# CURRICULUM ALIGNMENT GUIDE

## ISTE STANDARDS

## FOR COMPUTER SCIENCE EDUCATORS

# 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 free 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. KNOWLEDGE OF CONTENT

Computer science educators demonstrate knowledge of computer science content and model important principles and concepts.

| STANDARD | DESCRIPTION | ACTIVITY |
|---|---|---|
| 1a | Demonstrate knowledge of and proficiency in data representation and abstraction<br>i. Effectively use primitive data types<br>ii. Demonstrate an understanding of static and dynamic data structures<br>iii. Effectively use, manipulate and explain various external data stores: various types (text, images, sound, etc.), various locations (local, server, cloud), etc.<br>iv. Effectively use modeling and simulation to solve real-world problems | • Idea 52: Image compression<br>• Idea 58: Binary representation of images (unplugged)<br>• Idea 62: Binary addition<br>• Idea 63: Binary numbers<br>• Idea 64: Binary representation of images<br>• Idea 65: Binary representation of sound<br>• Idea 66: Binary bingo<br>• Idea 67: It's all about hex<br>• Idea 68: ASCII 'secret' agent |
| 1b | Effectively design, develop, and test algorithms<br>i. Using a modern, high-level programming language, construct correctly functioning programs involving simple and structured data types; compound boolean expressions; and sequential, conditional, and iterative control structures<br>ii. Design and test algorithms and programming solutions to problems in different contexts (textual, numeric, graphic, etc.) using advanced data structures<br>iii. Analyze algorithms by considering complexity, efficiency, aesthetics and correctness<br>iv. Demonstrate knowledge of two or more programming paradigms<br>v. Effectively use two or more development environments<br>vi. Demonstrate knowledge of varied software development models and project management strategies | • Idea 80: Magic 8-ball®<br>• Idea 81: Shakespearean insult generator<br>• Idea 82: Chatting robot<br>• Idea 83: Just dance<br>• Idea 84: Adventures in text<br>• Idea 85: Mad Libs®<br>• Idea 86: Sorting Hat<br>• Idea 87: Turtle power (a lesson using PRIMM)<br>• Idea 88: Guess my number<br>• Idea 89: Mind-reading algorithm<br>• Idea 90: Cat and mouse<br>• Idea 91: Reaction timer<br>• Idea 93: Lights, camera, action<br>• Idea 94: Making music<br>• Idea 95: Coding probability<br>• Idea 96: Physical computing<br>• Idea 97: Turtle snowflakes<br>• Idea 98: Coding the weather<br>• Idea 101: What's your elf name?<br>• Idea 102: Guess / _ _e / word<br>• Idea 103: Cards against humanities<br>• Idea 104: Shakespearean complement generator<br>• Idea 105: Hacking the news<br>• Idea 106: Make me happy |

| | | |
|---|---|---|
| 1c | Demonstrate knowledge of digital devices, systems and networks<br>i. Demonstrate an understanding of data representation at the machine level<br>ii. Demonstrate an understanding of machine-level components and related issues of complexity<br>iii. Demonstrate an understanding of operating systems and networking in a structured computer system<br>iv. Demonstrate an understanding of the operation of computer networks and mobile computing devices | • Idea 33: Little Man Computer<br>• Idea 35: Role reversal (Operating systems)<br>• Idea 54: World Wide Web unplugged<br>• Idea 59: How computers work<br>• Idea 60: Memory unplugged<br>• Idea 61: Network topologies unplugged |
| 1d | Demonstrate an understanding of the role computer science plays and its impact in the modern world<br>i. Demonstrate an understanding of the social, ethical, and legal issues and impacts of computing, and attendant responsibilities of computer scientists and users<br>ii. Analyze the contributions of computer science to current and future innovations in sciences, humanities, the arts and commerce | • Idea 16: Socratic debate<br>• Idea 25: Copycat<br>• Idea 34: What a waste!<br>• Idea 39: Internet of Things (IoT)<br>• Idea 41: Moral machine (AI)<br>• Idea 55: Intelligent piece of paper (AI)<br>• Idea 108: Make me happy (AI) |

## 2. EFFECTIVE TEACHING AND LEARNING STRATEGIES

Computer science educators demonstrate effective content pedagogical strategies that make the discipline comprehensible to students.

| STANDARD | DESCRIPTION | ACTIVITY |
|---|---|---|
| 2a | Plan and teach computer science lessons/units using effective and engaging practices and methodologies<br>i. Select a variety of real-world computing problems and project-based methodologies that support active and authentic learning and provide opportunities for creative and innovative thinking and problem solving<br>ii. Demonstrate the use of a variety of collaborative groupings in lesson plans/units and assessments<br>iii. Design activities that require students to effectively describe computing artifacts and communicate results using multiple forms of media<br>iv. Develop lessons and methods that engage and empower learners from diverse cultural and linguistic backgrounds<br>v. Identify problematic concepts and constructs in computer science and appropriate strategies to address them<br>vi. Design and implement developmentally appropriate learning opportunities supporting the diverse needs of all learners<br>vii. Create and implement multiple forms of assessment and use resulting data to capture student learning, provide remediation and shape classroom instruction | • Idea 3: Code golf<br>• Idea 8: Hour of code<br>• Idea 9: Code bug<br>• Idea 10: Code combat<br>• Idea 19: Using QR codes<br>• Idea 53: Bubble sort dance algorithm<br>• Idea 56: Envelope variables<br>• Idea 57: Card sort<br>• Idea 72: PEE (Point, Evidence, Explain)<br>• Idea 75: Command word bingo<br>• Idea 76: Bug hunt<br>• Idea 79: Match IT |

## 3. EFFECTIVE LEARNING ENVIRONMENTS

Computer science educators apply their knowledge of learning environments by creating and maintaining safe, ethical, supportive, fair and effective learning environments for all students.

| STANDARD | DESCRIPTION | ACTIVITY |
|---|---|---|
| 3a | Design environments that promote effective teaching and learning in computer science classrooms and online learning environments and promote digital citizenship<br>i. Promote and model the safe and effective use of computer hardware, software, peripherals and networks<br>ii. Plan for equitable and accessible classroom, lab and online environments that support effective and engaging learning | • Idea 23: Fake News<br>• Idea 25: Copycat<br>• Idea 27: Fakebook<br>• Idea 29: Did you meme it?<br>• Idea 33: Wayback machine |

## 4. EFFECTIVE PROFESSIONAL KNOWLEDGE AND SKILLS

Computer science educators demonstrate professional knowledge and skills in their field and readiness to apply them.

| STANDARD | DESCRIPTION | ACTIVITY |
|---|---|---|
| 4a | Participate in, promote and model ongoing professional development and lifelong learning relative to computer science and computer science education<br>i. Identify and participate in professional computer science and computer science education societies, organizations and groups that provide professional growth opportunities and resources<br>ii. Demonstrate knowledge of evolving social and research issues relating to computer science and computer science education<br>iii. Identify local, state, and national content and professional standards and requirements affecting the teaching of secondary computer science | • Idea 1: Paired programming<br>• Idea 4: Game design<br>• Idea 5: PRIMM<br>• Idea 6: Parsons problems<br>• Idea 7: Use-modify-create<br>• Idea 13: DART<br>• Idea 17: Peer instruction<br>• Idea 22: Flipped learning<br>• Idea 107: Code tracing<br>• Idea 109: Worked examples |