# Computing

Alice Ingham

**‘The purpose of computing is insight, not numbers.’ – Richard Hamming**

# Intent

At Alice Ingham RC Primary School, our Computing curriculum aims to give our pupils the fundamental life-skills, knowledge and understanding that they will embrace and utilise for the rest of their life. We want our pupils to have a breadth of experience to develop their understanding of themselves as individuals within their community but also as members of a wider global community, as responsible digital citizens. Our high-quality computing education equips pupils to develop an understanding of how it has deep links with other subjects.

We believe it is important to engage the children with cross-curricular learning through interacting with a variety of technology. Therefore, we endeavor to provide computing opportunities to support learning across the entire curriculum and ensure that our curriculum is assessable to every child. We focus on a progression of skills in digital literacy, computer science, information technology and online safety (digital citizenship) to ensure that children become competent in safely using, as well as understanding, technology. Through the strands of computing, children are introduced to a

# Impact

The impact of our computing curriculum can not only be seen in displays around school and on the children's individual computer accounts but also can be measured by speaking to the children themselves. The teaching of the computing curriculum enables our children to use a computer with confidence.

We measure the impact of our curriculum using the following methods:

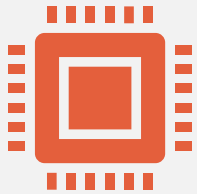
- Summative assessment of pupil discussions about their learning.
- Images of the children's practical learning in a class portfolio
- Children's work saved onto their individual accounts
- Interviewing the pupils about their learning (pupil voice).
- Class portfolios are scrutinised and there is the opportunity for a dialogue between teachers to understand their class's work.
- Annual reporting of standards across the curriculum.

Core principles

The Computing Curriculum has been written to support all pupils. Each lesson is sequenced so that it builds on the learning from the previous lesson, and where appropriate, activities are scaffolded so that all pupils can succeed and thrive. Scaffolded activities provide pupils with extra resources, such as visual prompts, to reach the same learning goals as the rest of the class. Exploratory tasks foster a deeper understanding of a concept, encouraging pupils to apply their learning in different contexts and make connections with other learning experiences. As well as scaffolded

# Pedagogy

Computing is a broad discipline, and computing teachers require a range of strategies to deliver effective lessons to their learners. Our pedagogical approach consists of 12 key principles underpinned by research: each principle has been shown to contribute to effective teaching and learning in computing.



These 12 principles are embodied by The Computing Curriculum, and you can find examples of their application throughout the units of work at every key stage.


Lead with concepts
Support learners in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, support this approach.
Work together
We encourage collaboration, specifically using pair programming and peer instruction and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding.
Get hands-on
Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides learners with a creative, engaging context to explore and apply computing concepts.
Unplug, unpack, repack
Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach, called ‘semantic waves,’ can help learners develop a secure understanding of complex concepts.

Model everything

Model processes or practices — everything from debugging code to binary number conversions— using techniques such as worked examples and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away.

Foster program comprehension

Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.

Create projects

Use project-based learning activities to provide learners with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Learners can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria. Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all learners engaged and encourage greater independence.

Challenge misconceptions

Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.

Make concrete

Bring abstract concepts to life with real-world, contextual examples, and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives.

Structure lessons

Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and UMC (Use-Modify-Create). These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.

Read and explore code first

When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils' ability to write code.

| Computing in EYFS | | | |
|---|---|---|---|
| **Three and Four-Year-Olds** | Personal, Social and Emotional Development | | • **Remember rules without needing an adult to remind them.** |
| | Physical Development | | • **Match their developing physical skills to tasks and activities in the setting.** |
| | Understanding the World | | • **Explore how things work.** |
| **Reception** | Personal, Social and Emotional Development | | • **Show resilience and perseverance in the face of a challenge.**<br>• **Know and talk about the different factors that support their overall health and wellbeing:**<br>**- sensible amounts of 'screen time'.** |
| | Physical Development | | • **Develop their small motor skills so that they can use a range of tools competently, safely and confidently.** |
| | Expressive Arts and Design | | • **Explore, use and refine a variety of artistic effects to express their ideas and feelings.** |
| **ELG** | Personal, Social and Emotional Development | Managing Self | • **Be confident to try new activities and show independence, resilience and perseverance in the face of challenge.**<br>• **Explain the reasons for rules, know right from wrong and try to behave accordingly.** |
| | **Expressive Arts and Design** | **Creating with Materials** | • **Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design** |

This document demonstrates which statements from the Development Matters are prerequisite skills for computing within the national curriculum. The table below outlines the most relevant statements taken from the Early Learning Goals in the EYFS statutory framework and the Development Matters age ranges for Three and Four-Year-Olds and Reception to match the programme of study for computing.

The most relevant statements for computing are taken from the following areas of learning:

- Personal, Social and Emotional

# Computing Curriculum at glance KS1 and KS2

| | Computing systems and networks | 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 England's key stage 1 national curriculum for computing, but the title is used as a strand across primary.

*The numbers in the brackets are a 'quick code' reference for each unit, e.g. 1.3 refers to the third Year 1 unit in the recommended teaching order.

# Unit Summaries KS1

| | Computing system and networks | Creating media | Prommaning 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 | ✓ | | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | |

| | 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) | Webpage creation (6.2) | Variables in games (6.3) | Introduction to spreadsheets (6.4) | 3D Modelling (6.5) | Sensing movements (6.6) |

# Unit Summaries KS2

| | 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 by modifying text, images, and page layouts for a specified 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 the internet as 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 | **Repetition in shapes** Using a text-based programming language to explore count-controlled | **Data logging** Recognising how and why data is collected over time, before using data loggers to carry | **Photo editing** Manipulating digital images, and reflecting on the impact of changes and whether the required | **Repetition in games** Using a block-based programming language to explore count-controlled and |

| | 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 evaluating 3D computer models of physical objects. | **Sensing movement** Designing and coding a project that captures inputs from a physical device. |

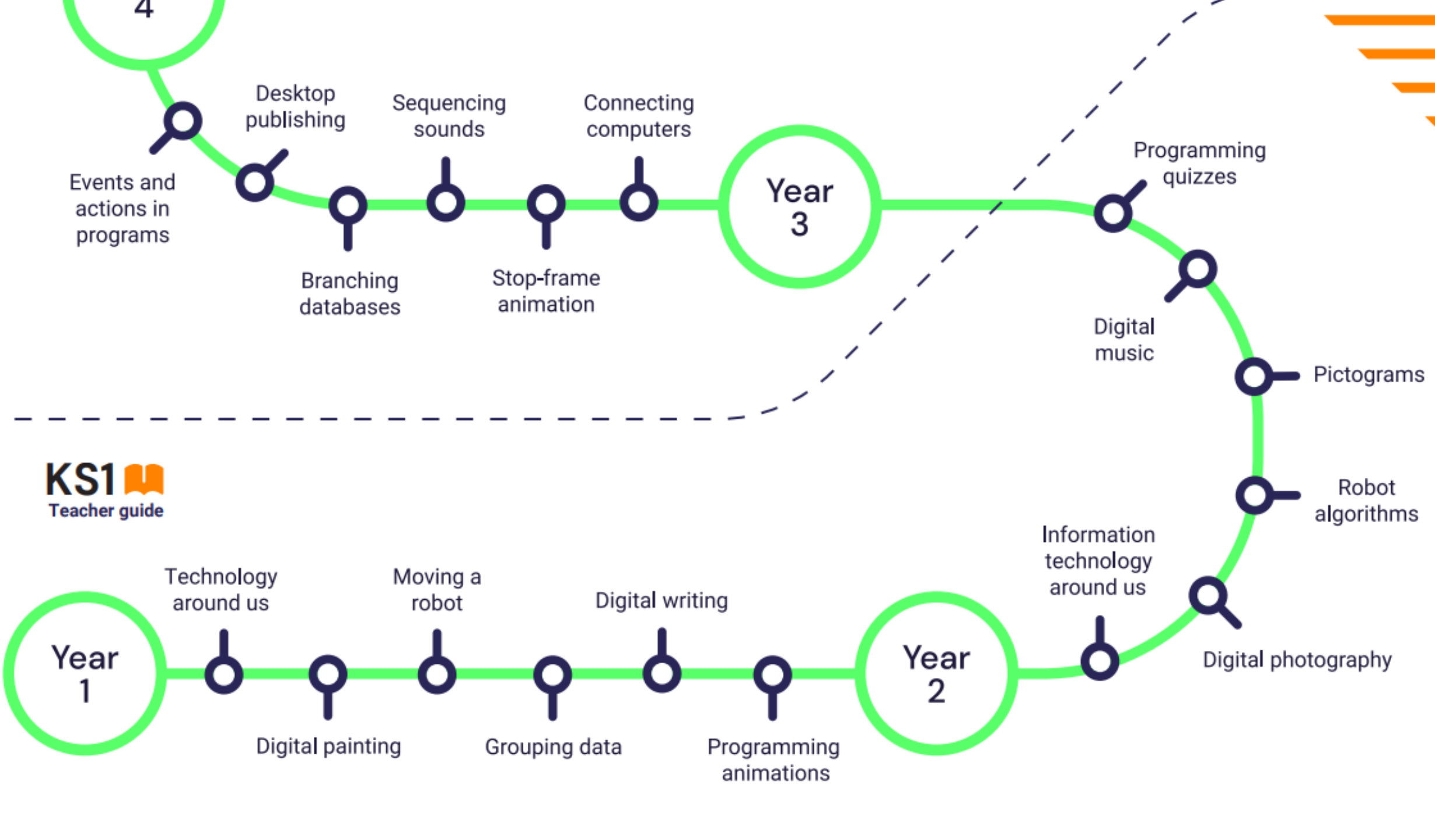
| National curriculum coverage - Years 3 and 4 | 3.1 Connecting computers | 3.2 Stop-frame animation | 3.3 Sequencing sounds | 3.4 Branching databases | 3.5 Desktop publishing | 3.6 Events and actions in programs | 4.1 The internet | 4.2 Audio production | 4.3 Repetition in shapes | 4.4 Data logging | 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 databases | 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 | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ | |

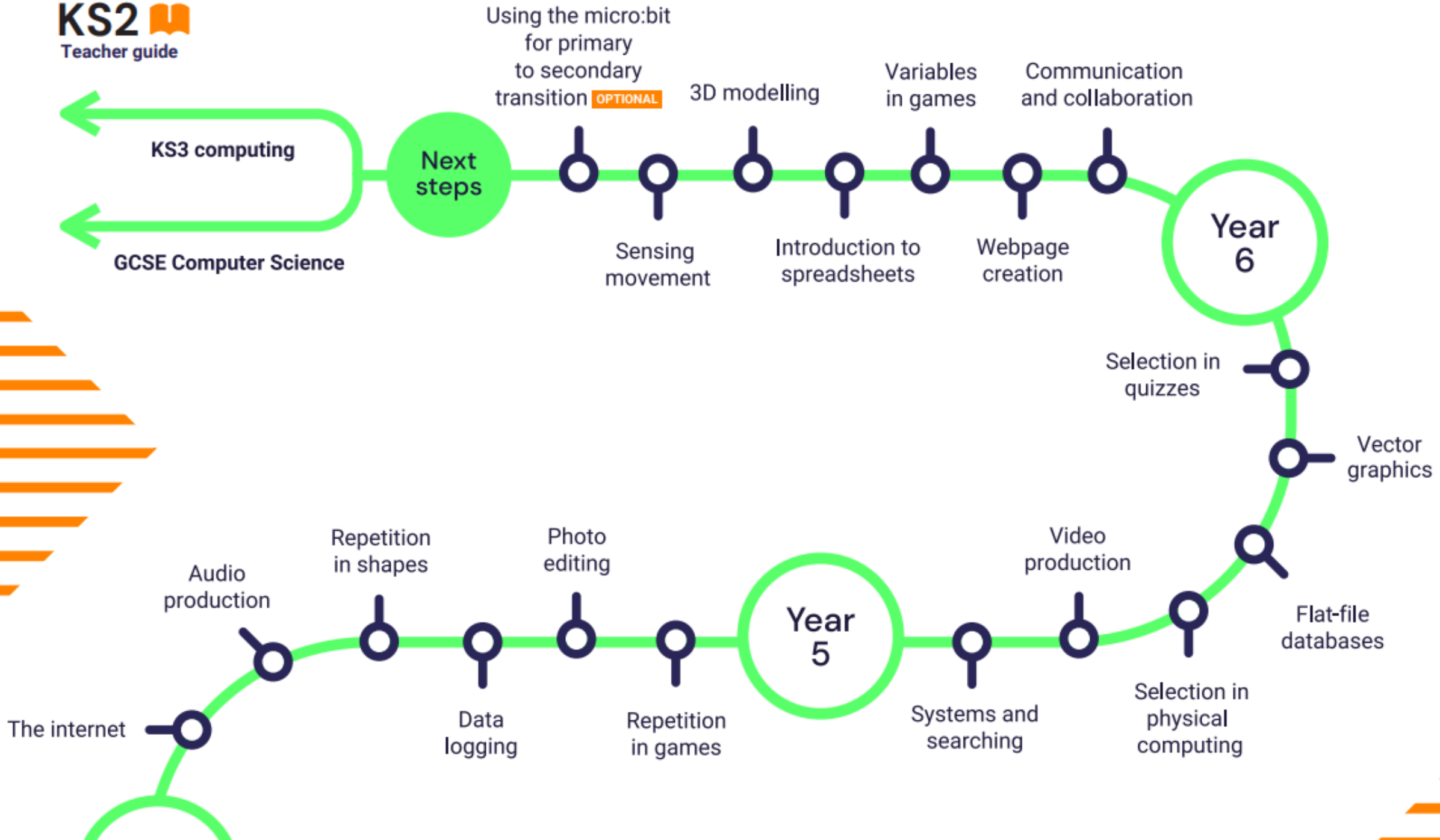
# KS1 & KS2

The units for key stages 1 and 2 are based on a spiral curriculum. Each of the themes are 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.


4
Events and actions in programs
Desktop publishing
Branching databases
Sequencing sounds
Stop-frame animation
Connecting computers
Year 3
Programming quizzes
Digital music
Pictograms
Robot algorithms
Digital photography
Information technology around us
Year 2
Programming animations
Grouping data
Digital writing
Moving a robot
Digital painting
Technology around us
Year 1
KS1
Teacher guide

# KS2
Teacher guide



# Progression

All learning objectives (for all key stages) have been mapped to the computing taxonomy of eleven strands, which ensures that units build on each other from one key stage to the next.

Every year group learns through units within the same four themes, which combine ten strands of the taxonomy.

This approach allows us to use the spiral curriculum approach to progress skills and concepts from one year group to the next.

| Primary Themes | Computing systems and networks | Programming | Data and information | Creating media |
|---|---|---|---|---|
| Taxonomy strands | Computing systems<br><br>Computing networks | Programming<br><br>Algorithms<br><br>Design and development | Data and information | Creating media<br><br>Design and development |
| Effective use of tools | | | | |
| Impact of technology | | | | |
| Safety and security | | | | |

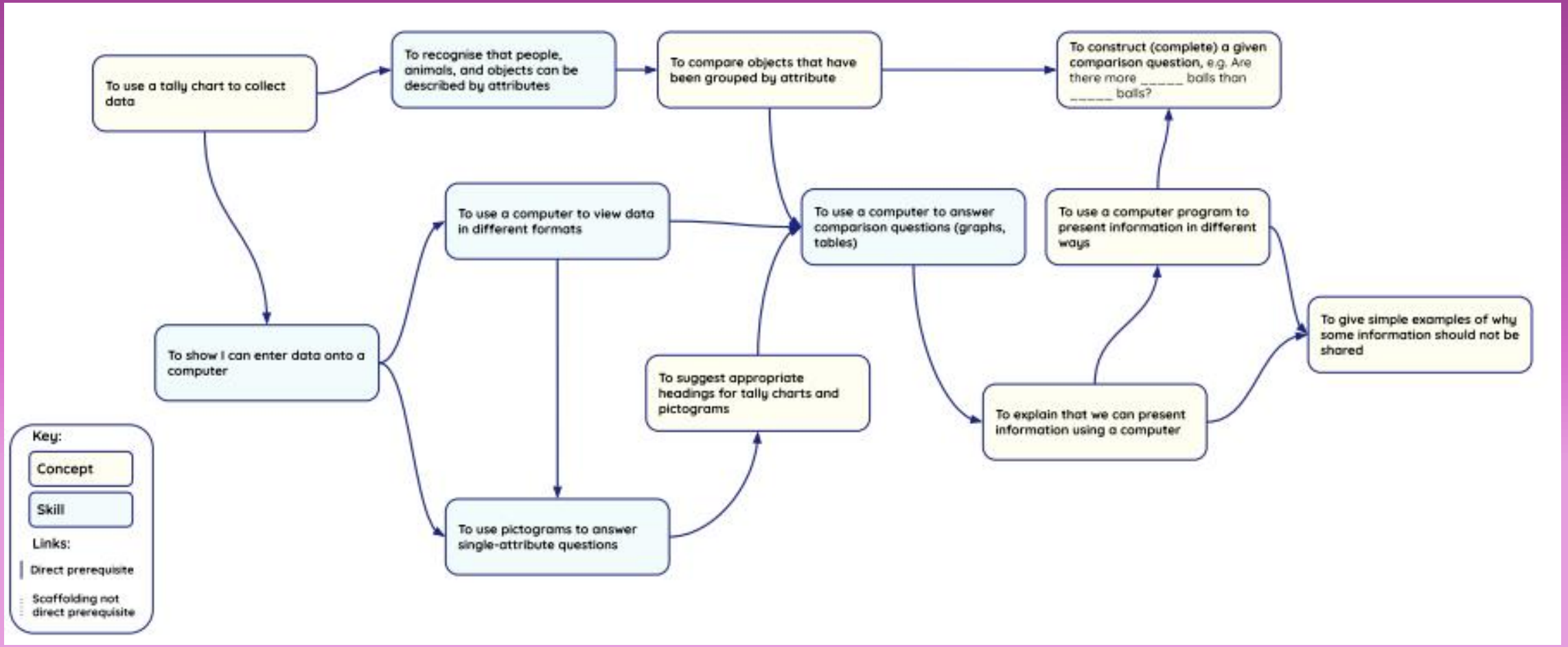
| Computing Systems and Networks | |
|---|---|
| 1 | Technology around us |
| 2 | IT around us |
| 3 | Connecting Computers |
| 4 | The Internet |
| 5 | Systems and searching |
| 6 | Communication and Collaboration |

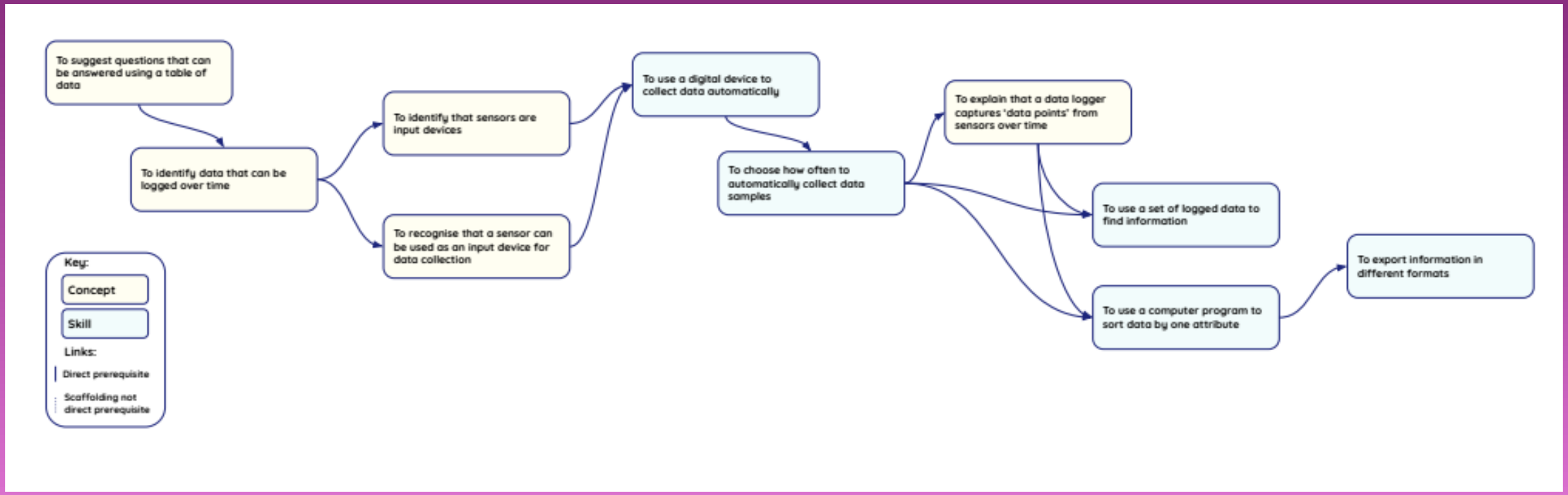
| Data and Information | |
|---|---|
| 1 | Grouping data |
| 2 | Pictograms |
| 3 | Branching databases |
| 4 | Data logging |
| 5 | Flat file databases |
| 6 | Introduction to spreadsheets |

| Programming | | |
|---|---|---|
| 1 | Moving a robot | Programming animations |
| 2 | Robot algorithms | Programming quizzes |
| 3 | Sequencing sounds | Events and actions in programs |
| 4 | Repetition in shapes | Repetition in games |
| 5 | Selection in physical computing | Selection in quizzes |
| 6 | Variables in games | Sensing movement |

| Creating media | | | | |
|---|---|---|---|---|
| | Text | Graphics | Photo and video | Audio |
| 1 | Digital Writing | Digital painting | | |
| 2 | | | Digital photography | Digital Music |
| 3 | Desktop publishing | Stop frame animation | | |
| 4 | | Photo edition | | Audio production |
| 5 | | Introduction to graphics | Video production | |
| 6 | Web page creation | 3D modelling | | |

# KS1 Learning graph






# KS2 Learning graph

*Year 4 example

# Assessment

## Formative assessment

Every lesson includes formative assessment opportunities for teachers to use. These opportunities are listed in the lesson plan and are included to ensure that you can recognise and address alternate conceptions if they occur. They vary from teacher observation or questioning, to marked activities.

These assessments are vital to ensure that you can adapt your teaching to suit the needs of the learners you are working with, and you are encouraged to change parts of the lesson, such as how much time you spend on a specific activity, in response to these assessments.

The learning objective and success criteria are introduced in the slides at the beginning of every lesson. At the end of every lesson, learners are invited to assess how well they feel they have met the learning objective using thumbs up, thumbs sideways, or thumbs down. This gives learners a reminder of the content that has been covered, as well as a chance to reflect. It is also a chance for you to see how confident your class is feeling so that you can make changes to

# Assessment

## Summative assessment

Pedagogically, when we assess, we want to ensure that we are assessing a learner's understanding of computing concepts and skills, as opposed to their reading and writing skills. Therefore, while learners are still developing their literacy skills, we encourage observational assessment. We believe that this is the most reliable way to capture an accurate picture of

# Resources

Requirements for pupils – below

✓ Used for the unit – reflected in screenshots ● Could be used as an alternative

| | Desktop or laptop | Chromebook | Tablet | Software or hardware |
|---|---|---|---|---|
| 1.1 Technology around us | ✓ | ✓ | ● | paintz.app |
| 1.2 Digital painting | ✓ | ✓ | ● | Microsoft Paint or similar |
| 1.3 Moving a robot | | | | Bee-Bot, Blue-Bot, or other fixed-movement floor robot |
| 1.4 Grouping data | ✓ | ✓ | | Google Slides or Microsoft PowerPoint |
| 1.5 Digital writing | ✓ | ✓ | ● | Google Docs or Microsoft Word |
| 1.6 Programming animations | ● | ● | ✓ | ScratchJr |
| 2.1 Information technology around us | ✓ | ✓ | | Google Slides or Microsoft PowerPoint |
| 2.2 Digital photography | ✓ | | ● | Digital camera |
| 2.3 Robot algorithms | | | | Bee-Bot, Blue-Bot, or other fixed-movement floor robot |
| 2.4 Pictograms | ✓ | ✓ | ● | j2data Pictogram |
| 2.5 Digital music | ✓ | ✓ | ● | Chrome Music Lab |
| 2.6 Programming quizzes | ● | ● | ✓ | ScratchJr |

# Resources

| | Desktop or laptop | Chromebook | Tablet | Software or hardware |
|---|---|---|---|---|
| 3.1 Connecting computers | ✓ | ● | ● | Painting program (any) |
| 3.2 Stop-frame animation | ● | ● | ✓ | iMotion (app for iOS) |
| 3.3 Sequencing sounds | ✓ | ✓ | ● | Scratch |
| 3.4 Branching databases | ✓ | ✓ | ● | j2data Branch and Pictogram |
| 3.5 Desktop publishing | ✓ | ● | ● | Canva.com |
| 3.6 Events and actions in programs | ✓ | ✓ | ● | Scratch |
| 4.1 The internet | ✓ | ✓ | ✓ | Various websites |
| 4.2 Audio production | ✓ | | | Audacity |
| 4.3 Repetition in shapes | ✓ | ● | ● | FMSLogo |
| 4.4 Data logging | ✓ | + | + | Data logger and associated software |
| 4.5 Photo editing | ✓ | ● | | Paint.NET (for Microsoft Windows) |
| 4.6 Repetition in games | ✓ | ✓ | ● | Scratch |

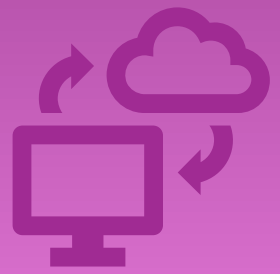
✓ Used for the unit – reflected in screenshots ● Could be used as an alternative + Data loggers that work with Chromebooks or tablets are available. Check with suppliers.

# Resources

| | Desktop or laptop | Chromebook | Tablet | Software or hardware |
|---|---|---|---|---|
| 5.1 Systems and Searching | ✓ | ✓ | | Google Slides |
| 5.2 Video production | ✓ | ● | ● | Microsoft Photos (for Microsoft Windows 10) |
| 5.3 Selection in physical computing | ✓ | ✓ | | Crumble controller + starter kit + motor |
| 5.4 Flat-file databases | ✓ | ✓ | ● | j2data Database |
| 5.5 Introduction to vector graphics | ✓ | ● | | Google Drawings |
| 5.6 Selection in quizzes | ✓ | ✓ | | Scratch |
| 6.1 Communication and collaboration | ✓ | ✓ | | Google Slides |
| 6.2 Webpage creation | ✓ | ✓ | | Google Sites |
| 6.3 Variables in games | ✓ | ✓ | | Scratch |
| 6.4 Introduction to spreadsheets | ✓ | ✓ | ● | Google Sheets or Microsoft Excel |
| 6.5 3D modelling | ✓ | ✓ | ● | Tinkercad |
| 6.6 Sensing movement | ✓ | ✓ | ● | micro:bit and Microsoft MakeCode |

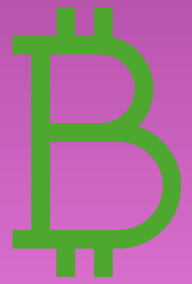
✓ Used for the unit – reflected in screenshots ● Could be used as an alternative

# Physical computing



The physical computing units in The Computing Curriculum are:

Year 5 – Selection in physical computing, which uses a Crumble controller



Year 6 – Sensing movement, which uses a micro:bit

# Online safety

For each unit, the unit overview document shows the links between the content of the lessons and England's national curriculum and the Education for a Connected World framework. These references have been provided to show where aspects relating to online safety, or digital citizenship. The coverage required for England's computing national curriculum is provided.