CURRICULUM ALIGNMENT GUIDE

ISTE STANDARDS

FOR EDUCATORS

Computational Thinking Competencies

100 Ideas for Secondary Teachers:

Outstanding Computing Lessons

INTRODUCTION

100 ideas: Outstanding Computing Lessons is a collection of 100 practical, tried-and-tested ideas for teaching computing. It is aimed at computing / ICT teachers of all levels, whether specialist or non-specialist, newly qualified or experienced.

For more information on 100 Ideas: Outstanding Computing Lessons and to find additional education resources and supporting materials, including more than 50 worksheets to accompany the activities in the book, visit: teachwithict.com/100ideas

10 sample activities can be downloaded for free at teachwithict.com/bonus

1. COMPUTATIONAL THINKING (LEARNER)

Educators continually improve their practice by developing an understanding of computational thinking and its application as a cross-curricular skill. Educators develop a working knowledge of core components of computational thinking: such as decomposition; gathering and analyzing data; abstraction; algorithm design; and how computing impacts people and society. Educators:

STANDARD	DESCRIPTION	ACTIVITY
1a	Set professional learning goals to explore and apply teaching strategies for integrating CT practices into learning activities in ways that enhance student learning of both the academic discipline and CS concepts.	• Idea 42: Making the tea algorithm • Idea 43: Teaching with magic • Idea 44: Crazy characters • Idea 45: Puzzle me • Idea 46: Human robot • Idea 47: A-maze-ing algorithms • Idea 48: 20 questions • Idea 49: Breaking the code • Idea 50: Origami algorithms • Idea 51: Guess the object
1b	Learn to recognize where and how computation can be used to enrich data or content to solve discipline-specific problems and be able to connect these opportunities to foundational CT practices and CS concepts.	• Idea 14: Contextualizing learning • Idea 42: Making the tea algorithm
1c	Leverage CT and CS experts, resources and professional learning networks to continuously improve practice integrating CT across content areas.	• Idea 110: Team teaching
1e	Recognize how computing and society interact to create opportunities, inequities, responsibilities and threats for individuals and organizations.	• Idea 23: Fake News • Idea 25: Copy cat • Idea 27: Fakebook • Idea 33: Wayback machine • Idea 34: What a waste • Idea 39: Internet of things • Idea 41: Moral machine

3. COLLABORATING AROUND COMPUTING (COLLABORATOR)

Effective collaboration around computing requires educators to incorporate diverse perspectives and unique skills when developing student learning opportunities, and recognize that collaboration skills must be explicitly taught in order to lead to better outcomes than individuals working independently. Educators work together to select tools and design activities and environments that facilitate these collaborations and outcomes. Educators:

STANDARD	DESCRIPTION	ACTIVITY
3a	Model and learn with students how to formulate computational solutions to problems and how to give and receive actionable feedback.	• Idea 2: Rubber duck debugging • Idea 5: PRIMM • Idea 6: Parsons problems • Idea 7: Use-modify-create • Idea 17: Peer instruction • Idea 43: Teaching with magic • Idea 107: Code tracing • Idea 109: Worked examples
3b	Apply effective teaching strategies to support student collaboration around computing, including pair programming, working in varying team roles, equitable workload distribution and project management.	• Idea 1: Paired programming • Idea 15: Go unplugged • Idea 16: Socratic debate • Idea 17: Peer instruction • Idea 20: Escape rooms • Idea 32: Dragon's Den • Idea 44: Crazy characters • Idea 46: Human robot • Idea 70: Round-robin revision • Idea 78: Revision speed dating • Idea 106: Data science detectives
3c	Plan collaboratively with other educators to create learning activities that cross disciplines to strengthen student understanding of CT and CS concepts and transfer application of knowledge in new contexts.	• Idea 110: Team teaching

4. CREATIVITY & DESIGN (DESIGNER)

Computational thinking skills can empower students to create computational artifacts that allow for personal expression. Educators recognize that design and creativity can encourage a growth mindset and work to create meaningful CS learning experiences and environments that inspire students to build their skills and confidence around computing in ways that reflect their interests and experiences. Educators:

STANDARD	DESCRIPTION	ACTIVITY
4a	Design CT activities where data can be obtained, analyzed and represented to support problem-solving and learning in other content areas.	• Idea 26: Mario Kart ™ spreadsheets • Idea 28: Database detectives • Idea 31: Infographics • Idea 36: Storage Top Trumps® • Idea 39: Internet of things • Idea 98: Coding the weather • Idea 106: Data science detectives • Idea 108: Make me happy (AI sentiment analysis)
4b	Design authentic learning activities that ask students to leverage a design process to solve problems with awareness of technical and human constraints and defend their design choices.	• Idea 31: Infographics • Idea 32: Dragon's Den • Idea 34: What a waste! • Idea 82: Chatting robot • Idea 99: Rubbish robots
4c	Guide students on the importance of diverse perspectives and human-centered design in developing computational artifacts with broad accessibility and usability.	• Idea 19: Using QR codes • Idea 30: Videography • Idea 31: Infographics • Idea 73: PechaKucha
4d	Create CS and CT learning environments that value and encourage varied viewpoints, student agency, creativity, engagement, joy and fun.	• Idea 16: Socratic debate • Idea 17: Peer instruction • Idea 18: Game-based learning • Idea 20: Escape rooms

5. INTEGRATING COMPUTATIONAL THINKING (FACILITATOR)

Educators facilitate learning by integrating computational thinking practices into the classroom. Since computational thinking is a foundational skill, educators develop every student's ability to recognize opportunities to apply computational thinking in their environment. Educators:

STANDARD	DESCRIPTION	ACTIVITY
5a	Evaluate and use CS and CT curricula, resources and tools that account for learner variability to meet the needs of all students.	Strategies for reducing cognitive load: • Idea 1: Paired programming • Idea 5: PRIMM • Idea 6: Parsons problems • Idea 7: Use-modify-create • Idea 17: Peer instruction • Idea 107: Code tracing • Idea 109: Worked examples
5b	Empower students to select personally meaningful computational projects.	• Idea 14: Contextualizing learning • Idea 30: Videography • Idea 32: Dragon's Den
5c	Use a variety of instructional approaches to help students frame problems in ways that can be represented as computational steps or algorithms to be performed by a computer.	• Idea 2: Rubber duck debugging • Idea 5: PRIMM • Idea 6: Parsons problems • Idea 7: Use-modify-create • Idea 43: Making the tea algorithm • Idea 44: Crazy characters • Idea 47: A-maze-ing algorithms • Idea 50: Origami algorithms • Idea 107: Code tracing • Idea 109: Worked examples