CURRICULUM ALIGNMENT GUIDE

CSTA K–12 CS Standards

OVERVIEW

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

PART 1: PROGRAMMING STRATEGIES

IDEA	DESCRIPTION	STANDARDS
001	**Paired programming** – A research-driven coding strategy for helping novice learners.	1B-AP-11, 1B-AP-16, 2-AP-15.
002	**Rubber duck debugging** – A programming strategy used to help students find bugs and in their code.	1B-AP-11, 1B-AP-16, 2-AP-15.
003	**Code golf** – A programming strategy to help students create more efficient code.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.
004	**Game design** – Using games as a hook to encourage students to learn how to code.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.
005	**PRIMM** – A research-based approach to teaching coding and for reducing cognitive load.	1B-AP-11, 1B-AP-13, 2-AP-15.
006	**Parsons problems** – Help students learn how to code by removing some of the barriers.	1B-AP-11, 1B-AP-12, 1A-AP-14, 2-AP-15.
007	**Use-modify-create** – Reduce anxiety while supporting growth with this simple three-stage approach to learning to code.	1B-AP-11, 1B-AP-13, 2-AP-15.
008	**Hour of Code** – Give your students a 'byte'-sized introduction to computer science with an hour of code.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.
009	**Code bug** – Build resilience and reduce anxiety when teaching children to code by purposely introducing 'bugs' early on in the learning process.	1A-AP-14.
010	**Code combat** – Put your students' coding skills to the test by pitting them against each other in code combat!	1B-AP-11, 1B-AP-12, 2-AP-15.
011	**Teaching with robots** – Coding can often be difficult for students to grasp. Robots can provide a simpler, more tangible introduction to programming.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.

PART 2: COMPUTING STRATEGIES

IDEA	DESCRIPTION	STANDARDS
012	**Take your screwdrivers to work** – Students explore how computers work by taking old devices apart.	1B-CS-01, 1B-CS-02.
013	**DART your students** – A strategy designed to improve students' reading comprehension.	
014	**Contextualise learning** – Explore strategies for making computing relevant and provide 'real-life' learning experiences for students.	
015	**Go unplugged** – Teaching computing without computers.	
016	**Socratic debate** – Students debate the social, ethical, and legal issues surrounding the use of computers.	3A-NI-05, 3A-NI-06, 3A-NI-07, 3A-NI-08, 3A-IC-24, 3A-IC-29.
017	**Peer instruction** – A research-driven approach to teaching difficult concepts that students often misunderstand.	
018	**Game-based learning** – Exploring the use of games, such as Minecraft: Education Edition, to teach children how to code.	
019	**Using QR codes** – Using QR codes to teach computing theory.	
020	**Escape rooms** – Students must solve a series of binary puzzles to open physical locks and stop a simulated 'virus attack'.	1A-NI-04, 1B-NI-05, 2-NI-05.
021	**Blogs and wikis** – Using blogs and wikis to teach computing theory.	1B-IC-20.
022	**Flipped learning** – Reversing the traditional way of teaching to make better use of classroom time.	
023	**Guided discovery** – An inductive approach to teaching and learning where students take an active role in discovering knowledge and developing understanding for themselves.	

PART 3: ICT AND DIGITAL LITERACY

IDEA	DESCRIPTION	STANDARDS
024	**Fake news** – Students learn how to identify 'fake news' articles before creating their own fake news story.	1A-IC-17.
025	**Copycat** – A fun activity that teaches students about copyright, public domain, fair use, and Creative Commons.	1B-IC-21.
026	**Mario Kart ™ spreadsheets** – An example of how to use game-based learning to teach students essential spreadsheet skills.	1B-DA-06, 2-DA-08.
027	**Fakebook** – An 'escape room' challenge which helps students understand the importance of protecting their online presence.	1A-IC-17, 1A-IC-18, 1B-NI-05, 2-NI-05.
028	**Database detectives** – Students test their sleuthing skills with this 'whodunnit' themed database challenge.	1A-DA-05, 1B-DA-06, 1B-DA-07, 2-DA-09.
029	**Did you meme it?** – Students explore the purpose and ethics of memes before creating their own meme on an agreed topic.	1A-IC-17.
030	**Videography** – Students create a YouTube-style instructional video whilst also explore the importance of concise instructions (algorithms).	
031	**Infographics** – Students create infographics about their mobile phone habits.	1B-DA-06, 2-DA-08, 2-DA-09, 3A-DA-11, 3A-DA-12.
032	**Dragon's Den** – Students work as a team to design an innovative solution to a global problem.	1B-IC-19, 2-CS-02, 2-AP-18.
033	**Wayback Machine** – Students learn about their digital footprint and the long-lasting impact of their online actions.	1A-IC-17

PART 4: COMPUTING ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
034	**What a waste** – Students, working in teams, explore innovative ways to reduce e-waste.	1B-DA-06, 1B-IC-18, 2-AP-18.
035	**Role reversal** – Students take on the role of a teacher and explore different types of software (system, application, utility).	1B-CS-02, 3A-CS-02.
036	**Storage Top Trumps®** – Students explore different storage devices before creating a game of Top Trumps® based on what they have learned.	1B-CS-02.
037	**Little Man Computer** – Students explore 'Little Man Computer' – a simulator that models the basic features of a modern computer that uses Von Neumann architecture.	1B-CS-02.
038	**Features of a CPU (a lesson using DART)** – Students explore the main features of a CPU.	1B-CS-02.
039	**Internet of things** – Students design a 'smart home' that utilises the internet of things.	2-CS-02, 2-DA-08.
040	**The great input/output QR hunt** – Students complete a QR hunt to discover facts about different input and output devices.	1B-CS-01.
041	**Moral machine** – Students explore the ethics behind creating AI for self-driving vehicles.	1B-IC-18, 2-IC-21, 3A-IC-24.

PART 5: COMPUTATIONAL THINKING

IDEA	DESCRIPTION	STANDARDS
042	**Making the tea algorithm** – Students explore the importance of creating precise instructions by creating an algorithm for making a cup of tea / coffee.	1B-AP-11, 2-AP-10, 2-AP-13, 2-AP-15.
043	**Teaching with magic** – Using magic to teach computational thinking skills.	1B-AP-11, 2-AP-13, 2-AP-15.
044	**Crazy characters** – Students write an algorithm for drawing a monster.	1B-AP-08, 1B-AP-11, 2-AP-15.
045	**Puzzle me** – Using puzzles to practise computational thinking skills (decomposition, pattern-matching, abstraction and algorithm design).	1B-AP-11, 2-AP-13.
046	**Human robot** – Exploring algorithms through physical activities such as movement and dance.	1B-AP-11, 2-AP-13.
047	**A-maze-ing algorithms** – Students explore the importance of clear and precise instructions by writing algorithms to solve a simple maze.	1B-AP-11, 2-AP-13, 2-AP-15.
048	**20 questions** – Students explore the efficiency of different search algorithms by playing a game of '20 questions'.	1B-AP-11, 2-AP-13.
049	**Breaking the code** – Students develop their problem-solving skills with a series of code-breaking challenges.	1B-AP-11, 2-AP-13, 2-NI-06
050	**Origami algorithms** – Students write algorithms for folding a paper aeroplane or origami animal.	1B-AP-11, 2-AP-13, 2-AP-15.
051	**Guess the object** – Getting students to model, draw or mime a variety of different objects can help them to understand the concept of abstraction.	1B-AP-11, 2-AP-13.

PART 6: UNPLUGGED ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
052	**Image compression** – Students learn about lossless compression without the use of a computer.	2-DA-07, 3A-DA-09.
053	**Bubble sort dance algorithm** – Students learn how a bubble sort algorithm works is via the medium of Hungarian folk dance!	1B-DA-06, 3B-AP-10, 3B-AP-11.
054	**World Wide Web unplugged** – Students role-play what happens when a user enters an address in a web browser.	1B-NI-04, 2-NI-04.
055	**Intelligent piece of paper (AI)** – Exploring artificial intelligence (AI) with a game of Tic-Tac-Toe.	3B-AP-08, 3B-AP-09.
056	**Envelope variables** – Demonstrate a simple program that uses variables and assignment by running them on a computer made entirely out of students.	1A-AP-09, 1B-AP-09.
057	**Card sort** – Students explore three common sorting algorithms (bubble, merge, and insertion) by sorting playing cards.	1B-DA-06, 3B-AP-10, 3B-AP-11.
058	**Binary representation of images (unplugged)** – Students explore how a computer represents images using binary.	2-DA-07, 3A-DA-09.
059	**How computers work** – Students take on the role of various parts of a computer and simulate the running of a simple program.	1B-CS-02.
060	**Memory unplugged** – Students explore how data is transferred between different storage locations inside a computer, such as RAM, cache memory, secondary storage and virtual memory.	1B-CS-02.
061	**Network topologies** – Using string and various other household objects, students simulate the three most common network topologies.	2-NI-04, 3A-NI-04.

PART 7: DATA REPRESENTATION

IDEA	DESCRIPTION	STANDARDS
062	**Binary addition** – Students learn how to add two numbers using binary.	2-DA-07, 3A-DA-09.
063	**Binary numbers** – Students learn about binary.	2-DA-07, 3A-DA-09.
064	**Binary representation of images** – Students explore how a computer represents images using binary.	2-DA-07, 3A-DA-09.
065	**Binary representation of sound** – Students explore how a computer represents sound using binary.	2-DA-07, 3A-DA-09.
066	**Binary bingo** – A fun strategy to test students' understanding of binary representation of numbers.	2-DA-07, 3A-DA-09.
067	**It's all about hex** – Students learn about the hex numbering system.	2-DA-07, 3A-DA-09.
068	**ASCII 'secret' agent** – Students explore how a computer represents text using binary by solving (and creating) as series of coded messages.	2-DA-07, 3A-DA-09.

PART 8: EXAM PREPARATION

IDEA	DESCRIPTION	STANDARDS
069	**Padlet** – Using online curation tools, such as Padlet, to collating resources in preparation for exams.	
070	**Round-robin revision** – Make revision fun and engaging with a series of mini games.	
071	**Revision podcasts** – Create revision resources that students can listen to anytime, anywhere!	
072	**PEE (point, evidence, explain)** – A simple strategy to help improve the quality of written answers to exam questions.	
073	**PechaKucha** – A great way to encourage students to be more concise and a little more creative with their presentations.	
074	**Sketch-noting** – A great way to empower students and allow them to synthesise information visually.	
075	**Command word bingo** – A simple starter activity that will pay dividends at exam time!	
076	**BUG hunt** – A technique for helping students understand thoroughly what is expected of them during exams.	
077	**Tweet IT** – A fun revision strategy that will help students to remember key information.	
078	**Revision speed dating** – A fun and engaging activity that gets students talking.	
079	**Match IT** – Make revision engaging and memorable by turning it into a game!	

PART 9: PROGRAMMING ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
080	**Magic 8-ball®** – Students create a Magic 8-ball® game using python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15.
081	**Shakespearean insult generator** – A fun way to introduce lists and file-handling in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15.
082	**Chatting robot** – Students learn how to create a 'rule-based' chat bot using python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15.
083	**Just dance** – A lesson which uses dance as a medium for introducing key programming concepts to children.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15.
084	**Adventures in text** – Students learn how to create an 80s-style text adventure game in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15, 3A-AP-17.
085	**Mad Libs®** – Students code the popular phrasal template word game in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15, 3A-AP-17.
086	**Sorting Hat** – Students create a Harry Potter-style sorting hat in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15.
087	**Turtle power (a lesson using PRIMM)** – Students learn how to create regular polygons using the turtle library in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15.
088	**Guess my number** – A fun programming challenge which teaches concepts such as variables, data types and selection.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15,

		1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15.
089	**Mind-reading algorithm** – Students learn how to create a mind-reading game in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15.
090	**Cat and mouse** – A simple cat and mouse game using Scratch.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15.
091	**Reaction timer** – Students create a simple reaction timer using python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.

PART 10: COMPUTING AND STEAM

IDEA	DESCRIPTION	STANDARDS
092	**Art attack** – Using art as a creative medium for exploring complex concepts in computing.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-IC-26.
093	**Lights, camera, action** – Students learn how to create light art using slow shutter speed photography and code.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-IC-26.
094	**Making music** – Students learn how to create music with code.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-IC-26.
095	**Coding probability** – Students explore probability, including relative frequency, with code.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-IC-26.
096	**Physical computing** – Exploring how to teach coding using physical devices.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 2-CS-01, 2-CS-02, 3A-AP-15. 3A-CS-01, 3A-AP-13, 3A-AP-16.
097	**Turtle snowflakes** – Students learn how to code snowflakes using the turtle library in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.
098	**Coding the weather** – Students learn how to manipulate 'real' weather data using python and OpenWeather data.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15, 3A-AP-17, 1B-DA-06, 1B-DA-07, 2-DA-08.
099	**Rubbish robots** – Students are challenged to build a robot using general household objects and electronic components.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-15, 3A-AP-17.

100	**Color splash** – Students learn how to create colour splash images using a free online image editor.	1B-AP-12, 1B-AP-14, 1B-IC-21.

PART 11: BONUS ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
101	**What's your elf name?** – Students create a name generator using python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15, 3A-AP-17.
102	**Guess / _ _e / word** – Students create a hangman style word game using python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15, 3A-AP-17.
103	**Cards against humanities** – Students code a phrasal template word game in python.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15, 3A-AP-17.
104	**Shakespearean complement generator** – Coding challenge based on the Shakespearean sonnet 'Shall I compare thee to a summer's day?'	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-15, 2-AP-16, 2-AP-17, 3A-AP-14, 3A-AP-15.
105	**Hacking the news** – Hacking the news with HTML.	
106	**Data science detectives** – Teaching computational thinking using historical data.	1A-DA-05, 1B-DA-06, 1B-DA-07, 2-DA-08, 2-DA-09.
107	**Code tracing** – A simple strategy for reducing cognitive load.	1A-AP-14.
108	**Make me happy** – Students create an AI powered sentiment analysis bot using Scratch.	1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-13, 1B-AP-15, 1B-AP-17, 2-AP-11, 2-AP-12, 2-AP-14, 2-AP-15, 2-AP-16, 2-AP-17, 2-DA-08, 3A-AP-14, 3A-AP-15, 3A-AP-17, 3A-AP-18, 3B-AP-09.
109	**Worked examples** – An effective strategy for reducing cognitive load for novice learners.	1A-AP-14.

110	**Team teaching** – Tips for planning a lesson with a colleague.	