Computing Curriculum Year 8

Year 8		HT1	HT2	HT3	HT4	HT5	HT6
Computing	Topic Big Idea/Question	Topic 1 **Computing Systems**	Topic 2 **Representation from clay to silicon**	Topic 3 **Development from the web**	Topic 4 **Design Vector graphics**	Topic 5 **Python Turtle – Text based coding**	Topic 6 **Python Introduction – Text based coding**
	Why this and why now? What is the content doing here? How does it integrate to prior learning or prepare students for future learning? Is it an opportunity for cumulative learning or to achieve proficiencies? Does it provide a step to collective sufficiency?	This unit takes learners on a tour through the different layers of computing systems: from programs and the operating system to the physical components that store and execute these programs, to the fundamental binary building blocks that these components consist of. The aim is to provide a concise overview of how computing systems operate, conveying the essentials and abstracting away the technical details that might confuse or put off learners. The last lessons cover two interesting contemporary topics: artificial intelligence and open-source software. These are linked back to	This unit conveys essential knowledge relating to binary representations. The activities gradually introduce learners to binary digits and how they can be used to represent text and numbers. The concepts are linked to practical applications and problems that the learners are familiar with.	In this unit, learners will explore the technologies that make up the internet and World Wide Web. Starting with an exploration of the building blocks of the World Wide Web, HTML, and CSS, learners will investigate how websites are catalogued and organised for effective retrieval using search engines. By the end of the unit, learners will have a functioning website.	This unit offers students the opportunity to design graphics using vector graphic editing software. The lessons are tailored to Inkscape (inkscape.org), which is open source and cross-platform, but the resources should be readily adaptable to any vector graphics editor. Vector graphics can be used to design anything from logos and icons to posters, board games, and complex illustrations. Through this unit, students will be able to better understand the processes involved in creating such graphics and will be provided with the knowledge and tools to create their own. One of the most interesting and	This unit introduces learners to text-based programming with Python. The lessons form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. Emphasis is placed on tackling common misconceptions and elucidating the mechanics of program execution. A range of pedagogical tools is employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples. The Year 7 Programming units (ncce.io/year7) are a prerequisite for this unit.	

		the content of the unit, helping learners to both broaden their knowledge and focus on the topics addressed in the unit. The unit assumes no prior knowledge. There are, however, links to the 'Representations' units taught in Years 8 and 9 and the 'Networks' units taught in Years 7 and 8.			challenging aspects of creating vector graphics is their unlikely link to computational thinking. Creating a complex design is a multi-step process that starts with elementary shapes and involves combining them into more intricate ones using operations such as union, difference, and intersection. There are usually multiple paths to achieving the goal and the process involves decomposition, evaluation, and plenty of inventiveness!	
	What is the essential knowledge that needs to be remembered? What are the key facts, skills, and experiences that you want students to remember? What are the substantive and disciplinary concepts? Does the knowledge selected mean students leave	Learning about computing systems, learners will focus on what sets these devices apart from other purpose-built machinery: it is their ability to execute programs that allows them to modify their operation and perform different tasks, and thus become our most versatile 'tool for thought.' Learners will develop an understanding of this unique characteristic; learners will compare calculating machines from the past to modern	Learners discuss familiar examples of representations, some of which date back millennia, to better understand their use and characteristics. This prepares learners for their encounter with binary representations in the context of computing, and places these within a much broader (and more familiar) context. Learners work in groups through an activity that requires them to encode, transmit, and decode short messages, with each group using a	We use web pages every day without questioning how they work. This Unit looks behind the curtain to help learners start to understand how web pages are constructed using HTML tags, and how they can be modified to start to resemble the websites to which they are accustomed. Learners will begin by considering the power of automation for repetitive tasks, before delving into some practical web page formatting activities using HTML tags.	Students are provided with examples of vector graphics, such as logos, icons, and illustrations, to get a sense of what they will be developing throughout the course of this unit. Through a sequence of guided steps, learners are acquainted with the basics of using software to draw geometrical shapes and manipulate them. These elementary shapes will form the building blocks of the more complex designs they will be creating in the lessons to come.	In this introductory lesson, learners will write and execute their first programs in Python. They will go through the basics of displaying messages, assigning values to variables, and receiving input from the keyboard. They will familiarise themselves with an entirely different programming environment than the block-based one that they may be accustomed to. It is an environment where they will need to know by heart all of the constructs that they can use, instead of having the options laid out in front of them. It is also an environment in which errors arise if they get a single letter or symbol wrong. One of the main goals of this lesson (and of the unit) is to support them in this transition, by providing associations with concepts that they are already familiar with and building their confidence in overcoming common obstacles. Before doing any programming, learners will be introduced to what algorithms and programs are, and

	with a good understanding? Substantive – key facts Disciplinary- Methods of subjects Procedural- Skills	general-purpose computers. Learners will look at how hardware and software are both needed for effective computing. Learners will identify how instructions are used for effective computing. Learners will discover the various hardware required for computers to operate effectively. Learners will discover how all computing systems, regardless of form or capabilities, make use of the same components: a processor, memory, storage, input and output devices, and communication components. The learner will be able to describe how the processor, memory, storage, and communication components interact with each other and function as a system will now be embedded in concrete, familiar scenarios that the learners will investigate. Learners will be introduced to the operating system, which	different coding scheme and communication medium (signals, light, sounds, holes on paper, etc.). The activity reinforces the learners' understanding of text representation using sequences of symbols, while emphasis is placed on distinguishing between symbols and the way in which they are embodied in physical media. Learners grasp what binary digits are by associating them with familiar sets of symbols such as letters and decimal digits. Learners solve simple problems that reinforce the connection between (alphanumeric) information and its binary representation. They also consider the question of why binary digits are used in conjunction with computing systems. Learners build upon their familiarity with using a decimal numbering system, in order to draw analogies with how numbers can be represented using binary. They use activities, either	Learners will then modify tags to change their appearance in a document, to make them different from the defaults provided. Learners will look at the fundamentals of web page design, specifically the use of tags and their modification. They will explore the structure and operation of the img tag and understand how they can be used to 'add' images to web pages. Learners will identify the benefit of using a computer is that it is a device that allows the easy editing of content. Learners will find efficient ways to automate what they do, and, in this lesson, learners will see that CSS is a more efficient way of styling HTML documents. Learners experiment with using CSS to format tags in a HTML document. They will then progress on to applying their own formatting schemes to	Using path operations such as union, difference, and intersection, students are able to combine simple shapes into more complex ones. The purpose here is simply to familiarise them with these operations and what they can do, so the tasks they perform are scaffolded and they never need to perform more than one or two operations in order to create the target shapes. Students are presented with a set of monochrome icons and are challenged to create some of them from scratch, starting from elementary shapes and combining them with path operations. The icons are carefully selected to range from simple ones that are straightforward to produce to more complex ones that require some creative thinking. Learners know the ingredients they can use (the elementary shapes) and how to combine them (the operations), but they need to work out	how they are different. Through this discussion, they will start to build an understanding of what it means to express instructions in a formal language, and how these instructions can eventually be executed by a machine. In the previous lesson, learners were introduced to displaying messages, assigning values to variables, and receiving input from the keyboard. This lesson will help them gain a deeper understanding of assignments, and explicitly address some of the common misconceptions around the semantics of assignment statements. Learners will also be introduced to using arithmetic expressions and receiving numerical input from the keyboard. These are two key components that will allow them to progress building more elaborate programs in the lessons to follow. The main activity in this lesson will require learners to construct their own short programs for the first time, through scaffolded tasks. This lesson introduces selection and randomness. These are two features that will allow learners to develop programs with a diverse range of behaviours. Learners will revisit some of the programs that they have encountered in previous lessons and extend them into more versatile programs that use selection. They will develop a simple number guessing game, which will eventually include randomness. This lesson progresses to multi-branch selection, then introduces while, the general-purpose iterative structure available in Python. Learners will explore problems that will allow them to deepen their comprehension of when and how selection should be used. For example, they will build programs that check the weather conditions where they are living and display appropriate responses. They will also be introduced to iteration, making sure that they understand the mechanics of how it works

		is responsible for managing the complexity of modern computing devices. Learners will bridge the gap between logic and circuits and make the direct link between them explicit. Learners will discover if 'Can machines think?'" Given the advances in artificial intelligence and machine learning, especially in the last decade. Learners will attempt to define the term 'artificial intelligence' and explore the kinds of problems with which it has traditionally dealt. They will also focus on machine learning and investigate its relationship with conventional programming. practices.	unplugged or software-based, to become familiar with binary number representation and convert between binary and decimal. This lesson familiarises learners with bytes and the prefixes used for measuring representation size, such as 'kilo-,' 'mega-,' 'giga-' and 'tera-'. Simple activities embed these concepts in real-life settings and introduce learners to conversions between the different units and multiples. The unit is concluded with a summative assessment quiz and a puzzle activity that challenges learners to unchain Alan Turing's mug.	work they have already created. Learners will consider how web pages are found and catalogued, ready for people to search for them. By considering how search engines find and rank web pages, they will learn how they can make their designs appear towards the top of search engine lists, so that more people will view what they have created. Learners will investigate advanced search techniques. They will understand how search operators can be used to combine or exclude search terms to either expand or narrow search results. They will practice using these terms for specific purposes. Learners will be able to hyperlink web pages into a complete website allowing navigation between the pages that they create.	the steps that will lead them from start to finish. In many cases there will be alternative paths to the same goal. Students undertake a short, open-ended project out of a range of suggestions and work in pairs to complete it. It is time to investigate what vector images are really made of. Students look at and modify an .svg file to grasp how it is a structured description of an image and how that image is rendered when we need to view it. With this knowledge in mind, students will explore cases where vector graphics are (or are not) useful.	before they go on to build their own iterative programs in the next lesson. At times, learners will import and use functions from 'home-grown' modules, i.e., modules that have been created exclusively for the purposes of the lesson. This will give them an insight into how a text-based language can be more powerful than block-based languages, without placing additional cognitive burden on them. In the first part of this lesson, learners will be introduced to counting. Counters are important, as they are the simplest example of variables that are used to compute results iteratively, with each new value accumulated over the previous ones. In the second part of the lesson, learners will apply the skills and knowledge that they have developed to create a times tables practice game. It is an example that naturally combines iteration and selection, while also being useful. In this final lesson of the unit, learners will apply and consolidate what they have learnt by extending the number guessing game that they developed previously into an iterative version that allows them multiple guesses. They will then conclude the unit with a summative assessment quiz.

	What is the assessment intent and how will you assess? What types of assessments and question stems are being used to demonstrate students are learning and progressing to produce ever higher standards of work? What formative assessment is there for component learning and summative for composite learning?	Assessment will be in a variety of forms. There will also be an ongoing formative assessment based on student work. This will be in the form of presentations and questioning. This will be both Peer and Teacher led Summative assessment will take place at the end of the unit of work based on topics learned. This will be a paper test. Each lesson will start with a mini quiz on forms. This will identify and test knowledge from the previous lesson and from previous topics covered. At the end of each lesson there will be a plenary on forms, and this will test knowledge and learning from the lesson.	Assessment will be in a variety of forms. There will also be an ongoing formative assessment based on student work. This will be in the form of presentations and questioning. This will be both Peer and Teacher led Summative assessment will take place at the end of the unit of work based on topics learned. This will be a paper test. Each lesson will start with a mini quiz on forms. This will identify and test knowledge from the previous lesson and from previous topics covered. At the end of each lesson there will be a plenary on forms, and this will assess knowledge and learning from the lesson.	Assessment will be in a variety of forms. There will also be an ongoing formative assessment based on student work. This will be in the form of presentations and questioning. This will be both Peer and Teacher led Summative assessment will take place at the end of the unit of work based on topics learned. This will be a paper test. Each lesson will start with a mini quiz on forms. This will identify and test knowledge from the previous lesson and from previous topics covered. At the end of each lesson there will be a plenary on forms, and this will assess knowledge and learning from the lesson.	Assessment will be in a variety of forms. There will also be an ongoing formative assessment based on student work. This will be in the form of presentations and questioning. This will be both Peer and Teacher led Summative assessment will take place at the end of the unit of work based on topics learned. This will be a paper test. Each lesson will start with a mini quiz on forms. This will identify and test knowledge from the previous lesson and from previous topics covered. At the end of each lesson there will be a plenary on forms, and this will assess knowledge and learning from the lesson.	Assessment will be in a variety of forms. There will also be an ongoing formative assessment based on student work. This will be in the form of presentations and questioning. This will be both Peer and Teacher led Summative assessment will take place at the end of the unit of work based on topics learned. This will be a paper test. Each lesson will start with a mini quiz on forms. This will identify and test knowledge from the previous lesson and from previous topics covered. At the end of each lesson there will be a plenary on forms, and this will assess knowledge and learning from the lesson.
	What does the end point look like? What is the impact of this component on the student's learning? What	Learners will be able to effectively identify how computing machines work. That they are a combination of hardware and software. Learners will be able to identify components that make a computer work	Learners will be able to effectively identify different coding schemes and how they relate to computing. Learners will be able to use binary digits effectively and how they relate to computing. Learners may also be	Learners will be able to create a web page from HTML. They will be able to insert text and images appropriately. Learners will be able to modify their web page to make it look better.	Learners will be able modify existing shapes. Learners will be able to select, move, resize, rotate, duplicate, flip and z-order shapes effectively.	Learners will be able to write algorithms that allow the program to complete successfully. Learners will be able to identify and correct errors appropriately. Learners will be able to use variables and operators to ensure successful code. Learners will be able to use iteration to make their code more efficient. • Describe what algorithms and programs are and how they differ

	should the learning now look like via the assessment? Is disciplinary language used?	Learners will be able to: • 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 • 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')	able to add binary digits together Learners will be able to: • 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 • 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 • Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters	• Describe what HTML is • Use HTML to structure static web pages • Modify HTML tags using inline styling to improve the appearance of web pages • Display images within a web page • Apply HTML tags to construct a web page structure from a provided design • 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 • Describe what a search engine is • Explain how search engines 'crawl' through the World Wide Web and how they select and rank results • Analyse how search engines select and rank results when searches are made	• 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) • Manipulate groups of objects (select, group/ungroup, align, distribute) • Combine paths by applying operations (union, difference, intersection) • Convert objects to paths • Draw paths • Edit path nodes • Combine multiple tools and techniques to create a vector graphic design • Explain what vector graphics are • Provide examples where using vector graphics would be appropriate • Peer assesses another pair's project work	• Recall that a program written in a programming language needs to be translated to be executed by a machine • Write simple Python programs that display messages, assign values to variables, and receive keyboard input • Locate and correct common syntax errors • Describe the semantics of assignment statements • Use simple arithmetic expressions in assignment statements to calculate values • Receive input from the keyboard and convert it to a numerical value • Use relational operators to form logical expressions • Use binary selection (if, else statements) to control the flow of program execution • Generate and use random integers • 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 • Use iteration (while loops) to control the flow of program execution • Use variables as counters in iterative programs • Combine iteration and selection to control the flow of program execution • Use Boolean variables as flags

		• 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 • 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 • Provide broad definitions of 'artificial	• Measure the size or length of a sequence of bits as the number of binary digits that it contains • Describe how natural numbers are represented as sequences of binary digits • Convert a decimal number to binary and vice versa • Convert between different units and multiples of representation size • Provide examples of the different ways that binary digits are physically represented in digital devices	• Use search technologies effectively • Discuss the impact of search technologies and the issues that arise they function and the way they are used • Create hyperlinks to allow users to navigate between multiple web pages • Implement navigation to complete a functioning website	• Improve your own project work based on feedback	

		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 • Explain the implications of sharing program code				
	How does it cover the NC? Refer explicitly to the NC or KS4 Assessment Objectives.	The topic meets the NC statement requirements for strands 3.4/3.5/3.6	The topic meets the NC statement requirements for strands 3.6	The topic meets the NC statement requirements for strands 3.8	The topic meets the NC statement requirements for strands 3.7/3.8	The topic meets the NC statement requirements for strands 3.1/3. /3.3/3.6

Copley Academy