Marsh Green Primary School

COMPUTING CURRICULUM

Computing programmes of study:
key stages 1 and 2 National curriculum in England

Purpose of study

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.

Attainment targets

By the end of each key stage, pupils are expected to know, apply and understand the matters, skills and processes specified in the relevant programme of study. Schools are not required by law to teach the example content in [square brackets].

Key stage 1

Pupils should be taught to:

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.

- create and debug simple programs.

- use logical reasoning to predict the behaviour of simple programs.

- use technology purposefully to create, organise, store, manipulate and retrieve digital content.

- recognise common uses of information technology beyond school.

- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

Key stage 2

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.

Curriculum design

The approach

Coherence and flexibility

The Teach Computing Curriculum is structured in units. For these units to be coherent, the lessons within a unit must be taught in order. However, across a year group, the units themselves do not need to be taught in order, with the exception of 'Programming' units, where concepts and skills rely on prior learning and experiences.

Knowledge organisation

The Teach Computing Curriculum uses the National Centre for Computing Education's computing taxonomy to ensure comprehensive coverage of the subject. This has been developed through a thorough review of the KS1-4 computing programme of study, and the GCSE and A level computer science specifications across all awarding bodies. All learning outcomes can be described through a high-level taxonomy of ten strands, ordered alphabetically as follows:

- **Algorithms** — Be able to comprehend, design, create and evaluate algorithms

- **Computer networks** — Understand how networks can be used to retrieve and share information, and how they come with associated risks

- **Computer systems** — Understand what a computer is, and how its constituent parts function together as a whole

- **Creating media** — Select and create a range of media including text, images, sounds and video

- **Data and information** — Understand how data is stored, organised, and used to represent real-world artefacts and scenarios

- **Design and development** — Understand the activities involved in planning, creating, and evaluating computing artefacts

- **Effective use of tools** — Use software tools to support computing work

- **Impact of technology** — Understand how individuals, systems, and society as a whole interact with computer systems

- **Programming** — Create software to allow computers to solve problems

- **Safety and security** — Understand risks when using technology, and how to protect individuals and systems

The taxonomy provides categories and an organised view of content to encapsulate the discipline of computing. Whilst all strands are present at all phases, they are not always taught explicitly.

For these units to be coherent, the lessons within a unit must be taught in order. However, across a year group, the units themselves do not need to be taught in order, with the exception of 'Programming' units, where concepts and skills rely on prior learning and experience.

Spiral curriculum

The units for key stages 1 and 2 are based on a spiral curriculum. This means that each of the themes is revisited regularly (at least once in each year group), and pupils revisit each theme through a new unit that consolidates and builds on prior learning within that theme.

This style of curriculum design reduces the amount of knowledge lost through forgetting, as topics are revisited yearly. It also ensures that connections are made even if different teachers are teaching the units within a theme in consecutive years.

Physical computing

The Teach Computing Curriculum acknowledges that physical computing plays an important role in modern pedagogical approaches to computing, both as a tool to engage pupils and as a strategy to develop pupils understanding in more creative ways. Additionally, physical   computing supports and engages a diverse range of pupils in tangible and challenging tasks.

The physical computing units in key stage 1 are:

- Year 1 – Programming A - Moving a robot

- Year 2 – Programming A - Robot algorithms

Your local Computing Hub may be able loan you the kit you need to teach the physical computing units from our curriculum   (ncce.io/hubs).

Online safety

The unit overviews for each unit show the links between the content of the lessons and the national curriculum and Education for a Connected World framework (ncce.io/ efacw). These references have been provided to show where aspects relating to online safety, or digital citizenship, are covered within the Teach Computing curriculum. Not all of the objectives in the Education for a Connected World framework are covered in the Teach Computing curriculum as some are better suited to personal, social, health, and economic (PSHE) education; spiritual, moral, social, and cultural (SMSC) development; and citizenship. However, the coverage required for the computing national curriculum is provided.

Schools should decide for themselves how they will ensure that online safety is being managed effectively in their setting, as the scope of this is much wider than just curriculum content.

EYFS	Autumn	Spring	Summer

Key Skills & Knowledge	Introduction to use of the mouse/track pad & experimenting with drawing		Introduction to Online Safety & looking after Hardware		Introduction to Programmable Toys	
	- To know how to 'hold' a mouse. - To be able to move a tack pad. - To know how to use a laptop touchpad - To be able to select colours. - To be able to mark make purposefully on the screen. - To be able to control the pencil width. - To be able to choose tools to experiment with. - To be able to use the undo button. - To be able to erase parts of pictures. - To be able to draw using a touch screen. - To be able to draw using mouse control. - REC To log on and off a computer.		- To be able to explain what 'private' means when using technology. - To be able to express how it feels to be uncomfortable with something. - To be able to name 5 people who can help with negative feelings. - To be able to think about how to show kindness to others. - To begin to be aware of the impact of a lot of screen time. - To be able to take appropriate actions before using technology. - To be able to understand why food should be kept away from devices. - To know safe ways to transport portable devices. - To be able to relate being gentle and sharing to the use of devices. - To be able to understand what technology is. - To be able to identify the main parts of a computer.		- To be able to describe a route taken by another person while it is being enacted. - To be able to follow a route taken by another person after it has been enacted. - To plan routes for toy vehicles. - To follow a plan for a toy vehicle. - To use the buttons on a floor robot to make it move. - To purposefully use the buttons on a floor robot to make it move one step at a time. - To be able to input a program of 2 or 3 steps into a floor robot and then run the program to make it move. - To be able to interpret simple instructions to predict the outcome. - To be able to plan and input instructions for a floor robot one step at a time. - To be able to plan and input instructions for a floor robot more than one step at a time.	

E-SAFETY	Online Reputation/Self-image & identity	Managing Online Information	Privacy & Security/Copyright & Ownership	Online Relationships & Bullying	Health, Wellbeing & Lifestyle	Review
	To recognise, online or offline, that anyone can say 'no to somebody who makes them feel sad, uncomfortable, embarrassed or upset. To identify ways that I can put information on the internet.	To talk about how to use the internet as a way of finding information online. To identify devices I could use to access information on the internet.	To identify some simple examples of my personal information (e.g. name, address, birthday, age, location). To describe who would be trustworthy to share this information with; I can explain why they are trusted. To know that work I create belongs to me. To name my work so that others know it belongs to me.	To recognise some ways in which the internet can be used to communicate To give examples of how I (might) use technology to communicate with people I know. To describe ways that some people can be unkind online. To offer examples of how this can make others feel	To identify rules that help keep us safe and healthy in and beyond the home when using technology. To give some simple examples of these rules.	

| VOCAB | Mouse, Scroll roller, Cursor , Screen, Touch pad, Keyboard, Keys , Delete key , Arrow keys, Enter key, Undo, Login | | Private, Screen time, Safe | | Up, down, left, right, path | |

Continuous Provision
To have role play opportunities for exploring technology around them:
  ➢ in the home such as toy microwave, cameras & clocks and use of Mash cams to talk about use of technology in role as a chef or refuse collector, in the outdoors such as traffic lights, walkie talkies and construction sites or magnifying tools and use of Mash cams to talk about outdoor technology such as a farmer or police officer, in the wider world such as a toy shopping till, calculator, barcode reader, QR code, spacecraft or doctors set and use Mash cams to talk.

		Long Term Overview			
		Key Stage One			
	Autumn		Spring		Summer	
1	Technology around us	Digital painting	Moving a robot	Group data	Digital writing	Programming quizzes
2	Information technology around us	Digital photography	Robot algorithms	Digital music	Pictograms	Programming quizzes
		Key stage Two			
	Autumn		Spring		Summer	
3	Connecting computers	Sequencing sounds	Stop frame animations	Branching database	Desktop publishing	Events and action in programs
4	The internet	Repetition in games	Repetition in shapes	Data logging	Photo editing	Audio production
5	Systems and searching	Selection in quizzes	Selection in physical computing	Flat file database	Introduction to vector graphics	Video production
6	Communication and collaboration	Variables in games	Web page creation	Spreadsheets	3D Modelling	Sensing movement

Teach Computing Curriculum overview

Brief overview

	Computing systems and networks[1]	Creating media	Programming A	Data and information	Creating media	Programming B
Year 1	Technology around us (1.1)*	Digital painting (1.2)	Moving a robot (1.3)	Grouping data (1.4)	Digital writing (1.5)	Programming animations (1.6)
Year 2	Information technology around us (2.1)	Digital photography (2.2)	Robot algorithms (2.3)	Pictograms (2.4)	Digital music (2.5)	Programming quizzes (2.6)

[1] Networks are not part of the key stage 1 national curriculum for computing, but the title is used as a strand across primary

*The numbers in brackets are a 'quick code' reference for each unit, e.g. 1.3 refers the to the third year one unit in the recommended teaching order.

Unit summaries

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 1	**Technology around us** Recognising technology in school and using it responsibly.	**Digital painting** Choosing appropriate tools in a program to create art, and making comparisons with working non-digitally.	**Moving a robot** Writing short algorithms and programs for floor robots, and predicting program outcomes.	**Grouping data** Exploring object labels, then using them to sort and group objects by properties.	**Digital writing** Using a computer to create and format text, before comparing to writing non-digitally.	**Programming animations** Designing and programming the movement of a character on screen to tell stories.
Year 2	**Information technology around us** Identifying IT and how its responsible use improves our world in school and beyond.	**Digital photography** Capturing and changing digital photographs for different purposes.	**Robot algorithms** Creating and debugging programs, and using logical reasoning to make predictions.	**Pictograms** Collecting data in tally charts and using attributes to organise and present data on a computer.	**Digital music** Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.	**Programming quizzes** Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.

National Curriculum Coverage — Years 1 and 2	1.1 Technology around us	1.2 Digital painting	1.3 Moving a robot	1.4 Grouping data	1.5 Digital writing	1.6 Programming animations	2.1 Information technology around us	2.2 Digital photography	2.3 Robot algorithms	2.4 Pictograms	2.5 Digital music	2.6 Programming quizzes
Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions.			✓			✓			✓			✓
Create and debug simple programs			✓			✓			✓			✓
Use logical reasoning to predict the behaviour of simple programs			✓			✓			✓			✓
Use technology purposefully to create, organise, store, manipulate, and retrieve digital content	✓	✓		✓	✓		✓	✓		✓	✓	✓
Recognise common uses of information technology beyond school	✓		✓				✓	✓				
use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.	✓			✓	✓		✓	✓	✓	✓		

Vocabulary Work Bank KS1

Year 1		
Computing systems and networks – Technology around us	**Creating media – Digital painting**	**Creating media – Digital writing**
technology, computer, mouse, trackpad, keyboard, screen, double-click, typing.	paint program, tool, paintbrush, erase, fill, undo, shape tools, line tool, fill tool, undo tool, colour, brush style, brush size, pictures, painting, computers	word processor, keyboard, keys, letters, type, numbers, space, backspace, text cursor, capital letters, toolbar, bold, italic, underline, mouse, select, font, undo, redo, format, compare, typing, writing.
Data and information – Grouping	**Programming A – Moving a robot**	**Programming B – Programming animations**
object, label, group, search, image, property, colour, size, shape, value, data set, more, less, most, fewest, least, the same	Bee-Bot, forwards, backwards, turn, clear, go, commands, instructions, directions, left, right, route, plan, algorithm, program.	ScratchJr, command, sprite, compare, programming, area, block, joining, start, run, program, background, delete, reset, algorithm, predict, effect, change, value, instructions, design.

Year 2		
Computing systems and networks – Information technology around us	**Creating media – Digital music**	**Creating media – Digital photography**

Information technology (IT), computer, barcode, scanner/scan	music, quiet, loud, feelings, emotions, pattern, rhythm, pulse, pitch, tempo, rhythm, notes, create, emotion, beat, instrument, open, edit.	device, camera, photograph, capture, image, digital, landscape, portrait, framing, subject, compose, light sources, flash, focus, background, editing, filter, format, framing, lighting,
Data and information – Pictograms	**Programming A – Robot algorithms**	**Programming B – Programming quizzes**
more than, less than, most, least, common, popular, organise, data, object, tally chart, votes, total, pictogram, enter, data, compare, objects, count, explain, attribute, group, same, different, conclusion, block diagram, sharing	instruction, sequence, clear, unambiguous, algorithm, program, order, prediction, artwork, design, route, mat, debugging, decomposition	sequence, command, program, run, start, outcome, predict, blocks, design, actions, sprite, project, modify, change, algorithm, build, match, compare, debug, features, evaluate, decomposition, code.

KS2

Teach Computing Curriculum overview

	Computing Systems and Networks	Creating Media	Programming A	Data and Information	Creating Media	Programming B
Year 3	Connecting computers (3.1)*	Stop-frame animation (3.2)	Sequencing sounds (3.3)	Branching databases (3.4)	Desktop publishing (3.5)	Events and actions in programs (3.6)
Year 4	The Internet (4.1)	Audio production (4.2)	Repetition in shapes (4.3)	Data logging (4.4)	Photo editing (4.5)	Repetition in games (4.6)
Year 5	Systems and searching (5.1)	Video production (5.2)	Selection in physical computing (5.3)	Flat-file databases (5.4)	Introduction to vector graphics (5.5)	Selection in quizzes (5.6)
Year 6	Communication and collaboration (6.1)	Web page creation (6.2)	Variables in games (6.3)	Spreadsheets (6.4)	3D modelling (6.5)	Sensing movement (6.6)

*The numbers in brackets are a 'quick code' reference for each unit, e.g. 1.3 refers the to the third year one unit in the recommended teaching order.

Unit summaries

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 3	**Connecting computers** Identifying that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks	**Stop-frame animation** Capturing and editing digital still images to produce a stop frame animation that tells a story	**Sequencing sounds** Creating sequences in a block-based programming language to make music.	**Branching databases** Building and using branching databases to group objects using yes/no questions.	**Desktop publishing** Creating documents and modifying text, images and page layouts for a specific purpose.	**Events and actions in programs** Writing algorithms and programs that use a range of events to trigger sequences of actions.
Year 4	**The internet** Recognising that the internet is a network of networks including the WWW, and why we should evaluate online content.	**Audio production** Capturing and editing audio to produce a podcast, ensuring that copyright is considered.	**Repetition in shapes** Using a text-based programming language to explore count-controlled loops when drawing shapes.	**Data logging** Recognising how and why data is collected over time, before using data loggers to carry out an investigation,	**Photo editing** Manipulating digital images, and reflecting on the impact of the changes and whether the required purpose is fulfilled,	**Repetition in games** Using a block-based programming language to explore count-controlled and infinite loops when creating a game.

Unit summaries

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 5	**Systems and searching** Recognising IT systems in the world and how some can enable searching on the internet.	**Video production** Planning, capturing, and editing video to produce a short film.	**Selection in physical computing** Exploring conditions and selection using a programmable microcontroller.	**Flat-file databases** Using a database to order data and create charts to answer questions.	**Introduction to vector graphics** Creating images in a drawing program by using layers and groups of objects.	**Selection in quizzes** Exploring selection in programming to design and code an interactive quiz.
Year 6	**Communication and collaboration** Exploring how data is transferred by working collaboratively online.	**Webpage creation** Designing and creating webpages, giving consideration to copyright, aesthetics and navigation.	**Variables in games** Exploring variables when designing and coding a game.	**Introduction to spreadsheets** Answering questions by using spreadsheets to organise and calculate data.	**3D modelling** Planning, developing, and evaluation 3D computer models of physical objects.	**Sensing movement** Designing and coding a project that captures inputs from physical devices.

National Curriculum Coverage – Years 3 and 4	3.1 Connecting computers	3.2 Stop-frame animation	3.3 Sequencing sounds	3.4 Branching databases	3.5 Desktop publishing	3.6 Events and actions in programs	4.1 The internet	4.2 Audio production	4.3 Repetition in shapes	4.4 Data logging	4.5 Photo editing	4.6 Repetition in games
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.		✓		✓			✓	✓			✓	

National Curriculum Coverage – Years 5 and 6	5.1 systems and searching	5.2 Video production	5.3 Selection in physical computing	5.4 Flat-file database	5.5 Introduction to vector graphics	5.6 Selection in quizzes	6.1 Communication and collaboration	6.2 Webpage creation	6.3 Variables in games	6.4 Introduction to spreadsheets	6.5 3D modelling	6.6 Sensing movement
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.	✓	✓						✓	✓		✓	

Vocabulary List KS2

Year 3		
Computing systems and networks – Connecting computers	**Creating Media – Desktop publishing**	**Creating Media – Stop-frame animation**
digital device, input, process, output, program, digital, non-digital, connection, network, switch, server, wireless access point, cables, sockets	text, images, advantages, disadvantages, communicate, font, style, landscape, portrait, orientation, placeholder, template, layout, content, desktop publishing, copy, paste, purpose, benefits.	animation, flip book, stop-frame, frame, sequence, image, photograph, setting, character, events, onion skinning, consistency, evaluation, delete, media, import, transition.
Data and Information – Branching databases	**Programming A – Sequencing sounds**	**Programming B – Events and actions in programs**
attribute, value, questions, table, objects, branching, database, objects, equal, even, separate, structure, compare, order, organise, selecting, information, decision tree.	Scratch, programming, blocks, commands, code, sprite, costume, stage, backdrop, motion, turn, point in direction, go to, glide, sequence, event, task, design, run the code, order, note, chord, algorithm, bug, debug, code.	motion, event, sprite, algorithm, logic, move, resize, extension block, pen up, set up, pen, design, action, debugging, errors, setup, code, test, debug, actions.

Year 4		
Computing systems and networks – Connecting computers – The internet	**Creating Media – Audio production**	**Creating Media – Photo editing**
internet, network, router, security, switch, server, wireless access point (WAP), website, web page, web address, routing, web browser, World Wide Web, content, links, files, use, download, sharing, ownership, permission, information, accurate, honest, content, adverts	audio, microphone, speaker, headphones, input device, output device, sound, podcast, edit, trim, align, layer, import, record, playback, selection, load, save, export, MP3, evaluate, feedback.	image, edit, digital, crop, rotate, undo, save, adjustments, effects, colours, hue, saturation, sepia, vignette, image, retouch, clone, select, combine, made up, real, composite, cut, copy, paste, alter, background, foreground, zoom, undo, font.
Data and Information – Data logging	**Programming A – Repetition in shapes**	**Programming B – Repetition in games**
data, table, layout, input device, sensor, logger, logging, data point, interval, analyse, dataset, import, export, logged, collection, review, conclusion.	Logo (programming environment), program, turtle, commands, code snippet, algorithm, design, debug, pattern, repeat, repetition, count-controlled loop, value, trace, decompose, procedure.	Scratch, programming, sprite, blocks, code, loop, repeat, value, infinite loop, count-controlled loop, costume, repetition, forever, animate, event block, duplicate, modify, design, algorithm, debug, refine, evaluate.

Year 5		
Computing systems and networks – systems and searching	**Creating Media – Introduction to vector graphics**	**Creating Media – Video production**
system, connection, digital, input, process, storage, output, search, search engine, refine, index, bot, ordering, links, algorithm, search engine optimisation (SEO), web crawler, content creator, selection, ranking.	vector, drawing tools, object, toolbar, vector drawing, move, resize, colour, rotate, duplicate/copy, zoom, select, align, modify, layers, order, copy, paste, group, ungroup, reuse, reflection	video, audio, camera, talking head, panning, close up, video camera, microphone, lens, mid-range, long shot, moving subject, side by side, angle (high, low, normal), static, zoom, pan, tilt, storyboard, filming, review, import, split, trim, clip, edit, reshoot, delete, reorder, export, evaluate, share.
Data and Information – Flat–file databases	**Programming A – Selection in physical computing**	**Programming B – Making Quizzes**
database, data, information, record, field, sort, order, group, search, value, criteria, graph, chart, axis, compare, filter, presentation.	microcontroller, USB, components, connection, infinite loop, output component, motor, repetition, count-controlled loop, Crumble controller, switch, LED, Sparkle, crocodile clips, connect, battery box, program, condition, Input, output, selection, action, debug, circuit, power, cell, buzzer	Selection, condition, true, false, count-controlled loop, outcomes, conditional statement, algorithm, program, debug, question, answer, task, design, input, implement, test, run, setup, operator

Year 6		
Computing systems and networks – Communication and collaboration	**Creating media – Webpage creation**	**Creating Media 3D Modelling**
communication, protocol, data, address, Internet Protocol (IP), Domain Name Server (DNS), packet, header, data payload, chat, explore, slide deck, reuse, remix, collaboration, internet, public, private, one-way, two-way, one-to-one, one-to-many.	website, web page, browser, media, Hypertext Markup Language (HTML), logo, layout, header, media, purpose, copyright, fair use, home page, preview, evaluate, device, Google Sites, breadcrumb trail, navigation, hyperlink, subpage, evaluate, implication, external link, embed.	TinkerCAD, 2D, 3D, shapes, select, move, perspective, view, handles, resize, lift, lower, recolour, rotate, duplicate, group, cylinder, cube, cuboid, sphere, cone, prism, pyramid, placeholder, hollow, choose, combine, construct, evaluate, modify.
Data and Information – Introduction to spreadsheets	**Programming – Variables in games**	**Programming – Sensing movement**
data, collecting, table, structure, spreadsheet, cell, cell reference, data item, format, formula, calculation, spreadsheet, input, output, operation, range, duplicate, sigma, propose, question, data set, organised, chart, evaluate, results, sum, comparison, software, tools.	variable, change, name, value, set, design, event, algorithm, code, task, artwork, program, project, code, test, debug, improve, evaluate, share, assign, declare	Micro:bit, MakeCode, input, process, output, flashing, USB, trace, selection, condition, if then else, variable, random, sensing, accelerometer, value, compass, direction, navigation, design, task, algorithm, step counter, plan, create, code, test, debug.

Adapting the curriculum for pupils with SEND

The Teach Computing Curriculum has been written to support all pupils, with units containing a number of scaffolding activities and utilising effective pedagogies to ensure high quality teaching. However, you may still need to adapt resources to enable some of your pupils, for example those with special educational needs and disabilities (SEND), to access lessons fully.

The following principles will help you make adaptations that benefit all learners, and these will be more effective if you identify clearly what it is your individual pupils need help with - do they have poor working memory that means that following instructions is more difficult, or do they need help to stay focussed when completing projects?

1. Identify essential learning and misconceptions: Determine the key learning in each unit that every child should know. Provide repeated opportunities for pupils to revisit this content in different ways. Identify any likely misconceptions and address these explicitly in lessons. For example, in the year 3 Animation unit, pupils tend to move characters too far between frames, so ensure this is highlighted and modelled well.

2. Pre-teach key vocabulary: Pre-teach the essential vocabulary for each unit, provide learners with a word list supported by images and use the vocabulary regularly throughout the unit with a consistent definition. Concentrate on a small number of terms and consider using a graphic organiser to highlight relationships between concepts, e.g. the Frayer model.

3. Create step-by-step instructions: Break down complex tasks and routine skills for using software and hardware into smaller steps and create pictorial instructions for children to follow. For example, in year 4 Audio Editing, you could create a handout with a sequence of instructions for trimming audio clips in Audacity based on the video guide.

4. Provide templates: In Creating Media or Data & Information unts, support task completion by providing a template for pupils to modify – removing the fear of the blank page and helping to build confidence. For example, in the year 6 Web Page Creation unit, you could set up a simple site with pages and navigation for pupils to fill in the content.

5. Consider non-computing barriers: Consider if difficulties in other areas, such as writing or maths, present barriers to completing a task and if so, modify the task to help mitigate these. For example, in the year 6 Communication and Collaboration unit, during lesson 3 where pupils work collaboratively to create slides, they could dictate content into the document rather than type it.

6. Use the PRIMM framework or Parson's problems: In programming units, add extra scaffolding using PRIMM and Parson's problems. Some pupils may not be able to create a program, but they can practise reading and exploring code in a working program, then modify it to make it more personalised. For example, in the year 4 Repetition in Games unit, for the final task learners can modify the bat catching game by changing the backdrop, adding a new costume to the sprites to change their appearance, and adding a different sound. The aim is to remove these scaffolds as children develop their skills, but some learners may not become fully independent.

7. Harness pupils' special interests: Increase engagement and make learning more relevant by incorporating pupils' special interests. This is also important in terms of culturally relevant pedagogy. For example, in the year 5 Vector Drawing unit, pupils could use what they have learnt to create a logo for their favourite sports team.

8. Use unplugged activities and the semantic wave: We can use unplugged activities to help make computing concepts more relevant and understandable for learners. However, it is very important to 'repack' the knowledge of the abstract concept so that learners understand what it means in a wider context and they can use the technical language. For example, in the year 5 Selection in Physical Computing unit pupils consider examples of selection in everyday life. Children then need to see the link between these examples and how this is used in a program and have the opportunity to use the key language in context. Pupils with SEND may need repeated examples and smaller steps to repack the knowledge, e.g. you could provide learners with some printed selection blocks from the Crumble software

9. Support planning: Break down the planning process into smaller parts which can be ticked off as each one is completed, and provide a planning scaffold for learners where required. For example, when planning out algorithms for Scratch, provide printed versions of the blocks to manipulate and order, to help pupils to focus on only the code required

10. Reinforce digital skills: A significant barrier to accessing the whole computing curriculum is a lack of key digital skills, for example being able to log on to a computer and use the keyboard effectively. Time spent revisiting digital skills across all units is important to develop fluency. Some pupils may also benefit from extra time to practise these skills in small groups, or may need image-supported help sheets to support specific repeated tasks, such as saving work.

It is important that your adaptations are informed by effective formative assessment to identify any gaps in learning and the approach which may support with these.

A further resource which can support you is the Universal Design for Learning Framework from CAST which outlines a number of approaches to include all learners in lessons by providing multiple means of engagement, representation, action and expression.

For support from fellow teachers with individual units, head over to the STEM community and join the discussions about adapting units for pupils with SEND.

Finally, there is CPD available to support you further. Complete this online course to improve your knowledge: Creating an Inclusive Classroom: Approaches to Supporting Learners with SEND in Computing or attend the face-to-face course: Inclusive Computing in Primary Schools.