CURRICULUM ALIGNMENT GUIDE
ISTE 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

Key (Standards)
E:CTC – EDUCATOR: COMPUTATIONAL THINKING COMPETENCIES
CSE – COMPUTER SCIENCE EDUCATORS
S – STUDENTS

PART 1: PROGRAMMING STRATEGIES

IDEA	DESCRIPTION	STANDARDS
001	**Paired programming** – A research-driven coding strategy for helping novice learners.	E:CTC: 1a, 3b, 5a CSE: 4a
002	**Rubber duck debugging** – A programming strategy used to help students find bugs and in their code.	E:CTC: 1a, 3a, 5a
003	**Code golf** – A programming strategy to help students create more efficient code.	CSE: 2a
004	**Game design** – Using games as a hook to encourage students to learn how to code.	CSE: 4a
005	**PRIMM** – A research-based approach to teaching coding and for reducing cognitive load.	E:CTC: 1a, 3a, 5a, 5c CSE: 4a
006	**Parsons problems** – Help students learn how to code by removing some of the barriers.	E:CTC: 1a, 3a, 5a, 5c CSE: 4a
007	**Use-modify-create** – Reduce anxiety while supporting growth with this simple three-stage approach to learning to code.	E:CTC: 1a, 3a, 5a, 5c CSE: 4a
008	**Hour of Code** – Give your students a 'byte'-sized introduction to computer science with an hour of code.	CSE: 2a
009	**Code bug** – Build resilience and reduce anxiety when teaching children to code by purposely introducing 'bugs' early on in the learning process.	CSE: 2a
010	**Code combat** – Put your students' coding skills to the test by pitting them against each other in code combat!	CSE: 2a
011	**Teaching with robots** – Coding can often be difficult for students to grasp. Robots can provide a simpler, more tangible introduction to programming.	CSE: 2a

PART 2: COMPUTING STRATEGIES

IDEA	DESCRIPTION	STANDARDS
012	**Take your screwdrivers to work** – Students explore how computers work by taking old devices apart.	S: 1d
013	**DART your students** – A strategy designed to improve students' reading comprehension.	E:CTC: 1a CSE: 4a
014	**Contextualise learning** – Explore strategies for making computing relevant and provide 'real-life' learning experiences for students.	S: 3d E:CTC: 1a, 1b, 5b
015	**Go unplugged** – Teaching computing without computers.	E:CTC: 1a, 3b

IDEA	DESCRIPTION	STANDARDS
016	**Socratic debate** – Students debate the social, ethical, and legal issues surrounding the use of computers.	E:CTC: 1a, 3b, 4d CSE: 1d
017	**Peer instruction** – A research-driven approach to teaching difficult concepts that students often misunderstand.	S: 7c E:CTC: 1a, 3a, 3b, 4d, 5a CSE: 4a
018	**Game-based learning** – Exploring the use of games, such as Minecraft: Education Edition, to teach children how to code.	E:CTC: 4d
019	**Using QR codes** – Using QR codes to teach computing theory.	S: 4c CSE: 2a
020	**Escape rooms** – Students must solve a series of binary puzzles to open physical locks and stop a simulated 'virus attack'.	E:CTC: 3b, 4d
021	**Blogs and wikis** – Using blogs and wikis to teach computing theory.	S: 1b, 1c, 3c
022	**Flipped learning** – Reversing the traditional way of teaching to make better use of classroom time.	E:CTC: 1a CSE: 4a
023	**Guided discovery** – An inductive approach to teaching and learning where students take an active role in discovering knowledge and developing understanding for themselves.	E:CTC: 1a

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.	S: 3b E:CTC: 1e CSE: 3a
025	**Copycat** – A fun activity that teaches students about copyright, public domain, fair use, and Creative Commons.	S: 2a, 2b, 2c, 6b E:CTC: 1e CSE: 1d, 3a
026	**Mario Kart ™ spreadsheets** – An example of how to use game-based learning to teach students essential spreadsheet skills.	S: 5b E:CTC: 4a
027	**Fakebook** – An 'escape room' challenge which helps students understand the importance of protecting their online presence.	S: 2a, 2c, 2d E:CTC: 1e CSE: 3a
028	**Database detectives** – Students test their sleuthing skills with this 'whodunnit' themed database challenge.	E:CTC: 4a
029	**Did you meme it?** – Students explore the purpose and ethics of memes before creating their own meme on an agreed topic.	S: 2a, 2b CSE: 3a
030	**Videography** – Students create a YouTube-style instructional video whilst also explore the importance of concise instructions (algorithms).	S: 3c, 6c E:CTC: 4c

IDEA	DESCRIPTION	STANDARDS
031	**Infographics** – Students create infographics about their mobile phone habits.	S: 3c, 5b, 6c, 6d E:CTC: 4a, 4b, 4c
032	**Dragon's Den** – Students work as a team to design an innovative solution to a global problem.	S: 1a, 3d, 4a,4b, 4c, 4d, 6a, 6d, 7c E:CTC: 3b, 4b, 5b
033	**Wayback Machine** – Students learn about their digital footprint and the long-lasting impact of their online actions.	S: 2a, 2b E:CTC: 1e CSE: 4a

PART 4: COMPUTING ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
034	**What a waste** – Students, working in teams, explore innovative ways to reduce e-waste.	S: 2a, 3d, 4a,4b, 4c, 4d, 6a, 7c, 7d E:CTC: 1e, 4b CSE: 1d
035	**Role reversal** – Students take on the role of a teacher.	S: 3a, 6a CSE: 1c
036	**Storage Top Trumps®** – Students explore different storage devices before creating a game of Top Trumps® based on what they have learned.	S: 3a
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.	S: 1d CSE: 1c
038	**Features of a CPU (a lesson using DART)** – Students explore the main features of a CPU.	S: 1d
039	**Internet of things** – Students design a 'smart home' that utilises the internet of things.	S: 4a, 5b E:CTC: 1e, 4a CSE: 1d
040	**The great input/output QR hunt** – Students complete a QR hunt to discover facts about different input and output devices.	S: 1d
041	**Moral machine** – Students explore the ethics behind creating AI for self-driving vehicles.	S: 1d E:CTC: 1e CSE: 1d

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.	S: 5c E:CTC: 1a, 5c
043	**Teaching with magic** – Using magic to teach computational thinking skills.	S: 5c E:CTC: 1a, 3a
044	**Crazy characters** – Students write an algorithm for drawing a monster.	S: 5c E:CTC: 3b, 5c
045	**Puzzle me** – Using puzzles to practise computational thinking skills (decomposition, pattern-matching, abstraction and algorithm design).	S: 5c E:CTC: 1a
046	**Human robot** – Exploring algorithms through physical activities such as movement and dance.	S: 5c E:CTC: 1a, 3b
047	**A-maze-ing algorithms** – Students explore the importance of clear and precise instructions by writing algorithms to solve a simple maze.	S: 5c E:CTC: 1a, 5c
048	**20 questions** – Students explore the efficiency of different search algorithms by playing a game of '20 questions'.	S: 5c, E:CTC: 1a
049	**Breaking the code** – Students develop their problem-solving skills with a series of code-breaking challenges.	S: 5c E:CTC: 1a
050	**Origami algorithms** – Students write algorithms for folding a paper aeroplane or origami animal.	S: 5c E:CTC: 1a, 5c
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.	S: 5c E:CTC: 1a

PART 6: UNPLUGGED ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
052	**Image compression** – Students learn about lossless compression without the use of a computer.	CSE: 1a
053	**Bubble sort dance algorithm** – Students learn how a bubble sort algorithm works is via the medium of Hungarian folk dance!	CSE: 2a
054	**World Wide Web unplugged** – Students role-play what happens when a user enters an address in a web browser.	CSE: 1c
055	**Intelligent piece of paper (AI)** – Exploring artificial intelligence (AI) with a game of Tic-Tac-Toe.	CSE: 1d
056	**Envelope variables** – Demonstrate a simple program that uses variables and assignment by running them on a computer made entirely out of students.	CSE: 2a

IDEA	DESCRIPTION	STANDARDS
057	**Card sort** – Students explore three common sorting algorithms (bubble, merge, and insertion) by sorting playing cards.	CSE: 2a
058	**Binary representation of images (unplugged)** – Students explore how a computer represents images using binary.	CSE: 1a
059	**How computers work** – Students take on the role of various parts of a computer and simulate the running of a simple program.	CSE: 1c
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.	CSE: 1c
061	**Network topologies** – Using string and various other household objects, students simulate the three most common network topologies.	CSE: 1c

PART 7: DATA REPRESENTATION

IDEA	DESCRIPTION	STANDARDS
062	**Binary addition** – Students learn how to add two numbers using binary.	CSE: 1a
063	**Binary numbers** – Students learn about binary.	CSE: 1a
064	**Binary representation of images** – Students explore how a computer represents images using binary.	CSE: 1a
065	**Binary representation of sound** – Students explore how a computer represents sound using binary.	CSE: 1a
066	**Binary bingo** – A fun strategy to test students' understanding of binary representation of numbers.	CSE: 1a
067	**It's all about hex** – Students learn about the hex numbering system.	CSE: 1a
068	**ASCII 'secret' agent** – Students explore how a computer represents text using binary by solving (and creating) as series of coded messages.	CSE: 1a

PART 8: EXAM PREPARATION

IDEA	DESCRIPTION	STANDARDS
069	**Padlet** – Using online curation tools, such as Padlet, to collating resources in preparation for exams.	S: 3c
070	**Round-robin revision** – Make revision fun and engaging with a series of mini games.	E:CTC: 3b
071	**Revision podcasts** – Create revision resources that students can listen to anytime, anywhere!	S: 1b, 1,c, 3c
072	**PEE (point, evidence, explain)** – A simple strategy to help improve the quality of written answers to exam questions.	CSE: 2a
073	**PechaKucha** – A great way to encourage students to be more concise and a little more creative with their presentations.	E:CTC: 4c
074	**Sketch-noting** – A great way to empower students and allow them to synthesise information visually.	S: 3c
075	**Command word bingo** – A simple starter activity that will pay dividends at exam time!	CSE: 2a
076	**BUG hunt** – A technique for helping students understand thoroughly what is expected of them during exams.	CSE: 2a
077	**Tweet IT** – A fun revision strategy that will help students to remember key information.	S: 1b, 3c
078	**Revision speed dating** – A fun and engaging activity that gets students talking.	E:CTC: 3b
079	**Match IT** – Make revision engaging and memorable by turning it into a game!	CSE: 2a

PART 9: PROGRAMMING ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
080	**Magic 8-ball®** – Students create a Magic 8-ball® game using python.	CSE: 1b
081	**Shakespearean insult generator** – A fun way to introduce lists and file-handling in python.	CSE: 1b
082	**Chatting robot** – Students learn how to create a 'rule-based' chat bot using python.	E:CTC: 4b
083	**Just dance** – A lesson which uses dance as a medium for introducing key programming concepts to children.	CSE: 1b

084	**Adventures in text** – Students learn how to create an 80s-style text adventure game in python.	CSE: 1b
085	**Mad Libs®** – Students code the popular phrasal template word game in python.	CSE: 1b
086	**Sorting Hat** – Students create a Harry Potter-style sorting hat in python.	CSE: 1b
087	**Turtle power (a lesson using PRIMM)** – Students learn how to create regular polygons using the turtle library in python.	CSE: 1b
088	**Guess my number** – A fun programming challenge which teaches concepts such as variables, data types and selection.	CSE: 1b
089	**Mind-reading algorithm** – Students learn how to create a mind-reading game in python.	CSE: 1b
090	**Cat and mouse** – A simple cat and mouse game using Scratch.	CSE: 1b
091	**Reaction timer** – Students create a simple reaction timer using python.	CSE: 1b

PART 10: COMPUTING AND STEAM

IDEA	DESCRIPTION	STANDARDS
092	**Art attack** – Using art as a creative medium for exploring complex concepts in computing.	CSE: 1b
093	**Lights, camera, action** – Students learn how to create light art using slow shutter speed photography and code.	CSE: 1b
094	**Making music** – Students learn how to create music with code.	CSE: 1b
095	**Coding probability** – Students explore probability, including relative frequency, with code.	CSE: 1b
096	**Physical computing** – Exploring how to teach coding using physical devices.	S: 4c, 5d CSE: 1b
097	**Turtle snowflakes** – Students learn how to code snowflakes using the turtle library in python.	CSE: 1b
098	**Coding the weather** – Students learn how to manipulate 'real' weather data using python and OpenWeather data.	S: 5a, 5b CSE: 1b
099	**Rubbish robots** – Students are challenged to build a robot using general household objects and electronic components.	E:CTC: 4b
100	**Color splash** – Students learn how to create color splash images using a free online image editor.	S: 6b

PART 11: BONUS ACTIVITIES

IDEA	DESCRIPTION	STANDARDS
101	**What's your elf name?** – Students create a name generator using python.	CSE: 1b
102	**Guess / _ _e / word** – Students create a hangman style word game using python.	CSE: 1b
103	**Cards against humanities** – Students code a phrasal template word game in python.	CSE: 1b
104	**Shakespearean complement generator** – Coding challenge based on the Shakespearean sonnet 'Shall I compare thee to a summer's day?'	CSE: 1b
105	**Hacking the news** – Hacking the news with HTML.	CSE: 1b
106	**Data science detectives** – Teaching computational thinking using historical data.	S: 3d, 5a, 5b, 5c E:CTC: 3b, 4a
107	**Code tracing** – A simple strategy for reducing cognitive load.	E:CTC: 1a, 3a, 5a, 5c
108	**Make me happy** – Students create an AI powered sentiment analysis bot using Scratch.	S: 5b CSE: 1b, 1d
109	**Worked examples** – An effective strategy for reducing cognitive load for novice learners.	E:CTC: 1a, 3a, 5a, 5c
110	**Team teaching** – Tips for planning a lesson with a colleague.	E:CTC: 1c, 3c