Progression of Skills and Knowledge
Computing

National Curriculum Requirements

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

Aims

The national curriculum for computing aims to ensure that all pupils:

- 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.
- Can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
- Are responsible, competent, confident and creative users of information and communication technology.

Pupils should be taught to:

- 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.
- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.
- Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Computing Systems	Creating Media A	Creating Media B	Programming A	Programming B	Data and Information

Year Group	Skills	Knowledge
Year 3	• identify input and output devices • explain how computer systems can change the way we work • explain how a computer network can be used to share information • explain how information is passed through multiple connections • explain the role of a switch, server, and wireless access point in a network • recognise that a network is made up of a number of components • identify how devices in a network are connected with one another • identify network devices around me • identify the benefits of computer networks • describe what an input is • explain that a process acts on the inputs • explain that an output is produced by the process • identify how changing the process can affect the output • explain that a computer system accepts an input and processes it to produce an output. • recognise that computers can be connected to each other • explain how networks can be connected to other networks	Connecting computers Know that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks.
	• recognise that an animation is made up of a sequence of images • recognise that an animation can be drawn images or captured photographs • recognise the relationship between frames and motion	Stop-frame animation Know how to capture and edit digital still images to produce a stop-frame animation that tells a story.

	• understand the terms ‘composition’, ‘stage’, and ‘capture area’. • deconstruct a known story (into characters, stages, key events) • recognise the need for consistency in working • recognise that a capturing device needs to be in a fixed position • know how to fix mistakes in captured images • recognise that impact of adding other media • use a computer to create an animation (tell a story) • set up a device to capture stop frame photos • capture a series of images • use tools (onion skinning) to review subject position • move a subject between captures • play a sequence of images back to review • remove images to improve an animation • add sound effects • add text (scenes, credits, captions) • play back and review a film • export a film	
	• recognise how text and images can be used together to convey information • recognise that a document is structured with placeholders • recognise that text can be edited • consider how different layouts can suit different purposes • consider the benefits of using a DTP application • combine text and graphics to share a message • Manage layout: • choose a suitable template for a particular purpose • adding, moving and replacing placeholders • page settings and orientation • add, resize and change the orientation of images • manipulate text: • use font size to amplify sections of text	Desktop publishing Know how to create documents by modifying text, images, and page layouts for a specified purpose.

	• use font colours • change the orientation of text • wrap text • arrange text and images • delete content • review a document	
	• explain that the order of commands can affect the outcome (same commands, different order -> same or different outcome) • identify that different sequences can achieve the same outcome • create a sequence of commands to produce a given outcome • build a sequence of commands • explain that a program has a start To explain what a sequence is • recognise that sequences can have an order • combine commands in a program To order commands in a program • identify that a program includes sequences of commands	Sequencing sounds Know how to create sequences in a block-based programming language to make music.
	• explain that the order of commands can affect the outcome — the same commands in a different order may produce the same or different outcomes • identify that different sequences can achieve the same outcome • create a sequence of commands to produce a given outcome • build a sequence of commands • explain that a program has a start • explain what a sequence is • recognise that sequences can have an order • combine commands in a program • order commands in a program • identify that a program includes sequences of commands	Events and actions in programs Know how to write algorithms and programs that use a range of events to trigger sequences of actions.

	• create questions with yes/no answers • select an attribute to separate objects into two similarly sized groups • identify the object attributes needed to collect relevant data • decide what data needs to be collected to answer a specific question • retrieve information from different levels of the branching database • relate two levels of a branching database using AND • explain that data can be used to answer questions • compare the information shown in a pictogram with a branching database • investigate questions with yes/no answers	Branching databases Know how to build and use branching databases to group objects using yes/no questions.
Year 4	• explain that the global interconnection of networks is the internet • describe how networks connect to other networks • outline how information can be shared via the World Wide Web • describe the types of content/media that can be added, created, and shared on the World Wide Web • explain how the content of the World Wide Web is created, owned, and shared by people • describe the current limitations of World Wide Web media • evaluate the reliability of content and the consequences of unreliable content • explain the benefits of the World Wide Web • recognise that the World Wide Web is part of the internet • know that the internet enables us to view the World Wide Web • explain that the World Wide Web comprises of websites and web pages • know how to access the World Wide Web • recognise the need for security on the internet	The internet Know that the internet is a network of networks including the WWW, and know why we should evaluate online content.

	• recognise that sound can be digitally recorded • recognise that some digital devices have microphones • recognise that recorded audio is stored as a file • recognise that audio can be edited and altered • recognise that sound can be layered • consider the results of editing choices made • record sound • press/tap buttons to start and stop recording • know where the microphone is on the device • locate recorded audio and playback • edit audio by selecting a section, deleting a section or apply effects to a section of audio • save/export an audio file	Audio production Know how to capture and edit audio to produce a podcast, ensuring that copyright is considered.
	• recognise that digital images can be manipulated • recognise that images can be changed for different purposes • use the most appropriate tool for a particular purpose • recognise that not all images are real • consider the impact of changes made on the quality of the image • use a computer to (further) manipulate images • open/retrieve an image • change the composition of an image: • arrange (rotate, flip) • crop • cut out a part • apply a change globally: • adjust colours • apply a filter • add effects • apply changes locally- retouch and reuse. • make additions by drawing, adding text and to add an element such as a border.	Photo editing Know how to manipulate digital images, and reflect on the impact of changes and whether the required purpose is fulfilled.
	• explain the importance of instruction order in a loop	Repetition in shapes Know a text-based programming language to explore count-controlled loops when drawing shapes.

	• create two or more sequences that run at the same time • recognise tools that enable more than one process to be run at the same time (concurrency) • recognise that not all tools enable more than one process to be run at once • relate what ‘repeat’ means • identify everyday tasks that include repetition as part of a sequence, e.g. brushing teeth, dance moves • list an everyday task as a set of instructions including repetition To explain that we can use a loop command in a program to repeat instructions • identify a loop within a program • explain that in programming there are indefinite loops and count-controlled loops • explain that an indefinite loop will run until the program is stopped • explain that you can program a loop to stop after a specific number of times • justify when to use a loop and when not to • plan a program that includes appropriate loops to produce a given outcome • use an indefinite loop to produce a given outcome • use a count-controlled loop to produce a given outcome • identify patterns in a sequence, e.g. ‘step 3 times’ means the same as ‘step, step, step’ • identify patterns in a sequence	
	• explain the importance of instruction order in a loop • create two or more sequences that run at the same time • recognise tools that enable more than one process to be run at the same time (concurrency) • recognise that not all tools enable more than one process to be run at once • relate what ‘repeat’ means	Repetition in games Know how to use a block-based programming language to explore count-controlled and infinite loops when creating a game

	• identify everyday tasks that include repetition as part of a sequence, e.g. brushing teeth, dance moves • list an everyday task as a set of instructions including repetition • explain that we can use a loop command in a program to repeat instructions • identify a loop within a program • explain that in programming there are indefinite loops and count-controlled loops • explain that an indefinite loop will run until the program is stopped • explain that you can program a loop to stop after a specific number of times • justify when to use a loop and when not to • plan a program that includes appropriate loops to produce a given outcome • use an indefinite loop to produce a given outcome • use a count-controlled loop to produce a given outcome identify patterns in a sequence, e.g. ‘step 3 times’ means the same as ‘step, step, step’ • identify patterns in a sequence	
	• suggest questions that can be answered using a given data set • identify the data that we need to answer questions • use a digital device to collect data automatically • recognise that a sensor can be used as an input device for data collection • use a larger data set to find information • export information in different formats • explain that a data logger captures ‘data points’ from sensors over time • identify that sensors are input devices • present data in a table • present data in a graph • choose how often to automatically collect data samples • use a computer program to sort data by one attribute	Data logging Know how and why data is collected over time, before using data loggers to carry out an investigation.

Year 5	• recognise that computers can be part of a system in an electronic device • recognise input, process, and output in larger computer systems • recognise how information is transferred across the internet recognise that connections between computers allow us to work together • explain that the internet lets people in different places work together • recognise that connections between computers allow us to access shared stored files • explain that the internet allows different media to be shared recognise that internet collaborations can be public or private • understand that computers can be connected together to form systems • see that computers communicate with other devices (including other computers) • evaluate different ways of working together • recognise the role of computer systems in our lives • explain that data is transferred in packets • recognise that data is transferred using agreed protocols (methods)	Systems and searching Know that IT systems in the world can enable searching on the internet.
	• recognise that tools can be changed to produce different outcomes • choose options to achieve a desired effect • recognise that an image comprises of separate objects • recognise that objects are layered • combine options to achieve a desired effect • consider the impact of choices made • recognise that objects can be modified in groups • recognise that vector images can be scaled without impact on quality • create graphical objects on a computer screen: • select a shape type to add to a drawing • select a line type to add to a drawing	Video production Know how to plan, capture, and edit video to produce a short film.

	• add text to a drawing • drag out an object on the page • duplicate an object • select an object • delete an object • modify an object by repositioning, rotating, resizing, recolouring and altering object proportions • select multiple objects • group objects • modify multiple objects • change the layers of an object	
	• recognise videos as moving pictures combined with audio • review existing video content • identify the key concepts of composition • plan a video production using a storyboard • recognise that some digital devices can capture video using a camera and microphone • recognise that video can be captured by a person operating a camera • review, identify good features of a video and identify how a video can be improved • recognise that video can be improved through editing • consider the results of choices I have made • use a computer to make a video • capture a video: • locate the function on the device to record the video • hold the device safely in landscape orientation • pan left/right up/down • focus, zoom and compose • use techniques to create special effects (e.g green screen) • press the start/stop button to end recording • play back video • edit a section of a video • delete a section of a video	<u>Introduction to vector graphics</u> Know how to create images in a drawing program by using layers and groups of objects.

	• save and export a video file	
	• experiment with a ‘repeat until’ loop • relate that a count-controlled loop contains a condition • relate that a conditional statement connects a condition to an outcome • explain that instructions in a program will produce specific outcomes • outline that a condition is something that can be either true or false • show that a condition can switch program flow in one of two ways • use a condition in an ‘if... then…’ statement to produce a given outcome • explain that program flow can branch according to a condition • define that conditional statements are used in computer programs • conclude that a loop can be used to repeatedly check whether a condition has been met • use a condition in an ‘if... then... else…’ statement to produce given outcomes • explain the importance of instruction order in ‘if... then…’ statements • explain the importance of instruction order in ‘if... then... else…’ statements • explain that a loop can stop when a condition is met, e.g. number of times • explain a sequence within a count- or event controlled loop • explain that a loop can stop when a condition is met, e.g. event • modify a count- or event-controlled loop • create a count- or event-controlled loop	<u>Selection in physical computing</u> Know how to explore conditions and selection using a programmable microcontroller.
	• experiment with a repeat-until loop • relate that a count-controlled loop contains a condition	<u>Selection in quizzes</u> Know how to explore selection in programming to design and code an interactive quiz.

	• relate that a conditional statement connects a condition to an outcome • explain that instructions in a program will produce specific outcomes • outline that a condition is something that can either be true or false • show that a condition can switch program flow in one of two ways • use a condition in an ‘if... then…’ statement to produce a given outcome • explain that program flow can branch according to a condition • define that conditional statements are used in computer programs • conclude that a loop can be used to repeatedly check whether a condition has been met • use a condition in an ‘if... then... else...’ statement to produce given outcomes • explain the importance of instruction order in ‘if... then…’ statements • explain the importance of instruction order in ‘if... then... else...’ statements • explain that a loop can stop when a condition is met, eg number of times • explain a sequence within a count-controlled or event-controlled loop • explain that a loop can stop when a condition is met, eg an event • modify a count-controlled or event controlled loop • create a count-controlled or event-controlled loop Repetition Selection Sequence	
	• design an approach to answer a question using a database • explain that a computer program can be used to organise data	Flat-file databases Know that a database can be used to order data and create charts to answer questions.

	• choose which attribute and value to search by to answer a given question (operands) • explain that computer programs can be used to compare data visually • select an appropriate graph to visually compare data • choose suitable ways to present information to other people • navigate a flat-file database • choose multiple criteria to search data to answer a given question (AND and OR) • design a structure for a flat-file database • choose different ways to view data • choose which attribute to sort data by to answer a given question • explain that tools can be used to select data to answer questions • explain that we present information to communicate a message • outline how ordering data allows us to answer some questions • outline how operands can be used to filter data To outline how 'AND' and 'OR' can be used to refine data selection • ask questions that need more than one attribute to answer	
Year 6	• explain how search results are selected • identify some of the limitations of search engines • demonstrate that different search terms produce different results • identify that results from search engines can include adverts, and that the adverts can be targeted • discuss the opportunities that technology offers for communication • list methods of communicating using the internet • recognise that some information is not searchable • explain why search engines exist • recognise that there are a number of search engines	Communication and collaboration Know how data is transferred by working collaboratively online.

	• define ‘communication’ • evaluate different methods of online communication • recall how to use a search engine • explain that search terms need to be chosen carefully • explain the role of web crawlers • decide what I should/should not share • evaluate the results of search terms • compare the results from different search engines • explain how ranking is determined by rules, and that different search engines use different rules • explain why search engines create indexes, and that they are different for each search engine • define the purpose of an index • explain that search results are ordered, and this is known as ranking • explain why the order of results is important, and to whom • explain how search engines make money by selling advertising space • explain that ranking narrows down the search results returned from the index, which makes it more useful • examine the role of the searcher, search engine, and content creator in the searching process Learning graph • identify different ways to communicate without technology To choose an appropriate method of internet communication for a given purpose • explain that communicating through the internet can be public or private • explain which types of media can be shared through the internet • classify internet communication by messenger and recipient or audience	
	• recognise that 3D objects comprise length, width, and height (depth)	Webpage creation Know how to design and create webpages, giving consideration to copyright, aesthetics, and navigation.

	• recognise the similarities and differences between real-life 3D and virtual 3D • recognise the role of scale in design • recognise the differences when working in 3D compared with 2D • recognise that structures can be broken down into a collection of 3D objects • recognise that blank objects must be used as placeholders to create holes • create 3D graphical objects on a computer screen • alter the view of the 3D space • place a 3D object in a 3D space • select, duplicate and delete an object • modify an object by: • To reposition, rotate, resize and recolour in three dimensions • select, group and modify multiple objects	
	• review an existing website (navigation bars, header) • recognise the relationship between HTML and visual display • recognise components of a web page layout • consider the ownership and use of images (copyright) • recognise the need to preview pages (different screens/devices) • recognise the need for a navigation path • recognise the implications of linking by content owned by others • create a web page • add text to a web page • change the appearance of text • change the position of text • add images to a web page • add other content • preview a page (different screen sizes) • add additional pages • insert hyperlinks between pages • insert hyperlinks to another site	3D modelling Know how to plan, develop, and evaluate 3D computer models of physical objects.

	• embed content	
	• recognise that the value of a variable can be updated • define a program variable as a placeholder in memory for a single value • explain that a variable has a name and a value • explain the importance of setting up a variable at the start of a program (initialisation) • recognise that a variable can be set as a constant (fixed value) • experiment with the value of an existing variable • define 'variable' as something that is changeable • explain that a variable is something that we can use in a program, e.g. 'score' • identify a variable in an existing program • identify that variables can hold numbers (integers) or letters (strings) To decide where in a program to set a variable • identify examples of information that is variable, e.g. a football score during a match • recognise that the value of a variable can be used by a program • use the same variable in more than one location in a program • explain that there is only one value for a variable at any one time • explain that the name of a variable is meaningless to the computer • explain that if you read a variable, the value remains • explain that if you change the value of a variable, you cannot access the previous value (cannot undo) • use an event in a program to update a variable • use a variable in a conditional statement to control the flow of a program • update a variable with a user input • define the way that a variable is changed and that the name of a variable needs to be unique	Variables in games Know how to explore variables when designing and coding a game.

	• choose a name that identifies the role of a variable to make it more usable (to humans)	
	• recognise that the value of a variable can be updated • define a program variable as a placeholder in memory for a single value • explain that a variable has a name and a value • explain the importance of setting up a variable at the start of a program (initialisation) • recognise that a variable can be set as a constant (fixed value) • experiment with the value of an existing variable • define 'variable' as something that is changeable • explain that a variable can be used in a program, e.g. 'score' • identify a variable in an existing program • identify that variables can hold numbers (integers) or letters (strings) To decide where in a program to set a variable • identify examples of information that is variable, e.g. a football score during a match • recognise that the value of a variable can be used by a program • use the same variable in more than one location in a program • explain that there is only one value for a variable at any one time • explain that the name of a variable is meaningless to the computer • explain that if you read a variable, the value remains • explain that if you change the value of a variable, you cannot access the previous value (cannot undo) • use an event in a program to update a variable • use a variable in a conditional statement to control the flow of a program • update a variable with a user input • explain that the name of a variable needs to be unique	Sensing movement Know how to design and code a project that captures inputs from a physical device.

	• define the way that a variable is changed • choose a name that identifies the role of a variable to make it more usable (to humans)	
	• propose simple, relevant questions that can be answered using data • identify questions that can be answered using data • explain that objects/artefacts can be described using data • explain what an item of data is • explain that computers deal with different data types in different ways • explain why data should be organised • choose suitable ways to represent data • evaluate results in comparison to the question asked • outline that there are different software tools to work with data recognise that changing inputs also changes outputs • recognise that data can be calculated using different operations • apply formulas to data, including duplication • explain that formulas can be used to produce calculated data • outline what makes good questions to answer with data	Introduction to spreadsheets Know how to answer questions by using spreadsheets to organise and calculate data.