THE ADEYFIELD ACADEMY
LEARNING WITHOUT LIMITS

Computer Science

Year 7 - Scheme of Learning

Word Rich - Oracy, Vocabulary, Reading, Writing	Character (SMSC & Values)	Careers & Employability	Enrichment & Cultural Capital	Equality, Diversity & Inclusivity
❖ Keywords in all lessons - digital language	• links made throughout	• links to future careers	• wider reading on the digital world	• Special projects based around Alan Turning and LGBTQ and Equity

Formal Assessments (Title/Date)	Blended Learning	Home Learning
Assessments are carried out at the end of each unit, both knowledge and skills are assessed	• Lessons and resources in google classroom	• Lessons and resources in google classroom

Unit of Work	Knowledge and *Skills*	Curriculum Links and Sequencing	National Curriculum
Impact of Technology **Collaborating online respectfully** *Autumn1*	★ Introduction/Induction to use logins, computers and equipment. ★ Create a memorable and secure password for an account on the school network ★ Remember the rules of the computing lab ★ Find personal documents and common applications ★ Recognise a respectful email ★ Construct an effective email and send it to the correct recipients ★ Describe how to communicate with peers online ★ Describe cyberbullying ★ Explain the effects of cyberbullying ★ Plan effective presentations for a given audience	**GCSE Link:** ➔ Data protection ➔ Encryption **Wider Curriculum Links:** ➔ Cultural ➔ SMSC ➔ GDPR **Sequencing Links:** ➔ Online relationships ➔ Online bullying ➔ Safeguarding ➔ Privacy and Security	National curriculum links • Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability • Understand a range of ways to use technology safely, respectfully, responsibly, and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns Education for a Connected World links **Online relationships** • I can explain strategies for assessing the degree of trust I place in people or organisations online. (Y7) • I can give examples of how to make positive contributions to online debates and discussions. (Y8)

	★ Check who you are talking to online		**Online bullying** ● I can describe how bullying may change as we grow older and recognise when it is taking place online. (Y7) ● I can identify and demonstrate actions to support others who are experiencing difficulties online. (Y7) **Privacy and security** ● I can create and use strong and secure passwords. (Y5) ● I can explain how my internet use is often monitored (e.g. by my school or internet service provider). (Y7)
Year 7 – Gaining support for a cause *Autumn 2*	● Select the most appropriate software to use to complete a task ● Identify the key features of a word processor ● Apply the key features of a word processor to format a document ● Evaluate formatting techniques to understand why we format documents ● Select appropriate images for a given context ● Apply appropriate formatting techniques ● Demonstrate an understanding of licensing issues involving online content by applying appropriate Creative Commons licences ● Demonstrate the ability to credit the original source of an image ● Critique digital content for credibility ● Apply techniques to identify whether or not a source is credible	**GCSE Link:** ➔ Copyrights ➔ Ownership ➔ Intellectual properties ➔ Referencing **Wider Curriculum Links:** ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ Copyrights ➔ GDPR ➔ Patents	**National curriculum links** ● Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users ● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability **Education for a Connected World links** **Managing online information** ● I can use a range of features to quality assure the content I access online. (11–14) ● I can explain how to use search effectively and use examples from my own practice to illustrate this. (11–14) **Copyright and ownership**

	• Apply referencing techniques and recognise the concept of plagiarism • Evaluate online sources for use in own work • Construct a blog using appropriate software • Create content for a blog based on credible sources • Apply referencing techniques that credit authors appropriately • Design the layout of the content to make it suitable for the audience • Construct a blog using appropriate software • Create content for a blog based on credible sources • Apply referencing techniques that credit authors appropriately • Design the layout of the content to make it suitable for the audience		• I know that commercial online content can be viewed, accessed, or downloaded illegally. (11–14) • I can accurately define the concept of plagiarism. (11–14) • I can use this definition to evaluate my own use of online sources. (11–14) • I understand the concept of software and content licensing. (11–14) • I understand Creative Commons Licensing protocols. (11–14) • I can identify the potential consequences of illegal access or downloading and how it may impact me and my immediate peers. (11–14)
Year 7 — Programming I *Spring 1*	• Compare how humans and computers understand instructions (understand and carry out) • Define a sequence as instructions performed in order, with each executed in turn • Predict the outcome of a simple sequence • Modify a sequence • Define a variable as a name that refers to data being stored by the computer • Recognise that computers follow the control flow of input/process/output	**GCSE Link:** ➔ Programming **Wider Curriculum Links:** ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ Sequencing ➔ Variables ➔ Selection ➔ Operators ➔ Count-controlled iteration	National curriculum links • Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures (e.g. lists, tables, or arrays); design and develop modular programs that use procedures or functions • Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem

	• Predict the outcome of a simple sequence that includes variables • Trace the values of variables within a sequence • Make a sequence that includes a variable • Define a condition as an expression that will be evaluated as either true or false • Identify that selection uses conditions to control the flow of a sequence • Identify where selection statements can be used in a program • Modify a program to include selection • Create conditions that use comparison operators (>,<,=) • Create conditions that use logic operators (and/or/not) • Identify where selection statements can be used in a program that include comparison and logical operators • Define iteration as a group of instructions that are repeatedly executed • Describe the need for iteration • Identify where count-controlled iteration can be used in a program • Implement count-controlled iteration in a program • Detect and correct errors in a program (debugging) • Independently design and apply programming constructs to solve a problem (subroutine, selection, count-controlled iteration, operators, and variables)		• Understand simple Boolean logic (e.g. and, or, and not) Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Networks: from semaphores to the internet *Spring 2*	• Define what a computer network is and explain how data is transmitted between computers across networks • Define 'protocol' and provide examples of non-networking protocols • List examples of the hardware necessary for connecting devices to networks • Compare wired to wireless connections and list examples of specific technologies currently used to implement such connections • Define 'bandwidth', using the appropriate units for measuring the rate at which data is transmitted, and discuss familiar examples where bandwidth is important • Define what the internet is • Explain how data travels between computers across the internet • Describe key words such as 'protocols', 'packets', and 'addressing' • Explain the difference between the internet, its services, and the World Wide Web • Describe how services are provided over the internet • List some of these services and the context in which they are used • Explain the term 'connectivity' as the capacity for connected devices ('Internet of Things') to collect and share information about me with or	**GCSE Link:** ➔ Network Topologies ➔ Network and Cyber Security for computer Science **Wider Curriculum Links:** ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ To understand the history of networks and telecommunications.	National curriculum links • Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems Education for a connected world links • Explain the term 'connectivity' as the capacity for connected devices ('internet of things') to collect and share information about me with or without my knowledge (including microphones, cameras and geolocation). • Describe how internet-connected devices can affect me.

	without my knowledge (including microphones, cameras, and geolocation) ● Describe how internet-connected devices can affect me ● Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together		
Year 7 — Programming II *Summer 1*	● Define a subroutine as a group of instructions that will run when called by the main program or other subroutines ● Define decomposition as breaking a problem down into smaller, more manageable sub problems ● Identify how subroutines can be used for decomposition ● Identify where condition-controlled iteration can be used in a program ● Implement condition-controlled iteration in a program ● Evaluate which type of iteration is required in a program ● Define a list as a collection of related elements that are referred to by a single name ● Describe the need for lists ● Identify when lists can be used in a program ● Use a list ● Decompose a larger problem into smaller sub problems ● Apply appropriate constructs to solve a problem	**GCSE Link:** ➔ Programming ➔ Algorithms **Wider Curriculum Links:** ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ Decomposition ➔ Subroutines ➔ Condition-controlled iteration ➔ Lists ➔ Problem solving	**National curriculum links** ● To use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; to make appropriate use of data structures (for example, lists, tables, or arrays); to design and develop modular programs that use procedures or functions ● To understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem ● To understand simple Boolean logic (for example, AND, OR, and NOT) ● To create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Modelling Data- Spreadsheets *Summer 2*	• Identify columns, rows, cells, and cell references in spreadsheet software • Use formatting techniques in a spreadsheet • Use basic formulas with cell references to perform calculations in a spreadsheet (+, -, *, /) • Use the autofill tool to replicate cell data • Explain the difference between data and information • Explain the difference between data and information • Explain the difference between primary and secondary sources of data • Collect data • Analyse data • Create appropriate charts in a spreadsheet • Use the functions SUM, COUNTA, MAX, and MIN in a spreadsheet • Analyse data • Use a spreadsheet to sort and filter data • Use the functions AVERAGE, COUNTIF, and IF in a spreadsheet • Use conditional formatting in a spreadsheet • Apply all of the spreadsheet skills covered in this unit	**GCSE Link:** ➔ Maths: Formulas ➔ Maths: Functions ➔ Maths: Identify Graphs and Diagrams **Wider Curriculum Links:** ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ Use cell references ➔ Use the autofill tool ➔ Format data ➔ Create formulas for add, subtract, divide, and multiply ➔ Create functions for SUM, COUNTA, AVERAGE, MIN, MAX, and COUNTIF ➔ Sort and filter data ➔ Create graphs ➔ Use conditional formatting	National curriculum links • Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems • Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users

Year 8 - Scheme of Learning

Word Rich - Oracy, Vocabulary, Reading, Writing	SMSC & Values	Careers & Employability	Enrichment & Cultural Capital	Equality, Diversity & Inclusivity
❖	•	•	•	•

Formal Assessments (Title/Date)	Blended Learning	Home Learning
❖ ❖ ❖	• Lessons and resources in google classroom	• Lessons and resources in google classroom

Unit of Work	Knowledge and *Skills*	Curriculum Links and Sequencing	National Curriculum
Y8 - Developing for the web *Autumn 1*	• 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 • Use search technologies effectively	**GCSE Link:** ➔ Introduction to Wed development ➔ Understand Html & CSS **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ **Searching** ➔ **Threats** ➔ **HTML and CSS**	National curriculum links • Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability.

	• 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 • Implement navigation to complete a functioning website • Complete summative assessment		
Representations – from clay to silicon *Autumn 2*	• 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 • 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	**GCSE Link:** ➔ Binary Representation **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links** ➔ Representing Data with Images and Sound ➔ How Computers Work: Demystifying Computation	National curriculum links **(Computing programmes of study: Key Stage 3)** • Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

	• Convert between different units and multiples of representation size • Provide examples of the different ways that binary digits are physically represented in digital devices • Apply all of the skills covered in this unit		
Year 8 – Vector graphics *Spring 1*	• 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 assess another pair's project work • Improve your own project work based on feedback • Complete a summative assessment	**GCSE Link:** ➔ Editing shapes and sizes ➔ Animations ➔ Photoshop **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC ➔ Art & Photography **Sequencing Links** ➔ Posters ➔ Logos ➔ Resize and edit shapes and images. ➔ Image manipulation.	**National curriculum links** • undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users • create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability

Introduction to Python programming *Spring 2*	• 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 • 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	**GCSE Link:** ➔ Computing: Python Programming **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ ***Computer Science Education: Perspectives on Teaching and Learning in School*,** ➔ **Effective computing pedagogy** ➔ **Programming Pedagogy in Secondary Schools: Inspiring Computing Teaching** ➔ **Programming 101: An Introduction to Python for Educators** ➔ **Scratch to Python: Moving from Block- to Text-based Programming**	National curriculum links **(Computing programmes of study: Key Stage 3)** **Aims** • can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation • can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems **Subject content** • use two or more programming languages, at least one of which is textual, to solve a variety of computational problems • understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem • understand how instructions are stored and executed within a computer system

			• design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
Year 8 – Computing systems *Summer 1*	●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’) ●Analyse how the hardware components used in computing systems work together in order to execute programs	**GCSE Link:** ➔ Computer Operating Systems ➔ Computer Infrastructure ➔ Understanding Maths and Logic in Computer Science. **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ The Computing Universe ➔ D is for Digital ➔ The Pattern on the Stone ➔ How Computers Work is a series of videos hosted by Khan Academy that help to visualise this unit’s content ➔ The ‘Revolution’ exhibition by the Computing History Museum is available online, providing an overview of the history of computing, with a wealth of annotated photographs and videos ➔ Effective computing pedagogy by the National Centre for Computing Education	**National curriculum links** **(Computing programmes of study: Key Stage 3)** **Aims** • can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation • can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems **Subject content** • understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming • understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems • understand how instructions are stored and executed within a computer system

	●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 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		
Year 8 – Mobile app development *Summer 2*	• Identify when a problem needs to be broken down • Implement and customise GUI elements to meet the needs of the user • 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 • Identify and fix common coding errors • Pass the value of a variable into an object • Establish user needs when completing a creative project • Apply decomposition to break down a large problem into more manageable steps • 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	**GCSE Link:** ➔ Mobile Application Development ➔ In app application and services **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links** ➔ Event handling ➔ Sequencing ➔ Variables ➔ Selection ➔ Operators	**National curriculum links** • Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems • Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables, or arrays]; design and develop modular programs that use procedures or functions • Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem • Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

	• 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 • 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		

Year 9 - Scheme of Learning				
Word Rich - Oracy, Vocabulary, Reading, Writing	**SMSC & Values**	**Careers & Employability**	**Enrichment & Cultural Capital**	**Equality, Diversity & Inclusivity**
❖	•	•	•	•
Formal Assessments (Title/Date)			**Blended Learning**	**Home Learning**
❖ ❖ ❖			• Lessons and resources in google classroom	• Lessons and resources in google classroom

Unit of Work	**Knowledge and *Skills***	**Curriculum Links and Sequencing**	**National Curriculum**

Cyber Security *Autumn1*	★ Explain the difference between data and information ★ Critique online services in relation to data privacy ★ Identify what happens to data entered online ★ Explain the need for the Data Protection Act ★ Recognise how human errors pose security risks to data ★ Implement strategies to minimise the risk of data being compromised through human error ★ Define hacking in the context of cyber security ★ Explain how a DDoS attack can impact users of online services ★ Identify strategies to reduce the chance of a brute force attack being successful ★ Explain the need for the Computer Misuse Act ★ List the common malware threats ★ Examine how different types of malware causes problems for computer systems ★ Question how malicious bots can have an impact on societal issues ★ Compare security threats against probability and the potential impact to organisations ★ Explain how networks can be protected from common security threats ★ Identify the most effective methods to prevent cyberattacks	**GCSE Link:** ➔ Cyber Security ➔ Online Fraud **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links** ➔ Profiling ➔ Data Protection Act ➔ Computer Misuse Act ➔ Hacking ➔ Malware ➔ Protection methods such as firewalls, anti-malware, and password authentication	**National curriculum links** ● Understand a range of ways to use technology safely, respectfully, responsibly, and securely, including protecting their online identity and privacy; recognise inappropriate content, contact, and conduct, and know how to report concerns **Education for a Connected World links** ● I can explain how contributors to social media may be 'social bots' ● I can explain what malware is and give some examples of how it operates and what its impact could be on a device or user (e.g. viruses, Trojans, ransomware) ● I can explain how to manage security software (e.g. anti-virus, security patches, adware blockers) on my devices and understand why regular updates are important ● I can explain how and assess when more secure use may require more advanced password management (e.g. dual-factor authentication, regular rolling, security questions, CAPTCHA, biometrics)

Python programming with sequences of data *Autumn 2*	• Write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements • Use selection (if-elif-else statements) to control the flow of program execution • Locate and correct common syntax errors • Create lists and access individual list items • Perform common operations on lists or individual items • Use iteration (while statements) to control the flow of program execution • Perform common operations on lists or individual items • Perform common operations on strings or individual characters • Use iteration (for statements) to iterate over list items • Perform common operations on lists or strings • Use iteration (for loops) to iterate over lists and strings • Use variables to keep track of counts and sums • Combine key programming language features to develop solutions to meaningful problems	**GCSE Link:** ➔ Computing: Python Programming II **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links:** ➔ ***Computer Science Education: Perspectives on Teaching and Learning in School*,** ➔ **Effective computing pedagogy** ➔ **Programming Pedagogy in Secondary Schools: Inspiring Computing Teaching**	**National curriculum links** **(Computing programmes of study: Key Stage 3)** **Aims** • can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation • can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems **Subject content** • use two or more programming languages, at least one of which is textual, to solve a variety of computational problems • understand how instructions are stored and executed within a computer system • understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem • design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems

Media – Animations *Spring 1*	**Blender projects (ncce.io/ks3-BlenderProjects)** ● Add, delete, and move objects ● Scale and rotate objects ● Use a material to add colour to objects ● Add, move, and delete key frames to make basic animations ● Play, pause, and move through the animation using the timeline ● Create useful names for objects ● Join multiple objects together using parenting ● Use edit mode and extrude ● Use loop cut and face editing ● Apply different colours to different parts of the same model ● Use proportional editing ● Use the knife tool ● Use subdivision ● Add and edit set lighting ● Set up the camera ● Compare different render modes ● Create a 3–10 second animation ● Render out the animation	**GCSE Link:** ➔ Editing shapes and sizes ➔ Animations ➔ Photoshop **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC ➔ Art & Photography **Sequencing Links** ➔ Posters ➔ Logos ➔ Resize and edit shapes and images. ➔ Image manipulation.	**National curriculum links** ● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability
Data Science *Spring 2*	● Define data science ● Explain how visualising data can help identify patterns and trends in order to help us gain insights ● Use an appropriate software tool to visualise data sets and look for patterns or trends	**GCSE Link:** ➔ Data Science **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC	**National curriculum links** ● Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals,

	• Recognise examples of where large data sets are used in daily life • Select criteria and use data set to investigate predictions • Evaluate findings to support arguments for or against a prediction • Define the terms 'correlation' and 'outliers' in relation to data trends • Identify the steps of the investigative cycle • Solve a problem by implementing steps of the investigative cycle on a data set • Use findings to support a recommendation • Identify the steps of the investigative cycle • Identify the data needed to answer a question defined by the learner • Create a data capture form • Describe the need for data cleansing • Apply data cleansing techniques to a data set • Visualise a data set • Visualise a data set • Analyse visualisations to identify patterns, trends, and outliers • Draw conclusions and report findings	**Sequencing Links** ➔ Profiling ➔ Data Protection Act ➔ Computer Misuse Act ➔ Hacking ➔ Malware ➔ Protection methods such as firewalls, anti-malware, and password authentication	including collecting and analysing data and meeting the needs of known users

Representations – going audio-visual *Summer 1*	• Describe how digital images are composed of individual elements • Recall that the colour of each picture element is represented using a sequence of binary digits • Define key terms such as 'pixels', 'resolution', and 'colour depth' • Describe how an image can be represented as a sequence of bits • Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity • Compute the representation size of a digital image, by multiplying resolution (number of pixels) with colour depth (number of bits used to represent the colour of individual pixels) • Describe the trade-off between representation size and perceived quality for digital images • Perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation • Explain how the manipulation of digital images amounts to arithmetic operations on their digital representation • Describe and assess the creative benefits and ethical drawbacks of digital manipulation (Education for a Connected World) • Recall that sound is a wave	**GCSE Link:** ➔ Binary Representation **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC **Sequencing Links** ➔ Representing Data with Images and Sound ➔ How Computers Work: Demystifying Computation	**National curriculum links** **(Computing programmes of study: Key Stage 3)** • Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

	• Explain the function of microphones and speakers as components that capture and generate sound • Define key terms such as 'sample', 'sampling frequency/rate', 'sample size' • Describe how sounds are represented as sequences of bits • Calculate representation size for a given digital sound, given its attributes • Explain how attributes such as sampling frequency and sample size affect characteristics such as representation size and perceived quality, and the trade-offs involved • Perform basic sound editing tasks using appropriate software and combine them in order to solve more complex problems requiring sound manipulation • Recall that bitmap images and pulse code sound are not the only binary representations of images and sound available • Define 'compression', and describe why it is necessary		
Physical computing *Summer 2*	• Describe what the micro: bit is • List the micro: bit's input and output devices • Use a development environment to write, execute, and debug a Python program for the micro: bit • Write programs that use the micro: bit's built-in input and output devices	**GCSE Link:** ➔ Artificial Intelligence ➔ Impact of AI Technology ➔ AI Robotics **Wider Curriculum Links:** ➔ Historical ➔ Cultural ➔ SMSC	National curriculum links **(Computing programmes of study: key stage 3)** **Aims** • Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation

	• Write programs that use GPIO pins to generate output and receive input • Write programs that communicate with other devices by sending and receiving messages wirelessly • Design a physical computing artefact purposefully, keeping in mind the problem at hand, the needs of the audience involved, and the available resources • Decompose the functionality of a physical computing system into simpler features • Implement a physical computing project, while following, revising, and refining the project plan • Implement a physical computing project, while following, revising, and refining the project plan	**Sequencing Links** ➔ The micro:bit Educational Foundation website (microbit.org) ➔ Introduction to Micro Python (ncce.io/mb-arm), a video series by Arm Education. ➔ Resources for schools (arm.com/resources/education/schools/content), by the Arm School Program ➔ *Computer Science Education: Perspectives on teaching and learning in school*, ➔ *Teaching Tech Together: How to create and deliver lessons that work and build a teaching community around them*, ➔ Effective computing pedagogy,	• Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems **Subject content** • Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems • Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem • Understand how instructions are stored and executed within a computer system • Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems