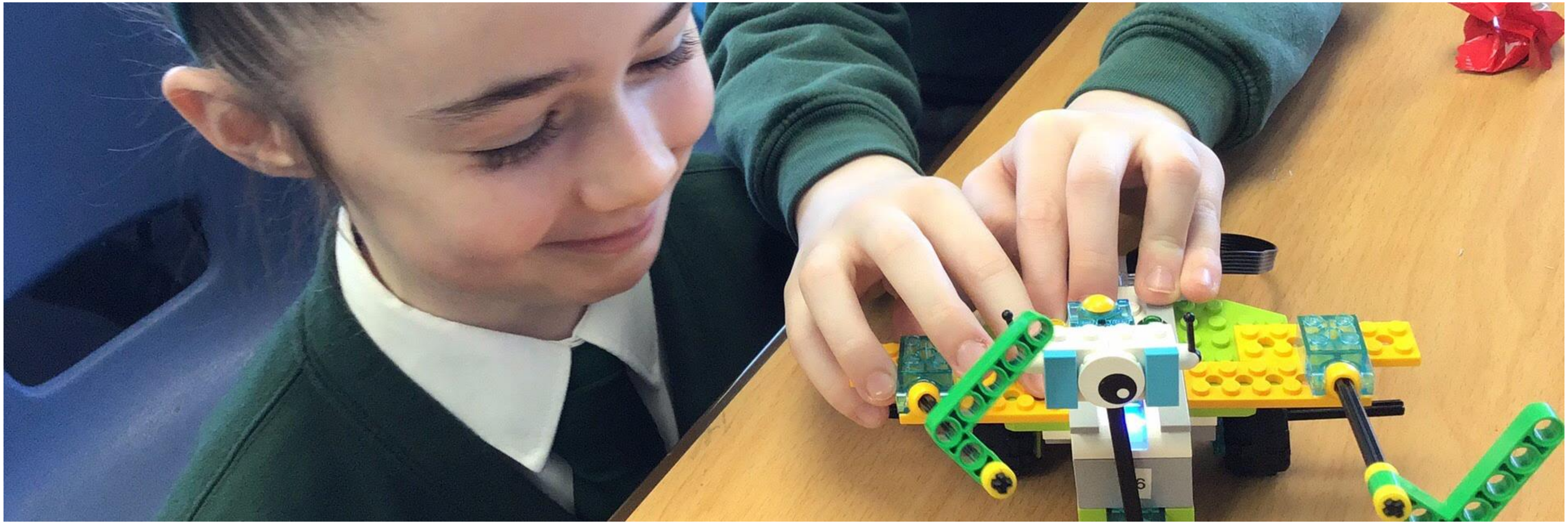
Children's Wellbeing – Learning and Achievement Service – Herefordshire Computing Support

The Herefordshire Primary Computing Progression 2020

Lower Key Stage 2 (Years 3 & 4) - a revised version of the 2014 materials



This document can be found, with supporting resources at
herefordshirecomputing.com

Table of Contents

Introduction

Computer Science (CS): Programming

Computer Science (CS): Computers & networks

Information Technology (IT)

Digital Literacy (Online safety)

Assessment

Introduction

1 From 2014 to 2020

Six years on from the introduction of the Computing Curriculum and the original version of these materials things have moved on somewhat and so here is a revision of the Herefordshire Progression materials. So what's changed?

Computer Science

No changes here to the learning statements, except to divide them into year groups rather than pairs which is always requested by single year group schools – however please don't take that too literally. What has changed though is the bank of recommended lesson material. These are drawn entirely from three key sources: Barefoot Computing, Code Club and Code-IT (Phil Bagge). Recently it has become possible to use iPads to run Scratch albeit missing certain functionality (especially keyboard presses). The suggested activities lists for each year group indicate what will work on iPads.

The *Computers and Networks* lists of resources (KS2) have been updated.

Information Technology

The headings have been simplified (reduced) here. Learning statements have been updated and reduced, as have the cross curricular examples. The content has been altered to reflect the far greater proportion of iPads now in schools and relatively few computers.

In response to many requests over the years I've finally given way and split the learning statements in each booklet between the two year groups. The most important driver of the use of technology remains the context in which it is used, however, and this will often dictate the skills that are developed. The progression is only a guide and should be secondary to use of technology to support children's learning in this strand.

Digital Literacy (Online Safety)

This section has been entirely replaced. There remains no shortage of lesson material for online safety and many of the old favourites, which teachers know their way around remain in the table of possible of resources (with a few newer ones). Primarily though, the structure of this strand, and the suggested resources comes from the recent document Education for a Connected World and the even more recent resource bank that supports it, Project Evolve. There are eight strands at every level and often several statements within each – so quite a bit more content here than before.

It should be said that these resources reflect something of a more informed and intelligent approach to online safety, recognising that some of the messages we have given to children in the past have never really had a reasonable chance of having any impact. This should be born in mind when evaluating which older materials should be retained and which replaced.

Assessment

The more complex assessment grids have been removed and only the "simple" one remains. Some learning statements have been reorganised to create six rather than 5 "levels"

Supporting resources

These have been adapted to reflect the changes above. The software maps have been removed and the recommended iPad App list updated - this is always an evolving document anyway.

2 Organisation of the programme of study

Computing breaks down into three main areas as follows (extracts from the programme of study that follow are from the *purpose of study* and *Aims* sections and are for KS1 – KS4):

Computer science (CS)

The programme of study states:

> 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.
>
> 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

The main component, then of CS is programming where children engage in practical, creative experiences. There is also a requirement for children (especially at KS2) to understand physical computer systems. We've called this **Computers and Networks** in these materials

> - Pupils should be taught to 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

Information technology (IT)

> … Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content.
>
> The national curriculum for computing aims to ensure that all pupils:
>
> - can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems

This strand involves the productive use of readymade computer applications to create content in a variety of forms (text, image, sound, video, animation, data gathering and processing, AR, VR …) often supporting the whole curriculum. It is important that we don't lose sight of this aspect of computing.

Digital literacy (DL)

> … 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.
>
> The national curriculum for computing aims to ensure that all pupils:
>
> - are responsible, competent, confident and creative users of information and communication technology.

This is online safety, "responsible" use of tech. This is far more explicit in each of the specific key stage statements of the programme of study than it is in these general ones.

3 Computational thinking

So where does computational thinking fit into all this then? It really pervades the whole subject; that said it sits most prominently in **Computer Science.**

It is a kind of literacy; a way of organising thinking that is not only relevant to computing but very much to life and learning and therefore goes well beyond the subject of computing and indeed beyond education itself – it is a skill for life. When children are "learning to code" they are also "coding to learn."

Computational thinking is a slightly problematic term; there are some very different interpretations of it. Google has a helpful page, where they identify core components of this as algorithms, decomposition, abstraction, and generalised patterns. The influence of these ideas on the computing curriculum is clear, with algorithms at KS1 and decomposing problems at KS2. Whilst these are certainly part of the computer scientist's toolkit, there's perhaps more going on than when they come to tackle new problems or design systems.

Resnick and Brennan, of Scratch fame, have an interesting paper in which they revisit break this down into concepts, practices and perspectives – it is worth reading. Whilst these concepts, practices and perspectives can (and should) all be learnt through practical experience of programming, teaching needs to go beyond the 'this is how you use Scratch' or 'this is how you use Kodu' if we are to do justice to the ambitions expressed here.

It's important that we all have an understanding of the principles and concepts behind computational thinking and keep them in mind when designing and delivering the curriculum.

4 How to use these materials

The progression is organised into three pairs of year groups:

- KS1 (Years 1 and 2)
- Lower KS2 (Years 3 and 4)
- Upper KS2 (Years 5 and 6)

Broad brush strokes are important and use of computing, especially in the IT strand should always be dictated by its relevance to learning generally. I've resisted calls to break down statements to individual year groups but Single year group classes often prefer to have "ownership" of learning statements within each class so "4" after a statement indicates that could be Y4 goal.

The Progression is based around the three headings from the programmes of Study (explained above) and each of these is broken down further but it different ways and with different resources depending on the strand.
This Computing Progression (especially the information technology [IT] strand) should be integrated into planning across all subjects. A **long term planning grid** is available as a simple mapping tool to help start this process and to ensure that the full range of computing entitlement is covered.

Each school should also develop its own resources map. A sample, containing recommendations for Herefordshire primary schools is available to use as a template and can be downloaded (along with all the supporting resources for this Progression) at www.herefordshirecomputing.com

5 Help and support in your classroom

Support in Herefordshire comes in the form of *Herefordshire Computing Support,* a part of Herefordshire Council's *Learning and Achievement Service.* Many primary schools in Herefordshire subscribe to our service level agreement which means we're at your disposal to come and help you with planning and with hands on support in your classroom. If your school doesn't buy into our service level agreement, or if you're in a neighbouring local authority we can still help but will need to charge on a "pay-as-you-go" basis for the support we provide. Please contact msanderson@herefordshire.gov.uk or talk to your school's Computing Coordinator to sort out what's possible.

Additionally a pool of resources is available for loan to schools to support areas of computing that are expensive to support. Courses are provided and the Digital Leaders Network is a popular initiative in schools.

Computer Science (CS): Programming

KS2 Programme of study extract

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

What does this mean?

You can be forgiven for asking that question! Programming comes with a language of its own and getting one's head around the terminology, then using it correctly with children is half the battle. So here's a blow-by-blow explanation of the terms used in the program of study:

- **Algorithm** – a set of steps that accomplish a task. So a program is an algorithm, but so is the story board the children created before they started programming. See Barefoot: Teach yourself
- **Design** – an important step before any programming takes place, children can design away from the computer, they need to decide what they are going to create, break it down into steps, story board etc.
- **Write** – the actual process of programming at the computer (based on the algorithm written in the design stage above).
- **Debug** – correcting errors in a program. This tends to happen naturally but needs to be discussed and children need to develop strategies.
- **Controlling or simulating physical systems** – writing program that control physical devices such as Lego WeDo or Data Harvest's FlowGo. The D&T PoS requires KS2 pupils to "apply their understanding of computing to program, monitor and control their products".
- **Sequence** – writing a program in the correct order.
- **Selection** – using "if … then …" type commands e.g. if the fish is touching the net then game over.
- **Repetition** – often called a loop. Commands that need to be repeated. So in Scratch a sprite can walk around the sides of a square by repeating 4 times: Forward 100, Turn right 90.
- **Variables** – scoring is a good example of this. At the start of a game a variable called "score" is set to 0 then 1 is added to the score every time the cat touches the mouse.
- **Input** – any means by which signals are given to a program to do things. So it could be pressing the up arrow on the keyboard, clicking the mouse, moving a tilt sensor, an activated light sensor…
- **Output** – the action that happens as a result of an input: the sprite moves, the game starts, the alarm goes off, the motor starts…

Here is a more comprehensive and detailed explanation of the above terminology (with a few others)

Somerset (Elim) also has some help: KS2 Computational language: Scratch or this one from Phil Bagge.

How shall we teach it?

This is likely to be the most challenging aspect of the computing curriculum for you to deliver (not necessarily so for the children, which somewhat adds to the problem!)

Ideally, programming should be embedded into the curriculum and once we all understand what's really involved this will be possible. For the moment it's going to be a lot easier to teach it as a discrete element on a timetabled basis, perhaps as one or two half term units per year.

Scratch is the best way to get going at KS2 and there is already a raft of excellent resources available of the internet to help you to do this. Many of these, from various sources, are signposted in the progression that follows.

It is very difficult to define a particular order in which skills should be introduced. Some commercially available schemes / programming environments (see below) do this very successfully but often by restricting the programming tools available for a particular activity and therefore stifling creativity. The open nature of programming environments such as Scratch, the skills that are required for any particular project and the spirit of creativity that runs through the programme of study means that you will almost certainly jump around in the skills list. You may very well plan that you are only going to cover some very basic skills in a particular session but some children will quickly want to build on them and make much more advanced things happen.

We have, however, attempted a progression of skills, and broken them down into upper and lower KS2, but this should not be taken too literally. It could, be used as a starting point for planning though. The suggested activities that follow have been chosen, as far as possible, to reflect this breakdown. But remember, children may want to take things in a different direction and at some point you should give them this freedom. The point at which you do this is up to your professional judgement and is part of your learning curve as a teacher.

Scratch is of course not the only programming environment, there are many others including quite a few which require an annual subscription. If your school is using one of those then you may prefer to follow the scheme of work that comes with your solution for the time being. That will give you a great opportunity to build up your skills for a year or so. Scratch (and Kodu) is great fun and it will be worth looking at them later on even if you don't immediately. Giving your children access to an open environment of this nature is really much more within the spirit of the computing curriculum than following a tightly defined sequence of lessons that will, to a degree, stifle creativity.

Whatever you choose to use, you will need to think carefully about the skills your children have, especially in the first year or two of teaching computing. If little in the way of programming has been taught in your school in the past then you will need to be working on activities that develop some more basic skills that will ultimately be desirable. Be prepared to go back and program floor robots, or even to have children programming other children acting as robots if you need to before attempting some of the more advanced activities that are suggested here for KS2.

How does it link to other aspects of computing?

- The programme of study extracts above are those primarily concerned with programming. The statement, which we have categorised as **information technology** mentions "design and create a range of programs, systems and content that accomplish given goals…" The implication of this is that an activity they have programmed may form a part of a wider "system" or "content" creation.
- Specifically, a certain amount of programming goes on when **using spreadsheets**, especially when using formulae. Certainly a fair bit of debugging!
- Programming obviously links to the other aspect of **computer science** concerning **computers and networks** as programs are fundamental to physical computer science
- If children share their projects on the Scratch community there are links to **digital literacy (e-safety)**

- It is worth noting that there are also strong links to other subjects, especially English and mathematics and these will become apparent as you travel further down the programming road.

What kit, software and resources are there to help me (highlights indicate specially recommended options)

Scratch - www.scratch.mit.edu Widely accepted as the mainstream coding environment, certainly for KS2. Scratch set the benchmark for all similar environments. With the advent of Scratch 3 came the ability to use it on tablets. There are stil limits though as any project requiring keyboard presses or mouse following behaviour are still not possible. If you plan to use Scratch on iPads you will also need to create a teacher's account and then make online accounts for your shildren so that they can save their projects onlineScratch is totally free. And still the best coding platform for KS2.

Scratch Junior - www.scratchjr.org Available as an iOS and Android app, and it works in a webrowser on a computer. It's really aimed at KS1. It's a lovely programming environment which includes a subset of commands from Scratch that are just right for KS1 and even lower KS2. The web link given here is to a website for education professionals where you'll find some introductory learning and teaching resources. Scratch Jr is a free app. Phil Bagge has some good planning at www.code-it.co.uk

Kodu - www.kodugamelab.com This is Microsoft's offering. It's very 3D based and games created on it can be played on the Xbox. The process of creating a game involves first creating the "world," a process very like the hugely popular (with children) Minecraft game environment. That done, characters are programed. Kodu is harder to get to grips with, especially if you're not a "gamer," than Scratch and not so adaptable. It also has quite high level specification requirement of the machine it runs on and some older computers in school will struggle. Kodu is free and is a Windows only application.

Coding apps for tablets – There's a huge array of apps available for coding for all tablets, some of the best are listed on the recommended apps list (www.herefordshirecomputing.com) It's true to say that none of them is quite as good as Scratch, and that certainly is true when you take into account what is available in terms of free planning to help you use them productively in the classroom. If you still have PCs available then this is the best option for most coding. Tablets are particularly good for controlling external devices now though.

Tynker – This is a popular iPad app for coding. It's particularly good for connecting to other devices (I've used it a lot with drones). There is a comprehensive collection of classroom resources but they have to be paid for. It is nice to use though and children love the characters. Coding blocks are arranged rather like those in Scratch. The app is free. https://www.tynker.com/

Swift Playgrounds – A text based coding application for iPad from Apple. It comes with some tutorials that children can work their way through, they take off quite quickly though and get quite challenging. This is really a resource for upper KS2 and beyond. That said, children enjoy it. https://www.apple.com/uk/swift/playgrounds/

Scheme of Work: Discovery Coding - www.discoveryeducation.co.uk (formerly Espresso) who created their own coding environment and supported it with many readymade, self-contained lessons (two six lesson units per year group, though most of them will be accomplished in fewer than six lessons). This is a great way of building a teacher's skills perhaps over the course of a year but the approach isn't really within the creative spirit of the computing programme of study. It certainly is not open ended enough for upper KS2. Annual subscription is

Scheme of Work: 2Code (Purple Mash) - www.purplemash.com A rather similar approach to Espresso Coding and the programming environment has a similar feel. It is much easier to access the full coding environment than in Espresso Coding and differentiation is built into each of the activities. A subscription to Purple Mash (of which 2Code forms a part) is required.

Scheme of Work: code-it by Phil Bagge – www.code-it.co.uk - This is an EXCELLENT site packed with comprehensive resources all ready for you to use in the classroom. It's particularly rich in lessons supporting Scratch and Scratch Jr and some material is appropriate for or can be can be adapted to work with iPad Scratch. You'll also find a great deal of support on this site for other areas of computing and *computers and networks* is well supported too. Phil Bagge is a Computing Adviser with Hampshire LA and regularly works in classrooms. He is a regional coordinator and master teacher for CAS (see below). It's possible to buy his material in printed form on his site. It's not the easiest site to navigate, however so persevere. There is also a lot of theory there and the many different "generations" of pedagogical thinking can get a bit confusing. It's a free resource.

Scheme of Work: Barefoot Computing – www.barefootcomputing.org – A national organisation, supported by DfE funding and many others to help primary school teachers teach computer science. Some very well structured support here and lots of lesson resources. Most of their Scratch activities don't involve keyboard presses or mouse following behaviour so will work well on tablets. The material lis beautifully organised on a lovely site with a good search tool. Lessons ar really well described and much supporting material is given. It's all free.

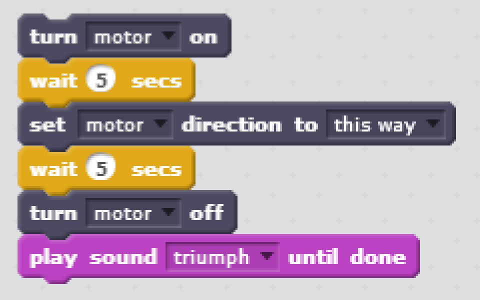
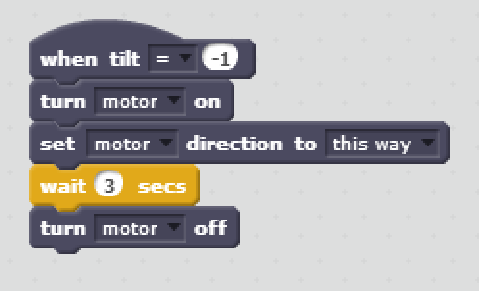
Code Club – www.codeclub.org.uk - Some excellent resources here, beautifully presented. Code Club supports volunteers running computing clubs in schools and is a part of the Raspberry Pi Foundation. The resources on this site are primarily designed for this and if you have a Code Club running in your school you should check with the person running it before potentially hijacking their material. All the activities on this site are at quite a high level, the targeted age range is 9-13. Because this is content for a "club" style environment it is possible for children to work their own way through the materials so great for homework or home learning. Again, it's free.

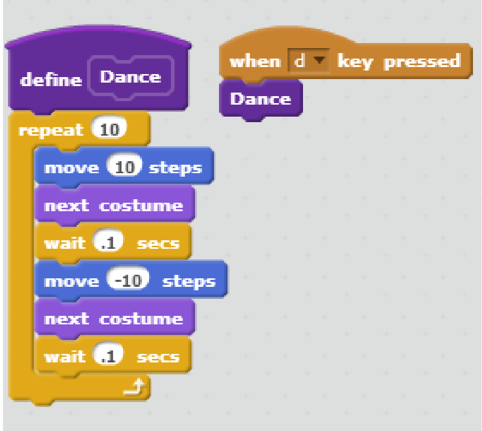
Computing at School (CAS) – www.computingatschool.org.uk - CAS is a community of teachers, academics, industry professionals, school governors, parents etc. with a mutual interest in computing. The CAS community is much more active in phases beyond primary education (and not very active locally at all) and most of the primary resources are now to be found on the Barefoot site (above). CAS has a wealth of material, especially on pedagogy and theory. It's just not very easy to find what you need. Not really the place to go for lesson material.

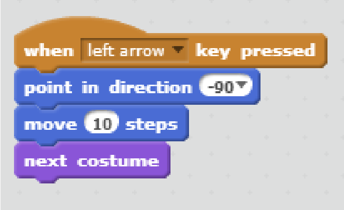
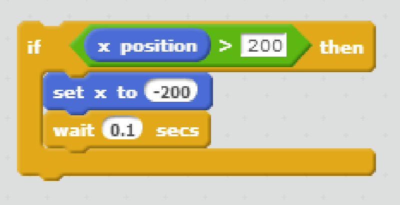
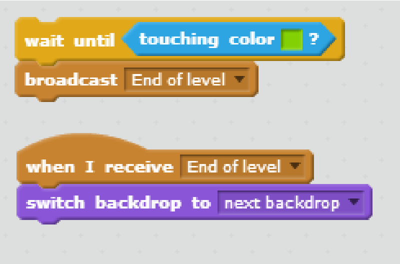
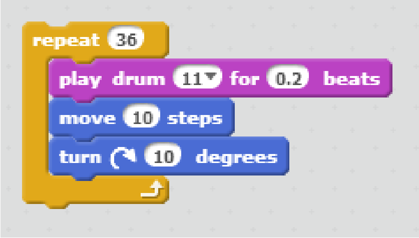
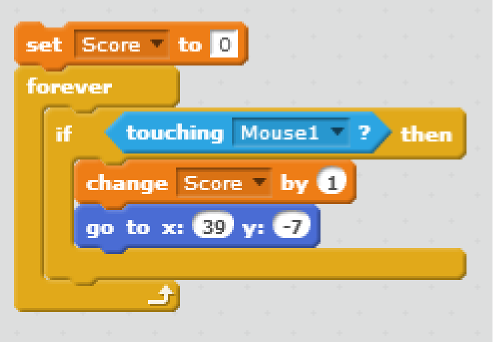
LEGO Education WeDo and WeDo 2.0: www.education.lego.com A really cool way into the world of control. The original set can be controlled with Scratch on a PC; WeDo 2.0 is perhaps best used and controlled with the free iOS app. As well as a comprehensive selection of bricks each kit comes with a motor, a tilt sensor and a distance sensor. Strong links to the whole of STEM, especially science. It's very expensive but we have whole class sets you can borrow. Children love it and it's an excellent product. The great thing is that it always works!

Micro:bit – www.microbit.co.uk Proclaimed as the natural successor to the BBC microcomputer from the 1980s. The DfE gave one to every Y7 student in 2016. A small programmable computer. The good news is that it's very relevant for upper KS2 and can be programmed via a Scratch-like interface from iPads. A very affordable (a bit over £10 each) and creative way into coding external devices with all manner of things possible. Children love it and I can thoroughly recommend it, I love leading sessions with children with my class set of these.

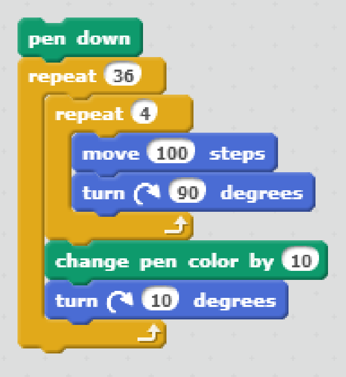
Progression in Scratch (mapped to the programme of study)

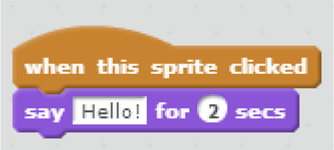
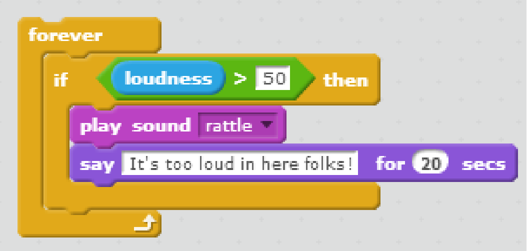
PoS extract	Lower KS2 Scratch objective	Upper KS2 Scratch objective
Design, write and debug programs that accomplish specific goals,	• Use a variety of blocks to create a program. • Write and describe the algorithm needed for a simple task. • Understand the need to keep testing a program while creating it.	• Explain and program each of the steps in my algorithm (for a device or onscreen activity). • Review and amend the original algorithm while programming.
including controlling or simulating physical systems;	• Use *more blocks* and LEGO WeDo to build programs that will control physical models they have made (also a D&T requirement) 4  turn motor on wait 5 secs set motor direction to this way wait 5 secs turn motor off play sound triumph until done • (Use Micro:bit, crumble boards, Raspberry Pi or similar with their associated software to code small portable computers to carry out simple tasks) 4	• Talk about how a computer model can provide information about a physical system. 6 • Use *more blocks* and LEGO WeDo to build programs that will control models they have made, including inputs. (Also a D&T requirement)  when tilt = -1 turn motor on set motor direction to this way wait 3 secs turn motor off • (Use Micro:bit, crumble boards, Raspberry Pi or similar with their associated software to create programs that will control models or on screen simulations, including making use of sensors monitoring physical conditions and with more complex coding, eg variables) 6

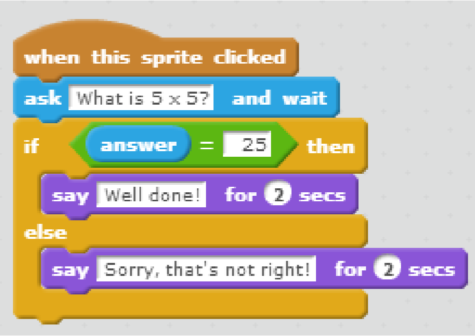
PoS extract	Lower KS2 Scratch objective	Upper KS2 Scratch objective
Solve problems by decomposing them into smaller parts	- Break an open ended problem up into smaller parts. - Use logical thinking to solve an open-ended problem by breaking it up into smaller parts. 4	- Decompose a problem into smaller parts to design an algorithm for a specific outcome and use this to write a program for a device or onscreen activity. - Deconstruct a problem into smaller steps, recognising similarities to solutions used before. 6 - When appropriate, investigate *more blocks – make a block* to create more commands for actions that repeat in a program (e.g. if a similar dance is required at different points in a program, create a block called "dance" to be called up when needed. 6  define Dance repeat 10 move 10 steps next costume wait .1 secs move -10 steps next costume wait .1 secs when d key pressed Dance
Use **sequence** in programs (LKS2) Use **selection** in programs (UKS2)	**Sequence (LKS2)** - Assemble programming commands into a sequence to achieve a specific outcome. - Use simple *motion, looks, sound* and *pen* blocks to program a sprite. Consider where order is important	**Selection (UKS2)** - Use *if … then* and *if … then … else* commands to select an action (below left)

PoS extract	Lower KS2 Scratch objective	Upper KS2 Scratch objective
	(e.g. making a turn before moving) 4  when left arrow key pressed point in direction -90 move 10 steps next costume	 if x position > 200 then set x to -200 wait 0.1 secs  wait until touching color ? broadcast End of level when I receive End of level switch backdrop to next backdrop - Use *broadcast …* and *when I receive …* commands (Events) to cause one sprite to affect the behaviour of another, or of a background. (E.g. when the sprite touches the "finish" box change background to the next level of the game – above right). 6
Use **repetition** in programs;	- Use *repeat* and *forever* commands.  repeat 36 play drum 11 for 0.2 beats move 10 steps turn ↻ 10 degrees	- Use *repeat until …* commands - Make a program more efficient by using repeat commands.
Work with **variables**	- Begin to understand the function of variables by using *change pen colour / shade / size by …* (there are similar commands in the *sound* blocks). 4	- Use *Data – Make a variable…,* to keep a score, for example. Program a sprite to add to the score on a certain action.  set Score to 0 forever if touching Mouse1 ? then change Score by 1 go to x: 39 y: -7

PoS extract	Lower KS2 Scratch objective	Upper KS2 Scratch objective

	
	pen down
	repeat 36
	  repeat 4
	    move 100 steps
	    turn ↻ 90 degrees
	  change pen color by 10
	  turn ↻ 10 degrees
		• Recognise when a variable is needed to achieve a required result. 6

Work with various forms of **input and output**	• Use *events* blocks and different inputs in programming (e.g. *when green flag is clicked, when I press the … key, when the sprite is clicked*)  when this sprite clicked say Hello! for 2 secs • Use a sensor to detect a change which can select an action within a program. 4	• Use the *sensing* commands to change behaviour / trigger events  forever if loudness > 50 then play sound rattle say It's too loud in here folks! for 20 secs • Use different inputs (including sensors) to control a device or onscreen action and predict what will happen. 6 • Use *ask … and wait* and *answer* commands use text imputed by users as inputs.

PoS extract	Lower KS2 Scratch objective	Upper KS2 Scratch objective
		 when this sprite clicked ask What is 5 x 5? and wait if answer = 25 then say Well done! for 2 secs else say Sorry, that's not right! for 2 secs
Use logical reasoning to explain how some simple algorithms work …	• Recognise that an algorithm will help when sequencing more complex programs.	• Use logical thinking, imagination and creativity to extend a program. • Evaluate the effectiveness and efficiency of an algorithm, continually testing the programming of that algorithm. 6
… and to detect and correct errors in algorithms and programs	• Keep testing programs and see the need to debug. Detect a problem in an algorithm which could result in unsuccessful programming.	• Link errors in a program to a problem in the algorithm on which it is based.

Suggested Scratch lessons covering upper KS2 objectives – Y3

The activities / suggested below are from three excellent providers of Scratch lesson resources and will give you some assistance with where to start.
The allocation to year group is only a rough guide but will help teachers in larger schools not to tread oneach others' toes!
Resources in **bold** with an ***i*** after them do not use keyboards or mouse following behaviour and should therefore work well in Scratch 3 on an iPad.
Links were correct in April 2020.

{code club}

Barefoot
Building skills for tomorrow

Animated poem decomposition ***i***

Fossil formation animation ***i***

Musical sequence activity (SEND) ***i***

Scratch tinkering activity ***i***

code-it.co.uk

Using loops to investigate 2D regular shapes 1 week
i *– but substitue a "tap sprite" for space bar*

Scratch Smoking Car 2-3 weeks

Scratch Music Machine 2-3 weeks ***i***

Scratch Conversation 1-2 weeks ***i***

Scratch Dressing Up Game 2-3 weeks ***i***

Suggested Scratch lessons covering upper KS2 objectives – Y4

The activities / suggested below are from three excellent providers of Scratch lesson resources and will give you some assistance with where to start.
The allocation to year group is only a rough guide but will help teachers in larger schools not to tread oneach others' toes!
Resources in **bold** with an ***i*** after them do not use keyboards or mouse following behaviour and should therefore work well in Scratch 3 on an iPad.
Links were correct in April 2020.

{code club}

Rock band ***i*** *– ignore space command*

Lost in space

Ghost busters

Barefoot
Building skills for tomorrow

Bug in the water cycle ***i***

Maths quiz selection ***i***

Shape and flowers repetition ***i***

Viking raid animation ***i***

code-it.co.uk

Scratch Music Score ***i***

Quiz ***i***

Scratch Slug Trail Game

Scratch Selection Investigation ***i***

Computer Science (CS): Computers & networks

KS2 Programme of study extract

Pupils should be taught to:

- **understand computer networks including the internet; how they can provide multiple services, such as the world wide web;**
- **appreciate how [search] results are selected and ranked**

What does this mean?

This Progression (like many others of its kind) attempts to break down the subject of computing into understandable and manageable chunks. But this programme of study extract perfectly demonstrates the joined up nature of computing.

Children will naturally be using “services such as the world wide web” and exploiting many of the “opportunities they offer for communication and collaboration” in the course of their computing work, particularly in the information technology strand but with obvious overlap with digital literacy.

That leaves us with the requirement for pupils to “understand computer networks including the internet; how they can provide multiple services”. This is about having a much better understanding (adults as well as pupils) about how computers are joined together in school, what the server does, what technologies (wired and wireless) connect them together and how all of this is connected to the wider world via the internet, etc.

How shall we teach it?

It follows that aspects of this strand (understanding the web, opportunities for communication and collaboration…) should be considered, referred to and delivered as part of work in those areas (IT), i.e. embedded. This is also true, to a degree, with gaining an understanding of networks and computers etc. For example, it is important to have conversations whenever necessary to ensure that children understand exactly where they are saving their work, why they are saving it there, where it is actually stored, who can access it and at what level … (This might even result in them being able to find it again next time!)

Given all of that, it would also seem appropriate to deliver some discrete sessions aimed at gaining this understanding and the suggestions that follow will provide resources for that. It would seem perfectly reasonable to cover this in a shout unit of half a term or less.

How does it link to other aspects of computing?

- There are obvious links to **information technology**, especially the data handling aspect but also when saving and sharing work locally and online in all aspects of computing.
- Links therefore to **digital literacy (e-safety)** when using or creating internet resources or communicating online.
- Clearly there are links to the programming strand of **computer science** through the obvious links between understanding programming and the systems on which programs run.

What kit, software and resources are there to help me?

CS Unplugged: http://csunplugged.org/ A great site from New Zeeland, packed with practical exercises you can do with your children in the classroom, school hall or on the playground to get them thinking, often through physical activity, how computers, applications and networks work. We've included a lot of specific links in this document but you might like to take a look at some of the other activities (many which go beyond KS2) that are on offer.

Junior Computer Science – Phil Bagge – www.code-it.co.uk - This is an excellent site packed with comprehensive resources all ready for you to use in the classroom. Many of the lessons that follow are from this site. Phil Bagge is a Computing Adviser with Hampshire LA and until recently has been teaching has been teaching across five primary schools. He is a regional coordinator and master teacher for CAS (see below)

Teach Computing is an excellent site with resources for the whole of the Computing curriculum, It's particularly rich on this strand with many really well constructed lessons by Andy Bush. You will need to register for a free account on the site before you can access the materials but it's well worth it. Resources from year groups other than your own are well worth looking at as the allocation of topics is not quite the same as here. You could just go with the structure suggested on this site. https://teachcomputing.org

Computational Fairy Tales – a wonderful collection of over 70 stories, in fairy tale style, that attempt to explain many aspects of computer science. Many will be found on the website and you can download them. A great way of reinforcing learning in a way that will appeal to children. You can search by topic or level. Topics covered well exceed the expectations of the KS1 and 2 programme of study.

Barefoot Computing – www.barefootcomputing.org – Already referenced thoroughly above in the coding strand. This is also a good place to find lessons for Computers and Networks.

Progression statements
CS: Computers & networks

LEARNING OBJECTIVES: SKILLS & KNOWLEDGE		Complete lessons			Other Resources	
• Understand the differences between the internet and the world wide web		Teach Computing has some good material in the Y4 lessons			Internet101: What is the Internet	BBC Bitesize: what is the WWW
• Be able to save (and successfully retrieve!) their work to the school network, online and locally to a device. Understand the reasons for saving in different places.		Phil Bagge: Network Internet & Web search planning				
• Understand the function of different externally visible parts of a computer (and peripherals) and classify as input or output devices.		Teach Computing has some good material in the Y5 lessons – L1 Systems				
• Understand the importance of hyperlinks on sites and be able to create some 4						
• Understand that the Internet is a collection on computers (servers) joined together across the world 4			Primary Computing sample Y4 lesson	Teach Computing has some good material in the Y4 lessons		
• Understand the basic structure of your school network, how it is connected (physical wiring, wireless …) and the services that are a part of it (printing, scanning, internet via server …) 4		Phil Bagge: Network Internet & Web search planning	Barefoot: Network hunt activity NEW	Teach Computing has some good material in the Y3 lessons (L6 connecting computers)		

Information Technology

Text & design | Image, film & Sound | Internet | Spreadsheets & Data

KS2 Programme of study extract

Pupils should be taught to:

- **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**.

What does this mean?

This aspect is all about using readymade applications to communicate; to create, gather and share information. It's the sort of thing that teachers have been doing, and an important part of Computing and the whole curriculum.

At the early stages children will be focussing on skill development and awareness of technologies but by the end of key stage two the focus should be shifting to real consideration of audience and purpose and to making sure that what they create is truly fit for purpose and to growing independence in selecting the right tool(s) for the job.

The programme of study does not mention specific technologies or media but the kind of things we should be doing here (in various combinations) are:

- **Text processing and design** – Yes, word processing, desk top publishing (typing skills even, at a basic level). Online this means blogging, posting, commenting, wikis and similar social media tools – Seesaw!
- **Digital image, film and sound** – including animation.
- **Using the Internet –** the focus in this strand on search skills etc.
- **Working with spreadsheets, capturing and processing data –** including data logging

How shall we teach it?

The better question would be "Why?" The answer should really be that it enhances learning in the curriculum as a whole, and there are links here across every subject. This then leads to the "how" question and the answer is within other subject areas. Having said that, there may well be times when computing concepts need to be developed in isolation first and this should be left to individual teacher's discretion. For this reason schools may decide not to timetable this aspect of computing, unless access to kit dictates that this is necessary.

It is most important that a long term planning grid is completed at the start of the year to ensure that there is coverage, perhaps across the two year groups, of a range of computing tools. A possible format for that will be found at herefordshirecomputing.com.

How does it link to other aspects of computing?

- There are strong and important links to **digital literacy (online safety)** when using or creating internet resources or communicating online
- Any work online, searching for resources links strongly to **computer science** where there is a requirement at KS2 to understand the internet, World Wide Web and how search engines rank their results.
- Spreadsheet link to **computer science** and **programming** as soon as formulae are used (which are a sort of programme).

What kit, software and resources are there to help?

Please see the recommended iPad apps list on the website. For precise lesson ideas Mr P ICT (Lee Parkinson) is also a truly excellent resource but you have to subscribe. He really is worth his money though!

Progression statements:

Text & design

- Use different font effects, layout, format, graphics and illustrations to communicate for a given audience.
- Log on to a blog or forum (perhaps Seesaw), create content and send appropriate comments.
- Select suitable text, sounds and graphics (their own and found resources) and use appropriately in their own work
- Create a range of hyperlinks and produce a non-linear, interactive presentations 4
- Insert and edit simple tables etc. 4
- Use Cut, copy and paste to refine and reorder content 4
- Use appropriate editing tools to ensure their work is clear and error free (using tools such as spell checker, thesaurus, find and replace). Recognise the importance of good design. 4
-

Cross curricular application of skills

- Create an interactive digital map of a quest for myths and legends
- Create a PowerPoint / Keynote presentation of a poetry reading incorporating altered digital images and altered recorded speech
- Take photos to create a Maths trail based on shape, symmetry or angles
- Take photos of view through window and annotate to illustrate change
- Write and present a newspaper article about evacuees
- Create a multi-page guide book or a film about a local place of worship
- On a school trip collect images, video, and sound samples and use these to produce a multimedia presentation to show to parents, other etc.
- Create a photomontage of parts of faces from a variety of sources
- Children record vocals and sound effects to accompany story book in Book Creator
- Planning a visit – research route, prices, use email to communicate with companies and with each other
- Use email or video conferencing to communicate with another school within the context of “connecting ourselves to the world”

Image, film & sound

- Add simple titles, music and narration to films.
- Create a short animated sequence using an animation app
- Capture "footage" and use movie editing apps. Arrange, trim and cut clips to create a short film for a given audience.
- Use recorded sound files in other applications
- Use music software or app to experiment with capturing, repeating and reordering sound patterns.
- Use music software / app to create a simple multipart percussion composition 4
- Understand that copyright exists on most recorded music 4
- Discuss and evaluate the quality of their own and others' photos and make decisions (e.g. keep, delete, change) 4
- Understand that evaluation and improvement is a vital part of a design processes and that tech allows us to make changes quickly 4
- Begin to understand the "language of film" 4
- Capture freeze frame in drama to show different shapes and emotions – modify the images
- Use the filters and cutting tools in photo editing software to create their own fantasy settings to support their writing.
- Create a persuasive trailer for a film with sound effects, music, voice-over and scanned children's art
- Take photographs to illustrate different shapes / symmetries for display
- Make animation of seasons, or plant growing, water cycle ...
- Make a video of weather forecasts around the world
- Design wrapping paper for Christmas or DT project
- Create pop art-style images by changing effects of a still image
- Create a photomontage of parts of faces from a variety of sources
- Use digital images to create a collage based on a theme (us Pic Collage)
- Create a spoof TV programme to fit with other work (eg. Weakest Link: Henry VIII's wives; historical Through the Keyhole or cookery programme)
- Create a multimedia presentation or "Ken Burns style" film of a poetry reading incorporating altered images and recorded music
- Create a persuasive trailer for a film, with sound effects, music, voice-over and scanned / photographed children's art work.
- Use sound editing software to investigate sound waves and patterns
- Add spoken French to a presentation / image (MFL)
- Compose music or sound effects to accompany poems, stories, drama etc
- Create a musical portrait of a character for a story or to describe a setting
- Compose pentatonic compositions in Garage Band to accompany films

Progression statements

Cross curricular application of skills

Internet Children’s use of the internet should still be guided by adults. It is not advisable to allow children unsupervised access to search engines.

- Use found information purposefully to complete specific tasks e.g. copy, paste and edit relevant information
- Understand a website has a unique address and the need for precision when typing it
- Understand that some information found through searching is more relevant than others 4
- Develop key questions and key words to search for specific information to answer a problem 4
- Understand the dynamics of search engines and know that there are different search engines - some within sites, and some for the whole of the Internet (e.g. Google). Use them appropriately 4

- Search for resources to support planning and writing
- Read background information on an author, through internet research
- Research a company's website to find out about the manufacturing of a product
- Children explore natural habits using online virtual habitats
- Explore a place of worship by taking a virtual tour
- Explore census on-line information to help build a picture of what local life was like a century ago
- Use the internet to investigate customs of other faiths
- Use a search engine to find photographs and information about children in the second world war or other areas of class work

Spreadsheets & data

- Generate and compare different charts and graphs and understand that different graphs are used for different purposes
- Organise, present, analyse and interpret the data in tables, tally charts, charts / graphs, using IT where appropriate
- Change the contents of cells in a spreadsheet to explore "What if …" questions
- Know, understand and use the vocabulary: file, record, field, data and information.
- Understand that data can be collected more efficiently with technology compared with manual methods.
- Discuss and interpret graphs illustrating the data collection.
- Begin to develop skills to identify what data needs to be collected and design a questionnaire or survey to aid its collection 4
- Raise questions of data and translate them into search criteria 4
- Understand the need to structure information properly in a database or spreadsheet 4
- Use data logging to capture measurements (sound, temperature, light) continuously over time. 4
- Compare experiences of data logging with manual means of collecting similar data. 4
- Explain the story of a line graph
- Use a prepared spreadsheet to make decisions on how pocket money could be spent in a shop.
- Use a spreadsheet to record the results of an experiment investigating friction and use the results to answer hypotheses. (e.g. cars down slopes)
- Explore relationships in personal data (e.g. leg length and long jump)
- Collect weather data and use this as part of their work in comparing weather around the world
- Design a questionnaire to collect data for a spreadsheet or graphing app
- Explore changing variables in pre-designed spreadsheets
- Sc: Statutory requirement - "WORKING SCIENTIFICALLY: making systematic and careful observations and, where appropriate, taking accurate measurements using standard units, using a range of equipment, including thermometers and data loggers"
- Sc: Statutory requirement – "WORKING SCIENTIFICALLY: gathering, recording, classifying and presenting data in a variety of ways to help in answering questions"
- Sc: WORKING SCIENTIFICALLY: Notes and guidance (non-statutory) "… learn how to use new equipment, such as data loggers, appropriately. They should collect data from their own observations and measurements … and help to make decisions about how to record and analyse this data."
- Use a heart monitor to measure heartbeat during different activities.
- Use data logging to compare volume of different sound sources or in different rooms, or in one place over time

KS1 Programme of study extract

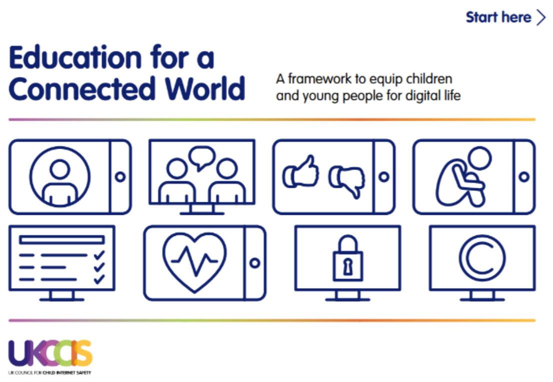
Pupils should be taught to:

- 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.

What does this mean?

Indeed! These are very short and simple statements but behind them there is a great deal to cover. Many ways have been suggested of organising learning (including one in earlier versions of this Progression), but one that is emerging, head and shoulders above others in recent months is Project Evolve.

Project Evolve is slowly providing a wealth of resources to support the statements produced in 2018 in Education for a Connected World. There are still gaps that are slowly being filled but there is a good structure and a splendid set of search tools to help you get to where you need to be.



Start here >

Education for a Connected World

A framework to equip children and young people for digital life

UKCIS

How shall we teach it?

Ideally there should be two approaches to the delivery of this strand of Computing:

- Primarily, online safety needs to be planned and taught systematically, with appropriate links to PSHE and behaviour. The easy way to deal with this would be to teach a unit of online safety each year, possibly half a term. You could use any of the lessons from Project Evolve, additionally the table that follows includes more traditionally used resources that are tried and tested.
- Secondly, you will want to have conversations with your children and if appropriate even ad-hoc lessons, on particular aspects of online safety as they arise. This may be from incidents that have affected the children (at home or in school) or perhaps because there is a strong link to a particular aspect in other teaching. Again, you will find the resources in Project Evolve very useful.

ProjectEVOLVE

Shaping a better online life for all

Access Toolkit

What kit, software and resources are there to help me?

Project Evolve and Education for a Connected World organises learning around 8 strands with learning statements for each year group, starting in Early Years.

Self-image and identity

This strand explores the differences between online and offline identity beginning with self-awareness, shaping online identities and how media impacts on gender and stereotypes. It identifies effective routes for reporting and support and explores the impact of online technologies on self-image and behaviour.

Online relationships

This strand explores how technology shapes communication styles and identifies strategies for positive relationships in online communities. It offers opportunities to discuss relationships and behaviours that may lead to harm and how positive online interaction can empower and amplify voice.

Online reputation

This strand explores the concept of reputation and how others may use online information to make judgements. It offers opportunities to develop strategies to manage personal digital content effectively and capitalise on technology's capacity to create effective positive profiles.

Online bullying

This strand explores bullying and other online aggression and how technology impacts those issues. It offers strategies for effective reporting and intervention and considers how bullying and other aggressive behaviour relates to legislation.

Managing online information

This strand explores how online information is found, viewed and interpreted. It offers strategies for effective searching, critical evaluation and ethical publishing.

Health, well-being and lifestyle

This strand explores the impact that technology has on health, well-being and lifestyle. It also includes understanding negative behaviours and issues amplified and sustained by online technologies and the strategies for dealing with them.

Privacy and security

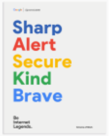
This strand explores how personal online information can be used, stored, processed and shared. It offers both behavioural and technical strategies to limit impact on privacy and protect data and systems against compromise.

Copyright and ownership

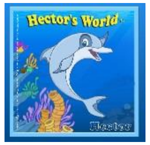
This strand explores the concept of ownership of online content. It explores strategies for protecting personal content and crediting the rights of others as well as addressing potential consequences of illegal access, download and distribution.

Older, trusted, resources for online safety

Lee and Kim's Adventure - 8 minute animation from CEOP introduces Lee & Kim, a brother and sister navigating the online world with the help of their trusted superhero friend SID! Teachers materials provide lesson plans



Google: Be Internet Legends – a brilliant scheme of work for KS2, it can also be used as "pick and mix" resource. Some really well constructed activities for you to use in the classroom. Teachers acan download this free resource and request a free printed copy. **Interland** is an adventuregame environment for chldren to use to reinforce their learning. Posters, certificates and badges are also on offer. Be Internet Citizens is available for KS3&4.



Hector's World - A tried and trusted resource from CEOP's Think U Know materials. An underground adventure for KS1. Hector and his friends are guided through potential traps in their under-water online world. 6 cartoons with complete lesson plans.

Cyber Café – CEOP's KS2 resources are based around their interactive Cyber Café. This is beginning to look a little "long in the tooth" but still very relevant. Teachers' resources centre around nine readymade lessons. You will need to create an account for your school or yourself (free) to these resources

Keeping Myself E-Safe – some resources from Learning Curve Education. A series of excellent animated real life stories packed with ideas. Supported by complete lesson plans and further background videos for teachers. We've paid for you to access these through WMnet but the best way to get them will be to download them from the Herefordshire My Learning site where you will find them in the *Herefordshire ICT Resources* course. You will need your school's login for this, let us know if you need a reminder.

Common Sense Media – Some very comprehensive materials from the USA. Complete lesson plans and full resources are provided. The South West Grid for Learning have produced their own *Digital Literacy and Citizenship* Scheme based on these resources and this is available free of charge.

Cyber Smart – Another complete scheme, this one from Australia. Some really useful material.

Some of the **ChildNet** and **Kid Smart** resources also contain whole lesson resources.

Education for a connected world Y3

1 Self-image and identity

- I can explain what is meant by the term ‘identity’.
- I can explain how I can represent myself in different ways online.
- I can explain ways in which and why I might change my identity depending on what I am doing online (e.g. gaming; using an avatar; social media).

2 Online relationships

- I can describe ways people who have similar likes and interests can get together online.
- I can give examples of technology-specific forms of communication (e.g. emojis, acronyms, text speak).
- I can explain some risks of communicating online with others I don’t know well.
- I can explain how my and other people’s feelings can be hurt by what is said or written online.
- I can explain why I should be careful who I trust online and what information I can trust them with. I can explain why I can take back my trust in someone or something if I feel nervous, uncomfortable or worried.
- I can explain what it means to ‘know someone’ online and why this might be different from knowing someone in real life. I can explain what is meant by ‘trusting someone online’. I can explain why this is different from ‘liking someone online’.

Education for a connected world Y4

- I can explain how my online identity can be different to the identity I present in ‘real life’
- • Knowing this, I can describe the right decisions about how I interact with others and how others perceive me.

- I can describe strategies for safe and fun experiences in a range of online social environments
- I can give examples of how to be respectful to others online.

Education for a connected world Y3

3 Online reputation

- I can search for information about myself online.
- I can recognise I need to be careful before I share anything about myself or others online.
- I know who I should ask if I am not sure if I should put something online.

4 Online bullying

- I can explain what bullying is and can describe how people may bully others.
- I can describe rules about how to behave online and how I follow them.

5 Managing online information

- I can use key phrases in search engines.
- I can explain what autocomplete is and how to choose the best suggestion.
- I can explain how the internet can be used to sell and buy things
- I can explain the difference between a ‘belief’, an ‘opinion’ and a ‘fact’.

Education for a connected world Y4

- I can describe how others can find out information about me by looking online.
- I can explain ways that some of the information about me online could have been created, copied or shared by others.

- I can identify some online technologies where bullying might take place.
- I can describe ways people can be bullied through a range of media (e.g. image, video, text, chat).
- I can explain why I need to think carefully about how content I post might affect others, their feelings and how it may affect how others feel about them (their reputation).

- I can analyse information and differentiate between ‘opinions’, ‘beliefs’ and ‘facts’. I understand what criteria have to be met before something is a ‘fact’.
- I can describe how I can search for information within a wide group of technologies (e.g. social media, image sites, video sites).
- I can describe some of the methods used to encourage people to buy things online (e.g. advertising offers; in-app purchases, pop-ups) and can recognise some of these when they appear online.
- I can explain that some people I ‘meet online’ (e.g. through social media) may be computer programmes pretending to be real people.
- I can explain why lots of people sharing the same opinions or beliefs online does not make those opinions or beliefs true.

Education for a connected world Y3

6 Health, wellbeing and lifestyle

- I can explain why spending too much time using technology can sometimes have a negative impact on me; I can give some examples of activities where it is easy to spend a lot of time engaged (e.g. games, films, videos).

7 Privacy and security

- I can give reasons why I should only share information with people I choose to and can trust. I can explain that if I am not sure or I feel pressured, I should ask a trusted adult.
- I understand and can give reasons why passwords are important.
- I can describe simple strategies for creating and keeping passwords private.
- I can describe how connected devices can collect and share my information with others.

8 Copyright and ownership

- I can explain why copying someone else's work from the internet without permission can cause problems.
- I can give examples of what those problems might be.

Education for a connected world Y4

- I can explain how using technology can distract me from other things I might do or should be doing.
- I can identify times or situations when I might need to limit the amount of time I use technology.
- I can suggest strategies to help me limit this time.

- I can explain what a strong password is.
- I can describe strategies for keeping my personal information private, depending on context.
- I can explain that others online can pretend to be me or other people, including my friends
- I can suggest reasons why they might do this
- I can explain how internet use can be monitored.

- When searching on the internet for content to use, I can explain why I need to consider who owns it and whether I have the right to reuse it.
- I can give some simple examples.

Assessment

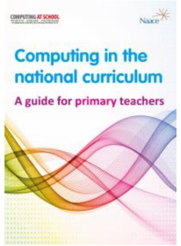
In every national curriculum programme of study the same statement on assessment is to be found:

> 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.

Further DfE guidance has made schools' responsibilities with regard to assessment clear (very much in line with previously):

> Schools will be able to introduce their own approaches to formative assessment, to support pupil attainment and progression. The assessment framework should be built into the school curriculum, so that schools can **check what pupils have learned and whether they are on track to meet expectations at the end of the key stage, and so that they can report regularly to parents.** Full DfE article here

Miles Berry, in his excellent publication *Computing in the national curriculum: A guide for primary teachers* (pp 22-25) makes some very helpful suggestions as to how we might approach formative and summative assessment. Briefly these are:



Self-assessment – where each child maintain a blog of their work; attaching examples and reflecting on learning
Peer-assessment –perhaps through shared blog entries and comments on peers' blogs
Open questioning – by teachers
Target setting - using KWL lists (what pupils already know, want to learn, what they have learned)

Were teacher assessment takes place, teachers will use their professional judgement to determine the most effective method of gathering evidence of progress but in computing it will certainly require knowledge of the context in which work was completed rather than simple scrutiny of a finished outcome.

One good approach is to consider, perhaps on an annual basis, what a child has accomplished for each of the strands (CS, IT, DL). Then take into account attainment across all aspects and adopt a "best fit" approach when arriving at an overall judgement.

Miles Berry offers a breakdown of the programme of study statements to create a hierarchy (or progression) of learning. An adapted version of his approach is below **in the simple assessment grid**. The 2014 version of these materials contained three more complex assessment grids which were not widely used.

SIMPLE ASSESSMENT GRID			
	COMPUTER SCIENCE	INFORMATION TECHNOLOGY	DIGITAL LITEACY
1	• Understand what algorithms are • Create simple programs • Understand that algorithms are implemented as programs on digital devices • Recognise common uses of information technology beyond school	• Use technology purposefully to create digital content • Use technology purposefully to store digital content • Use technology purposefully to retrieve digital content	• Use technology safely • Keep personal information private
2	• Understand that programs execute by following precise and unambiguous instructions • Debug simple programs • Use logical reasoning to predict the behaviour of simple programs	• Use technology purposefully to organise digital content • Use technology purposefully to manipulate digital content	• Use technology respectfully • Identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies
3	• Write programs that accomplish specific goals • Use sequence in programs • Work with various forms of input • Work with various forms of output	• Use search technologies effectively • Use a variety of software to accomplish given goals • Collect information Collect data • Design and create content • Present information	• Use technology responsibly • Identify a range of ways to report concerns about contact
4	• Design programs that accomplish specific goals • Design and create programs • Debug programs that accomplish specific goals • Use repetition in programs • Control or simulate physical systems • Use logical reasoning to detect and correct errors in programs • Appreciate how search results are selected	• Select a variety of software to accomplish given goals • Select, use and combine internet services • Analyse information • Evaluate information • Present data • Understand the opportunities computer networks including the internet offer for communication	• Identify a range of ways to report concerns about content • Recognise acceptable/unacceptable behaviour
5	• Solve problems by decomposing them into smaller parts • Use selection in programs • Work with variables • Use logical reasoning to explain how simple algorithms work • Use logical reasoning to detect and correct errors in algorithms • Understand how computer networks can provide multiple services, such as the World Wide Web	• Combine a variety of software to accomplish given goals • Select, use and combine software on a range of digital devices • Design and create systems • Analyse data	• Be discerning in evaluating digital content
6	• Understand computer networks, including the internet • Appreciate how search results are ranked	• Understand the opportunities computer networks including the internet offer for collaboration • Evaluate data	• Be discerning in evaluating digital content