

Computer Science Unit Overview Year 8

Computer Science - Year 8				
What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Computer systems	<ul style="list-style-type: none"> Recall that a general-purpose computing system is a device for executing programs Recall that a program is a sequence of instructions that specify operations that are to be performed on data Explain the difference between a general-purpose computing system and a purpose-built device 	<p>Learners can explain what sets devices apart from other purpose-built machinery.</p> <p>Learners develop an understanding of this unique characteristic.</p> <p>Learners will compare calculating machines from the past to modern general-purpose computers.</p> <p>Learners will execute a program themselves.</p>	<p>In Year 7 students will have explored many different uses of computer systems but not expressly covered specifics building towards this unit.</p> <p>Learners will have an awareness and common understanding of basic computer system elements but the topic as a whole will be new to students.</p>	<ul style="list-style-type: none"> ■ scratch.mit.edu ■ www.computerhistory.org ■ teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/the-intelligent-piece-of-paper-activity ■ thecrashcourse.com/courses/computerscience ■ www.youtube.com/watch?v=5ocq6_3-nEw ■ jessecrossen.github.io/ttsim ■ www.khanacademy.org/computing/computer-science/how-computers-work ■ en.wikipedia.org ■ youtu.be/DFBbSTvtpy4 ■ youtu.be/CO67EQ0ZWg ■ youtu.be/n-zeeRLBgd0 ■ teachablemachine.withgoogle.com ■ experiments.withgoogle.com/collection/ai ■ quickdraw.withgoogle.com ■ machinelearningforkids.co.uk ■ projects.raspberrypi.org ■ code.org/oceans ■ royalsociety.org
	<ul style="list-style-type: none"> Describe the function of the hardware components used in computing systems Describe how the hardware components used in computing systems work together in order to execute programs Recall that all computing systems, regardless of form, have a similar structure ('architecture') 	<p>Learners can explain how humans can read instructions, following them one at a time, and keeping track of where they are and what the current state is comes very naturally.</p> <p>Learners should explain the role of hardware components, i.e. the actual 'machinery' that allows computing systems to fulfil this purpose.</p> <p>Learners will discover how all computing systems, regardless of form or capabilities, make use of the same components: a processor, memory, storage,</p>		

		<p>input and output devices, and communication components. Learners should explain a simple, concise picture of what each of these 'universal' components does, and how they work together in order to execute programs.</p>		
	<ul style="list-style-type: none"> Analyse how the hardware components used in computing systems work together in order to execute programs Define what an operating system is, and recall its role in controlling program execution 	<p>Learners should explain the abstract descriptions of how the processor, memory, storage, and communication components interact with each other and function as a system. Learners should explain role of the operating system, which is responsible for managing the complexity of modern computing devices.</p>		
	<ul style="list-style-type: none"> Describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions Use logic gates to construct logic circuits, and associate these with logical operators and expressions Describe how hardware is built out of increasingly complex logic circuits Recall that, since hardware is built out of logic circuits, data and instructions alike need to be represented using binary digits 	<p>Learners should display the use of logical expressions in software. Learners should explain how logic and circuits work together to create a computer system.</p>		

	<ul style="list-style-type: none"> • Provide broad definitions of 'artificial intelligence' and 'machine learning' • Identify examples of artificial intelligence and machine learning in the real world • Describe the steps involved in training machines to perform tasks (gathering data, training, testing) • Describe how machine learning differs from traditional programming • Associate the use of artificial intelligence with moral dilemmas 	<p>Learners should explain the thoughts of Alan Turing “[proposed] to consider the question, ‘Can machines think?’”</p> <p>Learners should define the term ‘artificial intelligence’, and explore the kinds of problems that it has traditionally dealt with. Learners should explain machine learning, and investigate its relationship with conventional programming.</p>		
	<ul style="list-style-type: none"> • Explain the implications of sharing program code 	<p>In this final lesson, learners will take a quiz that will assess their understanding of the computing systems concepts that they have encountered throughout the unit.</p>		

Computer Science - Year 8

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Developing for the web	<ul style="list-style-type: none"> Describe what HTML is Use HTML to structure static web pages Modify HTML tags using inline styling to improve the appearance of web pages 	<p>Learners can explain how web pages are constructed using HTML tags, and how they can be modified to start to resemble the websites they are accustomed to.</p> <p>Learners can use formatting on sections of text to improve readability. Learners can modify tags to change their appearance in a document, to make them different from the defaults provided.</p>	<p>From KS2, linked learning students should have the learnt how to:</p> <ul style="list-style-type: none"> design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts use sequence, selection, and repetition in programs; work with variables and various forms of input and output use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content 	<ul style="list-style-type: none"> www.w3schools.com/html www.w3schools.com/css www.w3schools.com/cssref
	<ul style="list-style-type: none"> Display images within a web page Apply HTML tags to construct a web page structure from a provided design 	<p>Learners can explain the structure and operation of the img tag and understand how they can be used to 'add' images to web pages.</p>		
	<ul style="list-style-type: none"> Describe what CSS is Use CSS to style static web pages Assess the benefits of using CSS to style pages instead of in-line formatting 	<p>Learners can explain how CSS is a more efficient way of styling HTML documents.</p> <p>Learners will use CSS to format tags in a HTML document. They will then progress on to applying their own formatting schemes to work they have already created.</p>		
	<ul style="list-style-type: none"> Describe what a search engine is Explain how search engines 'crawl' through the 	<p>Learners can explain how web pages are found and catalogued, ready for people to search for them. Students can</p>		

	<p>World Wide Web and how they select and rank results</p> <ul style="list-style-type: none"> Analyse how search engines select and rank results when searches are made 	<p>explain how they can make their designs appear towards the top of search engine lists, so that more people will view what they have created.</p>		
	<ul style="list-style-type: none"> Use search technologies effectively Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used Create hyperlinks to allow users to navigate between multiple web pages 	<p>Learners can apply advanced search techniques. Learners can explain how search operators can be used to combine or exclude search terms to either expand or narrow search results. Learners will be able to hyperlink web pages into a complete website allowing navigation between the pages that they create.</p>		
	<ul style="list-style-type: none"> Implement navigation to complete a functioning website Complete summative assessment 	<p>Learners will create a web page to summarise their learning over the entire unit by creating an additional 'How to' web page, they will follow this up by adding navigation to all pages of the website.</p>		

Computer Science - Year 8

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
<p align="center">Introduction to Python Programming</p>	<ul style="list-style-type: none"> Describe what algorithms and programs are and how they differ Recall that a program written in a programming language needs to be translated in order to be executed by a machine Locate and correct common syntax errors Describe the semantics of assignment statements 	<p>learners will write and execute their first programs in Python. Learners will assign values to effectively named variables.</p>	<p>From KS2, linked learning students should have the learnt how to:</p> <ul style="list-style-type: none"> design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts use sequence, selection, and repetition in programs; work with variables and various forms of input and output use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs 	<ul style="list-style-type: none"> repl.it blog.teachcomputing.org/tag/pedagogy pythontutor.com/visualize.html trinket.io projects.raspberrypi.org docs.python.org/3 www.gutenberg.org/ebooks/345
	<ul style="list-style-type: none"> Write simple Python programs that display messages, assign values to variables, and receive keyboard input Use simple arithmetic expressions in assignment statements to calculate values Receive input from the keyboard and convert it to a numerical value 	<p>Learners will assign values to effectively named variables. Learners will apply arithmetic and mathematics in programming. Learners will display messages and receive inputs from a keyboard</p>		
	<ul style="list-style-type: none"> Use relational operators to form logical expressions Use binary selection (if, else statements) to control the flow of program execution 	<p>Learners will apply selection and randomness to programming concepts. Learners will revisit some of the programs that they have encountered in previous lessons and extend them into</p>		

	<ul style="list-style-type: none"> • Generate and use random integers 	more versatile programs that use selection.		
	<ul style="list-style-type: none"> • Use multi-branch selection (if, elif, else statements) to control the flow of program execution • Describe how iteration (while statements) controls the flow of program execution 	Learners will apply multi-branch selection using if, elif and else.		
	<ul style="list-style-type: none"> • Use iteration (while loops) to control the flow of program execution • Use variables as counters in iterative programs 	Learners will apply count controlled iteration in their programming and will use iteration in the form of a while loop.		
	<ul style="list-style-type: none"> • Combine iteration and selection to control the flow of program execution • Use Boolean variables as flags 	Learners will use conditions to create indefinite loops.		

Computer Science - Year 8

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Media Vector Graphics	<ul style="list-style-type: none"> • Draw basic shapes (rectangle, ellipse, polygon, star) with different properties (fill and stroke, shape-specific attributes) • Manipulate individual objects (select, move, resize, rotate, duplicate, flip, z-order) 	<p>Learners will create vector graphics. Learners will apply basic geometric shapes to create images. Learners will apply layering to create their images.</p>	<p>From KS2, linked learning students should have the learnt how to:</p> <ul style="list-style-type: none"> • use technology purposefully to create, organise, store, manipulate and retrieve digital content 	<ul style="list-style-type: none"> ■ inkscape.org
	<ul style="list-style-type: none"> • Manipulate groups of objects (select, group/ungroup, align, distribute) • Combine paths by applying operations (union, difference, intersection) 	<p>Learners will work with multiple objects. Learners will use tools to align, distribute, group, and combine objects.</p>		
	<ul style="list-style-type: none"> • Convert objects to paths • Draw paths • Edit path nodes 	<p>Learners will explain that vector graphics are made up of paths and that these paths include nodes at the start, end, and at changes of path direction. Learners will convert shapes to paths and then edit them.</p>		
	<ul style="list-style-type: none"> • Combine multiple tools and techniques to create a vector graphic design 	<p>Learners start an open-ended project to apply all previous learning effectively.</p>		
	<ul style="list-style-type: none"> • Explain what vector graphics are 	<p>Learners can explain how vector images are stored.</p>		

	<ul style="list-style-type: none"> • Provide examples where using vector graphics would be appropriate 			
	<ul style="list-style-type: none"> • Peer assess another pair's project work • Improve your own project work based on feedback • Complete a summative assessment 	<p>Learners can compare vector images with bitmaps images.</p>		

Computer Science - Year 8

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Mobile app development	<ul style="list-style-type: none"> • Identify when a problem needs to be broken down • Implement and customise GUI elements to meet the needs of the user 	<p>Learners will apply decomposition to a larger problem to design a solution.</p> <p>Learners can explain how decomposition is an important process to follow when tackling a large problem such as creating an app.</p>	<p>From KS2, linked learning students should have the learnt how to:</p> <ul style="list-style-type: none"> • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output • use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs 	<ul style="list-style-type: none"> ■ code.org/educate/applab ■ support.code.org/hc/en-us/articles/115000488132-Creating-a-classroom-section ■ www.youtube.com/watch?v=EhkxDlr0y2U ■ www.youtube.com/watch?v=e1St8LB4VJA ■ www.youtube.com/watch?v=fypSGGZZfzM
	<ul style="list-style-type: none"> • Recognise that events can control the flow of a program • Use user input in an event-driven programming environment • Use variables in an event-driven programming environment • Develop a partially complete application to include additional functionality 	<p>Learners can effectively and accurately apply the concept of event-driven programming in creating an app.</p>		
	<ul style="list-style-type: none"> • Identify and fix common coding errors • Pass the value of a variable into an object • Establish user needs when completing a creative project 	<p>Learners can identify and correct errors in code. Learners will create and store values in variables and then pass those values to other variables.</p>		
	<ul style="list-style-type: none"> • Apply decomposition to break down a large problem into more manageable steps 	<p>Learners will take user input in their programming.</p>		

	<ul style="list-style-type: none"> • Use user input in a block-based programming language • Use a block-based programming language to create a sequence • Use variables in a block-based programming language 			
	<ul style="list-style-type: none"> • Use a block-based programming language to include sequencing and selection • Use user input in a block-based programming language • Use variables in a block-based programming language • Reflect and react to user feedback 	<p>Learners will build and make programming on their apps using pair programming.</p>		
	<ul style="list-style-type: none"> • Use a block-based programming language to include sequencing and selection • Use user input in a block-based programming language • Use variables in a block-based programming language • Evaluate the success of the programming project • 	<p>Learners will evaluate their app projects. Learners will evaluate their success against their criteria.</p>		

Computer Science - Year 8

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Representation	<ul style="list-style-type: none"> • List examples of representations • Recall that representations are used to store, communicate, and process information • Provide examples of how different representations are appropriate for different tasks 	<p>Learners can explain the origins of representing information in the form of writing and symbols.</p>	<p>From KS2, linked learning students should have the learnt how to:</p> <ul style="list-style-type: none"> • use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs 	<ul style="list-style-type: none"> ■ scratch.mit.edu ■ en.wikipedia.org ■ teachinglondoncomputing.org/lego-braille ■ csunplugged.org/en ■ csfieldguide.org.nz/en ■ archive.org/details/advancementofl00baco/page/256 ■ curriculum.code.org ■ www.cs4fn.org ■ denninginstitute.com/pjd/GP/GP-site/welcome.html <p>http://www.futurelearn.com/courses/how-computers-work</p>
	<ul style="list-style-type: none"> • Recall that characters can be represented as sequences of symbols and list examples of character coding schemes • Measure the length of a representation as the number of symbols that it contains • Provide examples of how symbols are carried on physical media 	<p>Learners encode, transmit, and decode short messages using different coding scheme and communication mediums.</p>		
	<ul style="list-style-type: none"> • Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters • Measure the size or length of a sequence of bits as the number of binary digits that it contains 	<p>Learners can explain what binary digits are. Learners can explain that letters are stored in a computer system using binary codes.</p>		

	<ul style="list-style-type: none"> • Describe how natural numbers are represented as sequences of binary digits • Convert a decimal number to binary and vice versa 	<p>Learners will convert decimal to binary, binary to decimal and can explain how the binary number system works.</p>		
	<ul style="list-style-type: none"> • Convert between different units and multiples of representation size • Provide examples of the different ways that binary digits are physically represented in digital devices 	<p>learners can list the standard units of storage from bits to Petabytes. Students can explain how much is stored in each unit of measurement. Learners will convert between the different units.</p>		