, we’ll see how the computational thinking strategy of finding patterns can help your students find the root cause of any given problem and plan for the future. Finding patterns empowers students to predicting future trends by looking for patterns in current data.

2. Pre-activity: Consider setting up a virtual Padlet board for collaboration as you learn more about finding patterns. Label one section “What I Know Now” and another section ”What I Want to Know”. Invite your fellow group members or colleagues to collaborate with you.

3. Post-activity: Return to your virtual Padlet board and dig into the responses you’ve collected.

4. Discussion questions:
   a. How can finding patterns help break down complex problems in our everyday lives? When have you used this strategy in your life?
   b. What opportunities do you see for using the computational thinking strategy of finding patterns in your classroom?
   c. How could the computational thinking strategy of finding patterns be a useful tool for teaching students with special needs?

5. Resources:
   a. Recognizing An Invader, Drop the Beat and the family activity Secret Code for Mazes are computational thinking activities developed around the strategy of finding patterns.
   b. Other resources include the U.S. Office of Disease Prevention, The Letter Writing in America Project and New Mexico Tech University’s Fractals Project.
2. What is computational thinking?

The computational thinking modules continued...

Module: Abstraction

1. Overview: The word abstract has many meanings, but in the world of computational thinking, abstraction is all about problem-solving. In this module we’ll discuss how the computational thinking strategy of abstraction gives students the skills to solve many problems with one solution.

2. Pre-activity: Share an image or a short text example of abstracting complexity that you have used in your own life with colleagues.

3. Post-activity: Moderate a discussion on how the computational thinking strategy of abstraction can make preparing Thanksgiving dinner or another holiday meal easier?

4. Discussion questions:
   a. What are some ways that abstraction applies to your discipline?
   b. How do you think you can bring the concept of abstraction to life in your classroom?
   c. Why is abstraction a useful tool for students to understand?

5. Resources:
   a. The Smell Test, Earth 2.0, Emoji Essay, and A New View of Color lessons focus on the computational thinking skill of abstraction.
   b. Other resources include STEM on Demand by Johns Hopkins University and the Library of Congress: America’s Story website.
2. What is computational thinking?

The computational thinking modules continued...

Module: Decomposition

1. Overview: When faced with big projects, it’s easy to get caught up in the details or put things off until the last minute. This module covers why the computational thinking strategy of decomposing problems is so important: it teaches our students that they can tackle any problem by breaking it down into manageable pieces.

2. Pre-activity: To explore decomposing problems in a different way, consider setting up an account on the 3D immersive environment Second Life and visiting Alice Academy (a virtual environment on Second Life).

3. Post-activity: Re-visit Alice Academy (a virtual environment on Second Life) or set up your own academy. Invite your colleagues to join you as you debrief decomposing problems.

4. Discussion questions:
   a. How have you decomposed a problem in your life this week?
   b. Could you explain to a child or teenager how to use the computational thinking strategy of decomposing problems in his or her own life?
   c. What opportunities do you envision in your own classroom for decomposing problems?

5. Resources:
   a. The lessons Destination Vacation, Recycle and Reuse, If You’re Happy and You Know It utilize the computational thinking strategy of decomposing problems.
   b. Other resources available include Math Riddles by Steve Miller, the Library of Congress: Author Webcasts, and The National Security Agency (NSA) Puzzle Periodical.
2. What is computational thinking?

The computational thinking modules continued...

Module: Building Models

1. **Overview:** Whether it’s a draft outline, a mathematical model, or a 3D prototype, the strategy of building models is all about visualizing data and highlighting problems before events happen in real life.

2. **Pre-activity:** Collect some good examples of models that students have created in your classroom. Post pictures of your models on a Pinterest board.

3. **Post-activity:** Return to your Pinterest board and think about using the strategy of building models to assess problems and make predictions as part of your next lesson.

4. **Discussion questions:**
   a. When was the last time you built a virtual or physical model to help visualize data in your own life?
   b. How does the teaching strategy of building models make solving complex problems easier?
   c. What specific lessons can you think of that would utilize this teaching strategy most effectively?

5. **Resources:**
   a. Drone Delivery and Transportation Tech are good examples of lessons that utilize the computational thinking strategy of building models.
   b. If You’re Happy And You Know It and the family activity Build a Hydro-Garden are modeling-based activities that students and families can work on together.
2. What is computational thinking?

The computational thinking modules continued...

Module: Developing Algorithms

1. **Overview:** From following a recipe to getting ready in the morning, we interact with algorithms every day. Algorithms are sets of step-by-step instructions on how to complete a given task. This module discusses how algorithms help humans think like computers by creating detailed sets of instructions that produce an intended result.

2. **Pre-activity:** As an introduction to developing algorithms, try a hashtag search. Type #algorithms into the search bar on a social media platform. You will be surprised at the amount of information you’ll find from fellow educators.

3. **Post-activity:** Following up on your hashtag search, consider joining a chat to discuss developing algorithms and computational thinking. Type the following into the twitter search bar to join one of these popular chats: #satchat, #ADEchat, #IgniteMyFuture, and #1to1ipadchat.

4. **Discussion questions:**
   a. When you hear the word “algorithm,” what comes to mind?
   b. How do you feel about the potential to teach your students using the ideas behind developing algorithms?
   c. What are some examples of algorithms that you have developed and use in your daily life?

5. **Resources:**
   a. Teachers and students can practice developing their own technical computer algorithms with Scratch, MIT’s block style coding platform that allows drag-and-drop programming, as explained on the Ignite My Future in School Curriculum Connectors.
   b. Digital DIY and Capable Cobot are examples of lessons that focus on developing algorithms.
3. Why does this matter for students?

a Outcomes

Bloom’s Taxonomy is a trusted resource for identifying student outcomes. The taxonomy begins with more concrete student objectives like memorization and advances to increasingly abstract learning methods. The Ignite My Future in School Computational Thinking curriculum is driven by the objectives of Bloom’s Taxonomy and will help your students develop the higher-order thinking skills that 21st century careers demand. Each lesson plan includes a “students will be able to...” section that clearly lists intended learning outcomes. The following are some examples of outcomes from Ignite My Future in School computational thinking curriculum:

1. Students will be able to...
   a. Create technical sketches of drones that can use algorithms to perform autonomously.
   b. Evaluate the potential social benefits of using an emoji-based written system of communication.
   c. Analyze open-source datasets.
   d. Apply data on interest rate fluctuation to specific case studies.
   e. Understand how computational thinking can help determine the veracity of facts.

The Ignite My Future in School computational thinking curriculum has been designed in keeping with the K-12 Computer Science Framework, a set of standards that includes seven specific outcomes for students studying computer science.

The Ignite My Future in School computational thinking curriculum aligns with national standards designed by the Common Core State Standards Initiative, Next Generation Science Standards, C3 Framework for Social Studies and others.

b Careers

Every Ignite My Future in School computational thinking lesson includes a list of 3-4 careers that are directly related to the skills and content of that lesson. Each lesson also includes a simulation that guides students through a real-life example that a professional might encounter. By participating in this curriculum, your students will explore a wide variety of careers.

Here are just a few of the career simulations in the Ignite My Future in School lesson plans:

1. The Smell Test (Science): Imagine that you are a team of chemists challenged to create a virtual reality program in which you incorporate everyday smells into the movie-going experience. To do this, you will first identify the building blocks (groups) of chemicals that produce scents. Then, you will hypothesize what combinations of these building blocks would produce certain results. If you are successful, viewers will have an incredibly realistic experience. If you are less accurate, viewers may experience some pretty unpleasant smells. Let’s see what you come up with!

2. Rewriting History (Social Studies): Imagine that you are a filmmaker creating a movie about the Civil Rights Movement [or other historical movement/era]. You want your film to depict society during this time as accurately as possible. However, you suspect that not all representations of the era that we see today are true to history. Your reputation as an artist is at stake and depends on how well you can recreate life during this time. Let’s see how well you do!
3. Why does this matter for students?

Careers continued...

3. A New View of Color (Art): Imagine you are an interior designer working for a commercial architecture firm that currently has four contracts to design and build the following spaces: a medical facility, a law office, a restaurant and a daycare center. Interior designers are responsible for much more than decorating a space. People react to things in the environment, and interior designers have the power to create environments that shape behaviors and promote health. You have been given the task to create a digital portfolio to present to a corporate client. The portfolio will include color, furnishings, and other materials that provide a functional and safe environment. Let’s see how well you do!

4. Keep My Interest (Math/Business): Imagine you are a financial advisor with a new client. This client is 18 years old and wants to buy a $2,500 computer to take to college. She has saved up some money from summer jobs, but not enough to buy the computer outright. Your client wants to buy the computer using a credit card, but she has never had or used a credit card before, and she is concerned about diving into the world of credit card interest and debt. She has heard that people can get into financial trouble because of interest payments and she wants to avoid this. What advice would you give your client?

5. True or False (English/Language Arts): Bigfoot, the Loch Ness Monster, the New Jersey Devil—these fascinating legends have one thing in common: they haven’t been proved! Imagine you are an investigative journalist. You have been assigned a story about the truth behind several popular legends. Your editor warns you that the story must be accurate or you could lose your job. How will you go about separating fact from fiction?

The Ignite My Future in School website also features engaging career vignettes that show how a variety of professionals use computational thinking in their jobs. The wide range of career information listed throughout these computational thinking resources highlights how the seven computational thinking strategies are applicable to every profession.

Real-world problem solving skills

The magic of computational thinking lies in taking the strategies that computers use to solve problems and applying them to the real world. These are just a few of the real-world questions designed to ignite curiosity in this computational thinking curriculum:

1. Can technology speed up the process of learning a new language?
2. How can we take care of our friends and family members as they age?
3. How can we determine whether a planet is enough like Earth for us to live on?
4. Can computers help us put an end to cyberbullying?
5. What role do the arts play in improving the world?

Each computational thinking lesson culminates with a reflection that encourages students to think about how the subject matter of the lesson relates directly to their lives. In this reflection students consider the following questions:

1. How do this problem and solution connect to me?
2. How do this problem and solution connect to real-world careers?
3. How do this problem and solution connect to our world?

Every lesson also includes clear explanations for how its contents connect to students, careers and to the real world. These tools will help your students take the skills they learn in the classroom and apply them to situations they encounter in other areas of their lives.
3. Why does this matter for students?

Meeting students where they are: cultural responsiveness in the classroom

How does culturally responsive teaching relate to teaching computational thinking?

1. Culturally responsive teaching is instruction that takes into account how students learn in relation to their cultural or family history. Consider the following questions:
   a. How does learning differ by individual, home, family and culture?
   b. What is the role and value of tradition in the classroom?
   c. How can social interaction differ from student to student?
   d. What is the value of questions and scenarios that represent diverse individuals?
   e. How can we create a classroom where different points of view can be shared at the same time?
   f. What can we gain by learning about different cultures?

2. Computational thinking supports cultural responsive teaching by giving students strategies they can use to better understand and make sense of the world around them.

How can you create a culturally responsive classroom?

1. Connect lesson content directly to students’ lives
2. Use a variety of languages and cultural references when providing examples
3. Utilize props and visual aids in order to ensure understanding among students from all linguistic backgrounds
4. Lead frequent team-building activities to encourage collaboration
5. Provide individual instruction for students of different achievement levels
6. Collect responses equally from high-achieving and low-achieving students to create an environment where all student perspectives are valued
7. Ensure that examples, multimedia resources and classroom tools feature a wide diversity of racial, ethnic, gender and other forms of diversity
8. Diversify student groups
9. Give feedback in a variety of different formats (verbal, written, etc.)
10. Show how differences among students make for better learning
11. Allow various options for completing assignments
12. Bridge cultural differences through effective communication
13. Talk to students about differences between individuals
14. Attend and discuss community events with students
15. Model positive and respectful self-talk and talk about others

3. Computational thinking lessons that encourage culturally responsive teaching:

1. Celebrate Together (ELA)
2. Can iAccess? (Social Studies)
3. Destination Vacation (Math)
4. Healthy Eating is for Everyone (Science/Health)
3. Why does this matter for students?

Computational thinking for students with individual education plans (IEPs)

The seven computational thinking strategies align with recommended techniques for instructing students on individual education plans (IEPs). Use the following approaches to support a wide range of learners in your classroom:

1. Utilize graphic organizers (abstract)
2. Break down assignments and projects into manageable pieces (decompose)
3. Check in with students often to assess learning and adjust your approach as necessary (collect data)
4. Use formative assessments to measure student comprehension (analyze data)
5. Provide detailed examples and hands-on learning opportunities (build models)
6. Group students by shared interest and ability (find patterns)
7. Give specific step-by-step instructions (develop algorithms)

Computational thinking lessons that encourage differentiated instruction:

1. Find Your Focus (Engineering/Math)
2. Emoji Essay (ELA)
3. Smile Starter (Science)
4. Drop the Beat (Music)

Computational thinking for English-language learners

Here are some of the ways the Ignite My Future in School computational thinking curriculum is ideal for instructing English-language learning (ELL) students:

1. The Ignite My Future In School Lessons use academic language and define key computational thinking terms.
2. The Ignite My Future in School family activities are a great way to engage family and community members in the learning process.
3. Resources are designed to be applicable to students from many backgrounds.
4. Computational thinking strategies like finding patterns, abstraction and decomposition are helpful practices for English-language learners.

Computational thinking lessons that are ideal for English-language learners:

5. Emoji Essay (ELA)
6. Speak My Language (ELA)
7. Celebrate Together (ELA)
8. Abbrevs (ELA)
4. Who should teach computational thinking?

Computational thinking is applicable to every discipline and every classroom. Computational thinking strategies are strong supplements to your curriculum and are supported by a variety of national standards.

**Common Core State Standards Initiative**

1. **English/Language Arts**
   a. **Reading**
      Science and Technical Subjects
      1. Integration of Knowledge and Ideas (CCSS.ELA-LITERACY.RST.6-8.7, CCSS.ELA-LITERACY.RST.6-8.8, CCSS.ELA-LITERACY.RST.6-8.9)
      2. Key Ideas and Details (CCSS.ELA-LITERACY.RST.6-8.1, CCSS.ELA-LITERACY.RST.6-8.2, CCSS.ELA-LITERACY.RST.6-8.3)
      Literature
      1. Integration of Knowledge and Ideas (CCSS.ELA-LITERACY.RL.8.7, CCSS.ELA-LITERACY.RL.8.9)
      2. Key Ideas and Details (CCSS.ELA-LITERACY.RL.8.1, CCSS.ELA-LITERACY.RL.8.2, CCSS.ELA-LITERACY.RL.8.3)
   b. **Speaking and Listening**
      Comprehension and Collaboration (CCSS.ELA-LITERACY.SL.8.1, CCSS.ELA-LITERACY.SL.8.2, CCSS.ELA-LITERACY.SL.8.3)
      Presentation of Knowledge and Ideas (CCSS.ELA-LITERACY.SL.8.4, CCSS.ELA-LITERACY.SL.8.5, CCSS.ELA-LITERACY.SL.8.6)
   c. **Writing**
      Text Types and Purposes (CCSS.ELA-LITERACY.WHST.6-8.1, CCSS.ELA-LITERACY.WHST.6-8.2)
      Production and Distribution of Writing (CCSS.ELA-LITERACY.WHST.6-8.4, CCSS.ELA-LITERACY.WHST.6-8.5, CCSS.ELA-LITERACY.WHST.6-8.6)
      Research to Build and Present Knowledge (CCSS.ELA-LITERACY.WHST.6-8.7, CCSS.ELA-LITERACY.WHST.6-8.8, CCSS.ELA-LITERACY.WHST.6-8.9)

2. **Math**
   a. **The Number System** (CCSS.MATH.CONTENT.7.NS.A.1, CCSS.MATH.CONTENT.7.NS.A.2, CCSS.MATH.CONTENT.7.NS.A.3)
   b. **Ratios & Proportional Relationships** (CCSS.MATH.CONTENT.7.RP.A.1, CCSS.MATH.CONTENT.7.RP.A.2, CCSS.MATH.CONTENT.7.RP.A.3)
   c. **Expressions & Equations** (CCSS.MATH.CONTENT.8.EE.A.4, CCSS.MATH.CONTENT.8.EE.B.5, CCSS.MATH.CONTENT.8.EE.C.7)
   d. **Functions** (CCSS.MATH.CONTENT.8.F.A.1, CCSS.MATH.CONTENT.8.F.B.4, CCSS.MATH.CONTENT.8.F.B.5)
   f. **Statistics & Probability** (CCSS.MATH.CONTENT.8.SP.A.1, CCSS.MATH.CONTENT.8.SP.A.3, CCSS.MATH.CONTENT.8.SP.A.4)
4. Who should teach computational thinking?

National Standards continued...

Next Generation Science Standards

1. Physical Sciences
   a. Matter and Energy in Organisms and Ecosystems
   b. Matter and Its Interactions

2. Life Sciences
   a. Interactions, Energy and Dynamics
   b. Ecosystems
     Interdependent Relationships in Ecosystems
   c. Heredity, Inheritance and Variation of Traits
   d. Biological Evolution: Unity and Diversity
   e. Natural Selection and Adaptation

3. Earth and Space Sciences
   a. Earth’s Place in the Universe
   b. Earth and Human Activity

4. Engineering, Technology and Applications of Science
   a. Engineering Design

College, Career and Civic Life (C3) Framework for Social Studies

Computational thinking in elective courses

1. Health and Physical Education
2. Arts
3. World Languages

Additional standards and guidelines

1. K-12 Computer Science Framework
   a. Core Practices, Including Computational Thinking
      Fostering an Inclusive Computing Culture
      Collaborating Around Computing
      Recognizing and Defining Computational Problems
      Developing and Using Abstractions
      Creating Computational Artifacts
      Testing and Refining Computational Artifacts
      Communicating About Computing
5. Where should I begin?

**Helpful resources**

**Computational thinking resources:**
1. Don’t Just Learn to Code, Learn How to Think Like a Computer Scientist
2. Computational Thinking Across the Curriculum
3. Don’t Learn to Code. Learn to Think
4. How To Teach Computational Thinking
5. Exploring Computational Thinking
6. How To Teach Computational Thinking
7. Teaching Computational Thinking: First Step Bridging the Stem Skills Gap

**Resources for culturally responsive teaching:**
1. Culturally Responsive Teaching Strategies
2. A Framework for Culturally Responsive Teaching
3. Principles for Culturally Responsive Teaching

**Resources for differentiating instruction:**
1. Teaching Diverse Learners
2. Differentiation: Lessons From Master Teachers
3. Let Me Learn My Own Way
4. Differentiated Instruction: Maximizing the Learning of All Students

**Tech tools:**
1. Aurasma (augmented reality application)
2. Desmos (graphing and plotting tool)
3. Google Sketch-Up (CAD software)
4. Hopscotch (coding resource)
5. Instructables (tutorial and how-to site)
6. Made with Code (coding resource)
7. Poll Everywhere (data aggregation tool)
8. Quizlet (online quiz generator)
9. Scratch (coding and programming resource)
10. SurveyMonkey (survey tool)
11. ThingLink (image annotator)
12. TinkerCAD (CAD software)
13. Thingiverse (3D printing library)
14. Wemogee (emoji generator)
5. Where should I begin?

b How to gain confidence with computational thinking

Professional Learning Communities and Networks (PLCs and PLNs)

1. A professional learning community is a way to connect with your fellow educators while learning new material. Usually, members of your PLC are colleagues with whom you work with directly on a daily basis. Meeting with members of your PLC while exploring the modules can lead to brainstorming improved ideas and ways to use computational thinking practices in your classroom. A professional learning network (PLN) tends to include a wider range of members and communication usually takes place online rather than face-to-face. As you organize your PLC or PLN, keep in mind these starting ideas:

   a. Your principal or supervisor might allocate school resources or school time to work and learn when several educators are involved.

   b. Set a weekly or monthly time to meet and limit the time of each session. Limit the total number of sessions that you will devote to your goal to four or six.

   c. Before the first session, brainstorm what you think computational thinking is and how you might use or have used computational thinking in your life.

   d. At your first session, discuss how you think you might be able to use the material in your classroom in the next few weeks. Refer back to your ideas often to revise.

   e. Commit to one another that you will leave your four- to six-week session with a ready-to-use lesson plan. If you get stuck on your own lesson plan, don’t hesitate to ask group members for help or help others with their plans. The goal of a PLC or PLN is collaboration.

   f. Be sure to end your final discussion session with a conversation about how your can use computational thinking in your life in addition to in your lesson plan. Making the ideas relevant to your life will provide additional examples for your students and cement the ways computational thinking works.

   g. Consider using time-saving ways to communicate in addition to in-person meetings. Email, phone, messaging services, and other traditional resources are always good alternatives to in-person meetings.

Reflective Practices

1. It is important to chart your progress with teaching computational thinking. The following reflective strategies can help you assess and improve your ability to teach computational thinking to your students:

   a. Reflect with grade-alike and subject colleagues to discuss how students are responding to computational thinking material in different classrooms.

   b. After a lesson, take a few moments to write responses to the following questions. Track your answers to the questions over time to chart your improvement.

      What worked well in that lesson?

      What could I improve upon?

      How do I know what worked well and what didn’t in this lesson?

      Do I have any limiting beliefs about my students, my instruction or the material?

      If so, how can I change them?

Ignite My Future in School program

1. Utilize the free resources available at the Ignite My Future in School website, including webisodes, career vignettes, model lessons and curriculum connectors.

2. Consider recommending the Ignite My Future in School TECHademy program to your administrators to receive hands-on guidance and mentorship on computational thinking.