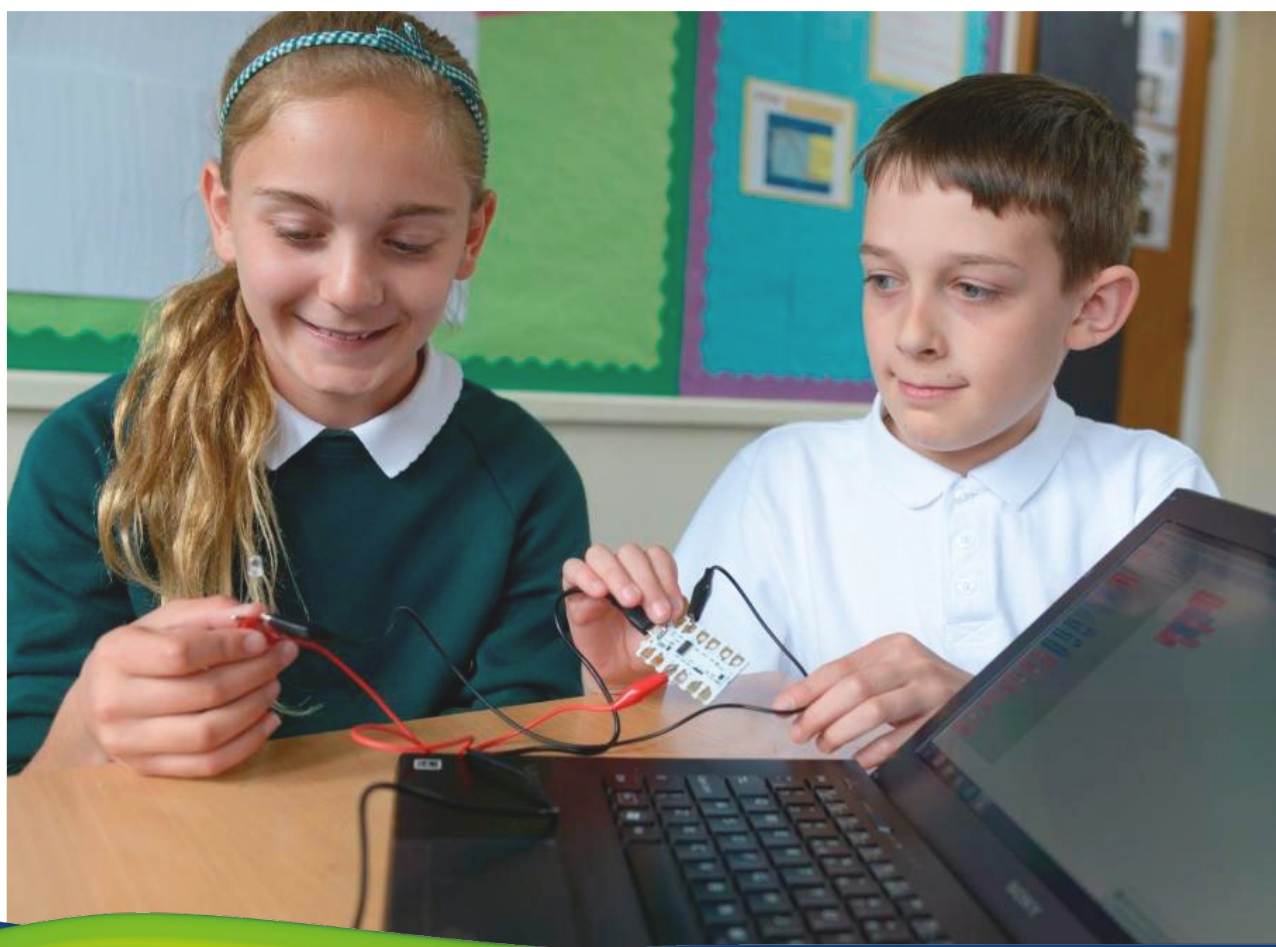


Applying Computing in D&T at KS2 and KS3:

the 2014 National Curriculum requirements



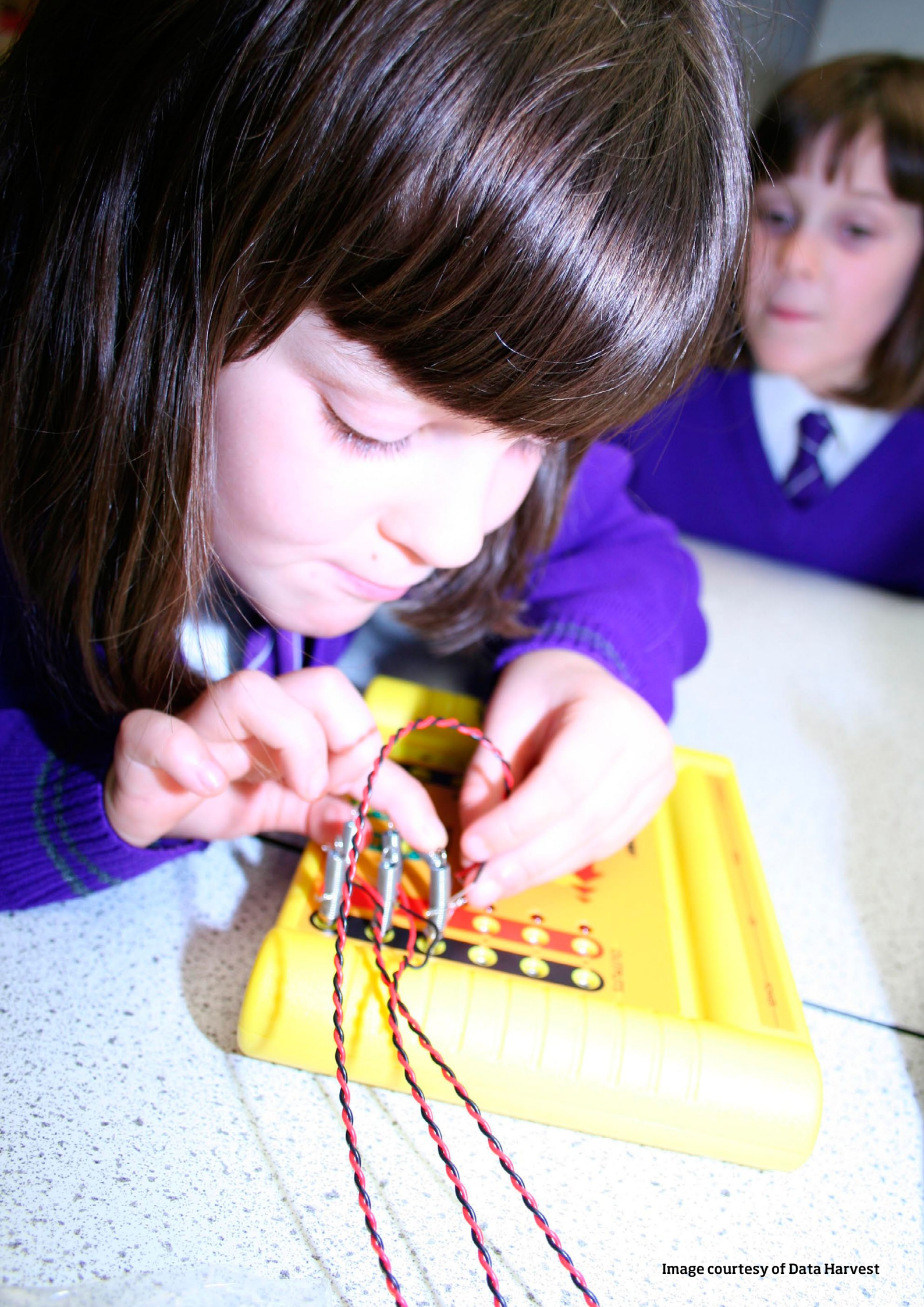
Introduction

The purpose of this guidance is to help primary and secondary teachers in England introduce the National Curriculum 2014 requirements that link computing with design and technology (D&T).

It explains in general terms how the two subjects are related and then focuses on the application of computing in design and technology to program and control physical systems in KS2 and KS3.

The guidance sets out:

- The nature of D&T education and computing education
- Points of overlap between the two subjects in the 2014 National Curriculum requirements
- Guiding principles for curriculum planning, teaching and learning
- A rationale for linking computing and D&T in order to teach programming and control
- Progression in pupils' learning in programming and control



The nature of D&T education and computing education

D&T and computing are based on principles and ways of working that are unique but complementary.

Design and technology

At the heart of D&T is the **'iterative' process of designing and making**. Through this process pupils' ideas are generated, externalised, communicated and evaluated through action. During an iterative process thought leads to action, resulting in further thought and action as pupils resolve design problems and address design opportunities. This is an intensely creative, reflective, evaluative way of working.

To ensure a genuine design and technological experience, iterative designing and making should take place within projects which address the following principles, agreed by the Expert Subject Advisory Group (ESAG) for D&T:

User – pupils should have a clear idea of who they are designing and making products for, considering their needs, wants, values, interests and preferences. The intended users could be themselves or others, an imaginary or story-based character, a client, a consumer or specific target group.

Purpose – pupils should be able to clearly communicate the purpose of the products they are designing and making. Each product they create should be designed to perform one or more defined tasks. Pupils' products should be evaluated through use.

Functionality – pupils should design and make products that work/function effectively in order to fulfil users' needs, wants and purposes.

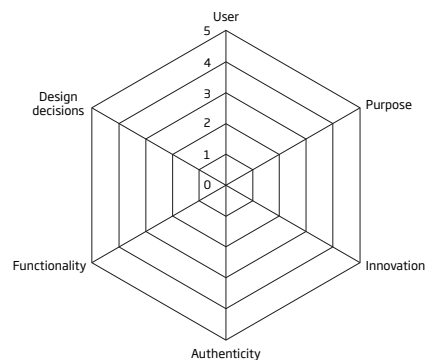
Innovation – when designing and making, pupils need some scope to be original with their thinking. Projects that encourage innovation lead to a range of design ideas and products being developed and are characterised by engaging open-ended starting points for learning.

Design decisions – pupils need opportunities to make their own design decisions. Making design decisions allows pupils to demonstrate their creative, technical and practical expertise, and draw on learning from other subjects. Through making design decisions pupils decide on the form their product will take, how their product will work, what task or tasks it will perform and who the product will be for.

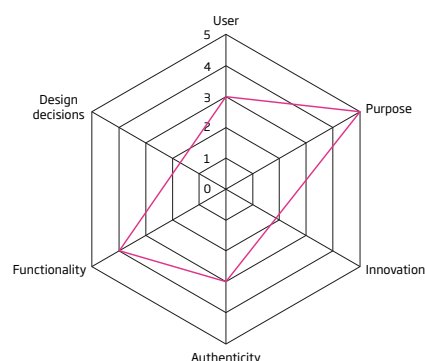
Authenticity – pupils should design and make products that are believable, real and meaningful to themselves and others.

Every design and technology project should include, to a greater or lesser extent, all of the D&T principles.

A star diagram can be used to rate the potential of D&T projects during curriculum planning and when evaluating the quality of teaching and learning.



Example



Computing

At the heart of computing is '**computational thinking**'.

Computational thinking embodies the set of mental skills that convert complex and perhaps partially defined problems into solutions which can be solved by a computer. Computational thinking includes the following concepts:

Logic (predicting and analysing) – logical reasoning helps us explain why something happens. It helps us predict what will happen and then explain an outcome. If you set up two computers in the same way, give them the same instructions (the program) and the same input, you can pretty much guarantee the same output. Thus computers are predictable, they really are! Because of this pupils should use logical reasoning to predict what an algorithm, program or computer system will do and to detect and correct errors in algorithms and programs.

Algorithmic thinking (making steps and rules) – an algorithm is a sequence of instructions or a set of rules to get something done. There are plenty of algorithms that will accomplish the same goal. All pupils should be given opportunities to develop algorithms to solve problems, consider alternative algorithms to solve the same problem and consider most appropriate algorithms for a given situation.

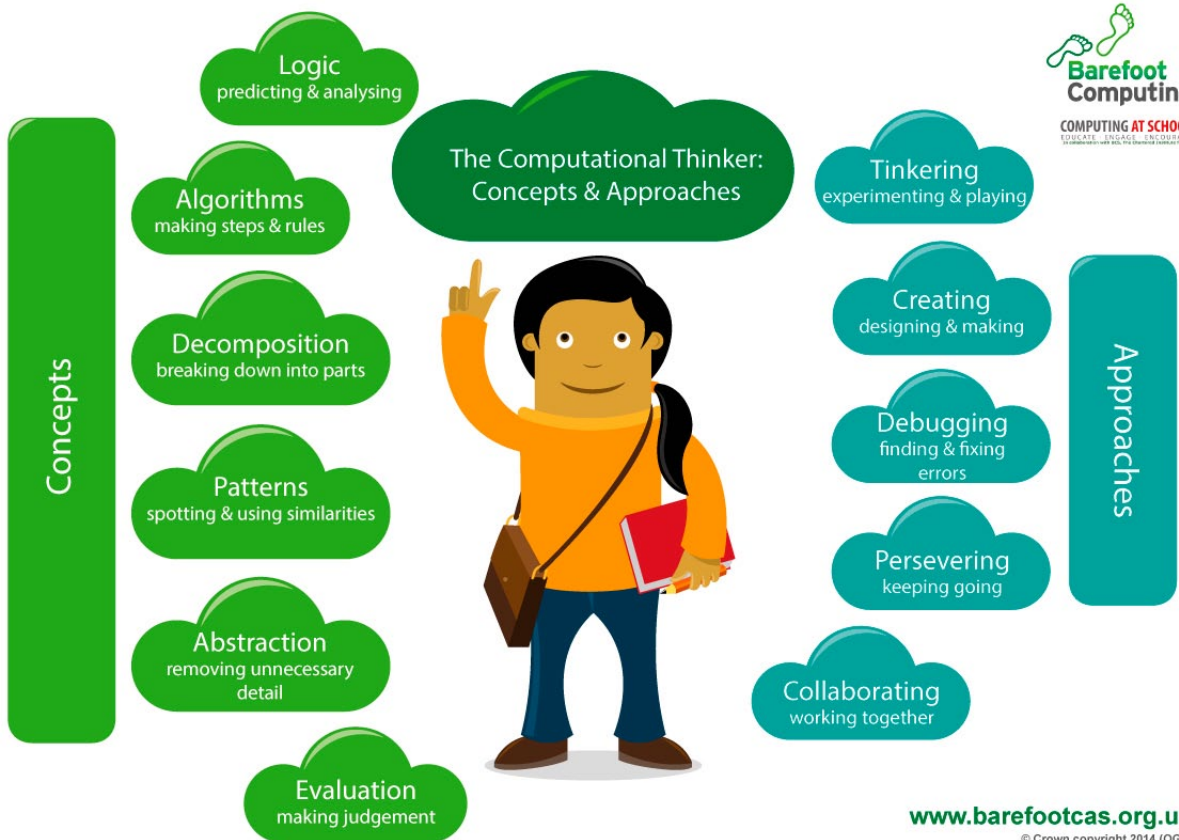
Decomposition (breaking down into parts) – the process of breaking down a problem into smaller manageable parts is known as decomposition. Decomposition helps us solve complex problems and manage large projects.

Patterns (spotting and using similarities) – patterns are everywhere, for example, we use weather patterns to create weather forecasts; children might notice patterns in how teachers react to their behaviour to work out how to behave next time. By identifying patterns we can make predictions, create rules and solve more general problems. In computing, the method of looking for a general approach to a class of problems is called generalisation.

Abstraction (removing unnecessary detail) – abstraction is about simplifying things; identifying what is important without worrying too much about the detail. Abstraction allows us to manage complexity. We use abstractions all the time! Famous examples would include the London Tube Map or even a school timetable which only captures information such as who is taught what subject where and by whom, but leaves to one side further layers of complexity, such as the learning objectives and activities planned in any individual lesson.

Evaluation (making judgements) – evaluation is about making judgements, in an objective and systematic way where possible. In computer science, evaluation is systematic and rigorous; it is about judging the quality, effectiveness and efficiency of solutions, systems, products and processes. Evaluation checks that solutions do the job they are designed to do and are fit for purpose.

Each aspect is best realised through designing, implementing and debugging a solution to a known problem but are also seen when applied to theoretical aspects of computer science as well and will enable pupils to better conceptualise and understand their digital world.



Barefoot would like to acknowledge the work of Julia Briggs and the eLIM team at Somerset County Council for their contribution to this poster.

National Curriculum requirements

The **2014 National Curriculum** requires schools to link computing with D&T. The **Purpose of Study** for each subject makes this clear:

Design and technology

Design and technology is an inspiring, rigorous and practical subject. Using creativity and imagination, pupils design and make products that solve real and relevant problems within a variety of contexts, considering their own and others' needs, wants and values. **They acquire a broad range of subject knowledge and draw on disciplines such as mathematics, science, engineering, computing and art.** Pupils learn how to take risks, becoming resourceful, innovative, enterprising and capable citizens. Through the evaluation of past and present design and technology, they develop a critical understanding of its impact on daily life and the wider world. High-quality design and technology education makes an essential contribution to the creativity, culture, wealth and well-being of the nation.

In the **Subject Content for D&T** there are requirements that link the two subjects. Please note that in D&T there are no KS1 requirements for programming and control.

At KS1 pupils should:

'generate, develop, model and communicate their ideas through ... where appropriate, information and communication technology'

At KS2 pupils should:

'generate, develop, model and communicate their ideas through ... computer-aided design'

'apply their understanding of computing to program, monitor and control their products'

At KS3 pupils should:

'develop and communicate design ideas using ... 3-D and mathematical modelling, ... and digital presentations and computer-based tools'

'select from and use specialist tools, techniques, processes, equipment and machinery precisely, including computer-aided manufacture'

'investigate new and emerging technologies'

'apply computing and use electronics to embed intelligence in products that respond to inputs [for example, sensors], and control outputs [for example, actuators], using programmable components [for example, microcontrollers]'



Computing

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

In the **Subject Content for Computing** there are requirements related to D&T.

At KS1 pupils should:

'generate, develop, model and communicate their ideas through ... where appropriate, information and communication technology'

'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'

At KS2 pupils should:

'design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts'

'work with 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'

'design and create a range of programs, systems and content that accomplish given goals'

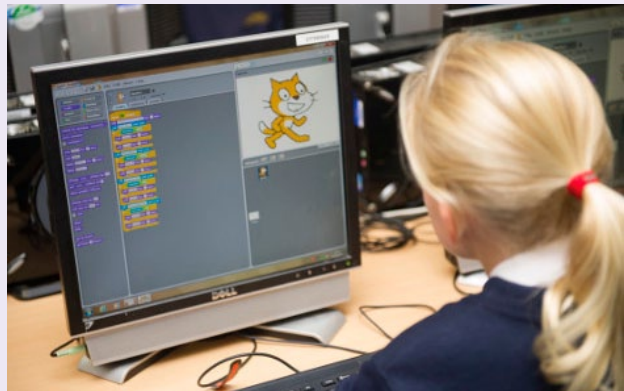
At KS3 pupils should:

'design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems'

'use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures or functions'

'understand simple Boolean logic and some of its uses in circuits and programs'

'understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems'



Curriculum planning, teaching and learning

The following criteria should be used to guide curriculum planning, teaching and learning.

Effective links between computing and D&T:

- **enhance** the quality of teaching and learning in both subjects
- provide a **natural overlap** in pupils' learning
- maintain the **distinctive nature** of both computing and D&T
- **match learning** in the same year groups in both subjects
- use the **same approach** to developing and applying computational thinking in both subjects
- use the **same resources** in both subjects, where appropriate
- **make sense** to pupils, enabling them to make connections between topics and subjects and see the bigger picture

Programming and control – why link computing and D&T?

Through programming and controlling physical systems, D&T offers real, relevant and practical contexts for pupils to develop and apply both computational thinking and coding skills. Creating programs in order to control products that pupils have designed and made themselves is a highly motivating, tangible experience, enabling them to test out and develop their capability in computer science within a range of authentic contexts.

The application of computer science enhances learning in D&T by challenging pupils to improve the effectiveness of their products. Importantly, it has the potential to ensure that what pupils design and make in the classroom reflects the design and technology they encounter in the wider world – at home, in school, in the local community and in industry.

As part of everyday life, pupils come into contact with hi-tech products that are controlled by computers, including embedded microprocessors. Both computing and D&T should build on these experiences if they are to prepare pupils adequately for later life.



Image courtesy of Data Harvest

Examples of programming and physical control

Example of effective teaching and learning in KS1:

In computing, Y1 pupils are introduced to a programmable toy, such as the Bee-Bot. They are first given an opportunity to find out how the toy works by tinkering with it, using logical reasoning to build an understanding of how the programming language for the toy works. They are then given a challenge to solve, such as requiring the toy to navigate between two places. Pupils then create an algorithm to solve the challenge, e.g. move forward 2, right turn, move forward 3. The algorithm is much like a plan. Pupils test out their algorithm by physically tracing out the route, predicting what will happen. They then use their algorithm to help them actually program the toy. They use the keys on the toy to program the commands e.g. fwd command, fwd command, right command, fwd command, fwd command, fwd command. When the program runs, pupils debug the program, fixing any errors and evaluating whether they met the challenge set.

Examples of challenges might include moving the toy to a named person in a circle, creating a times tables mat and moving the toy from a question to the correct answer, travelling to and stopping at a particular target on the floor, demolishing a construction kit tower and finding and collecting treasure (using a magnet and paper clips). For each activity, pupils design an algorithm, predict what will happen, press keys to write the program, make observations and debug, modify and evaluate their algorithms and programs.

Example of effective teaching and learning in KS2:

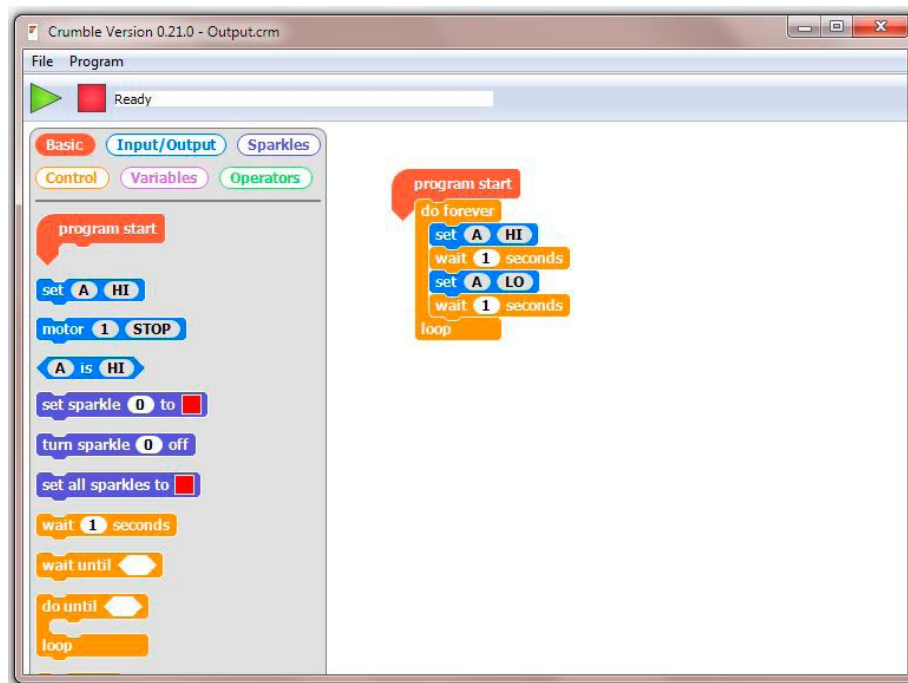
In computing, Y5 pupils are given a challenge to create an automated night light for a younger child. They design an algorithm, perhaps drawing a set of labelled diagrams or storyboard of what the night light will do. They then use their algorithm to create an on-screen prototype. They use a block-based programming language, such as Scratch, to write and debug their prototype. In doing this they learn how to design, write and debug programs using sequence, selection 'if... then' statements, and repetition. They then use what they have learned from their prototype to modify their algorithm and use their improved design to help them build a physical system. For example, they might use a Crumble interface, which uses a Scratch-inspired programming language to connect, program and control input devices such as switches and sensors, and output devices such as bulbs and buzzers, replacing their on-screen prototype with a physical system.

In D&T, the same class design and make manually-controlled, battery-powered night lights with a specific user and purpose in mind. To inform their design decisions, they investigate a range of components including switches, bulbs and light emitting diodes (LEDs), as well as applying an understanding of how night lights are controlled through electrical systems that incorporate an input, process and output.

Pupils are challenged by their teacher to think about how their night lights could be improved for the benefit of the intended user by using the knowledge and skills they have learnt in computing. They suggest that their night lights could work automatically, by programming them to turn on at dusk and off at dawn, as well as switching on and off in a repeating pattern to amuse the intended user before going to sleep. The pupils connect a Crumble interface to a light dependent resistor (LDR) and to the bulbs or LEDs in their night lights, and apply their understanding of computing to program, monitor and control their products.

N.B. In the primary classroom, light emitting diodes (LEDs) with internal resistors should be used.

The crumble interface uses a 'Scratch-like' programming language for controlling physical objects



Example of effective practice in KS3:

In D&T, pupils are set the task to design and make a prototype board game which contains 'intelligence' – it is able to sense the player's pieces on the board and at random creates lighting or sound effects which contribute to the playing of the game. They explore games already available and decide who their game is for. They use 2D/3D computer-aided design (CAD) packages in order to develop designs for component parts e.g. counters/players and the board and use computer aided manufacturing (CAM) including 3D printers to make it. They build in sensor triggers (input) and LEDs, speaker/transducer (output) which are connected to a printed circuit board including a PIC chip that they assemble.

In computing, pupils learn to programme using Picaxe, writing code and building on their previous programming of devices using icon based software. They develop the skills which are then used in D&T lessons to develop and test the program required to operate the game board as described in their specification.

Progression in pupils' learning in programming and physical control

KS1 - computing only

Pupils should know how to create and debug simple programs using a digital devices such as a programmable toy. They understand that the toy does what is expected when it is given one or more precise instructions in the correct sequence. They use logical reasoning to predict what the toy will do. They understand that how far the toy moves and the direction it turns are determined by the instructions they enter and that it stores instructions. When creating a program for a simple programmable toy, pupils start to learn about algorithms. As they plan the solution they jot down the route, then test this out and implement it by programming their toy.

Early KS2 - computing and D&T

Pupils should know how to program a computer to control products they have designed and made. This can be achieved either with a stand-alone computer control box, an interface connected to a desktop or laptop computer or by using a programmable controller embedded in their product. Programmable controllers are

programmed when connected to a desktop, laptop computer or mobile device. The application of pupils' computing knowledge and skills should focus on physically controlling 'output' devices, such as bulbs, buzzers, electric motors and light emitting diodes (LEDs). They learn to use a graphical programming language to control a single output device by turning it on and off according to the algorithm they designed and a sequence of instructions or rules. This can be developed by pupils controlling more than one output device at the same time, such as a two motors, a number of LEDs, a bulb and a buzzer. As pupils develop more complex algorithms, the idea of repetition should also be introduced so that sequences of instructions can be repeated.

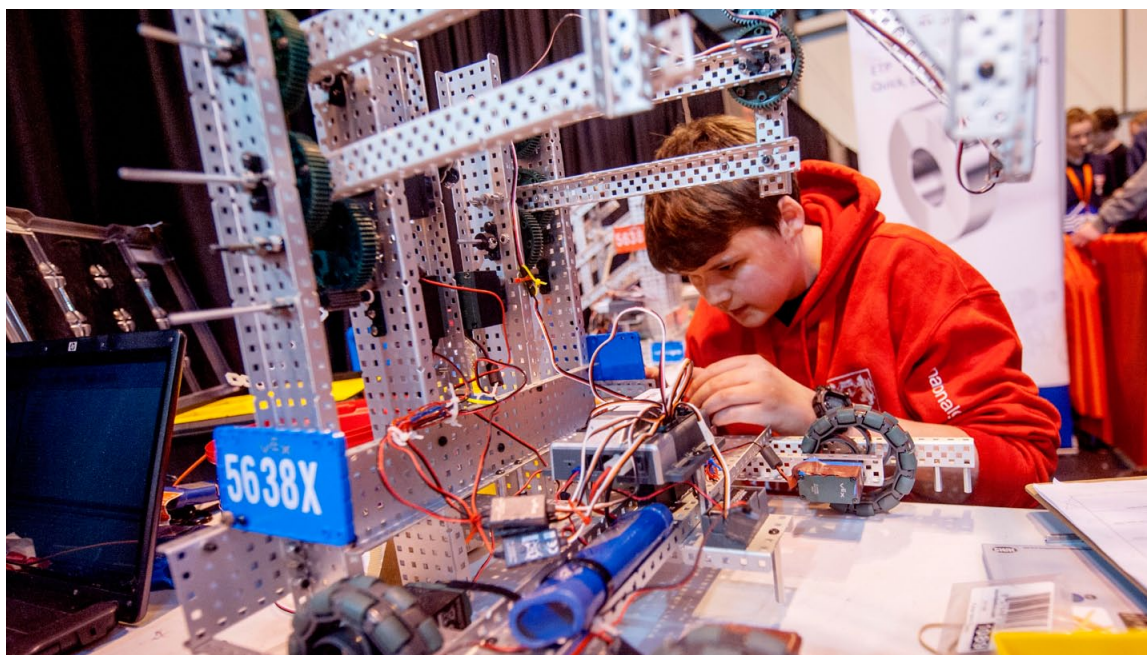
N.B. In the primary classroom, light emitting diodes (LEDs) with internal resistors should be used.

Late KS2 - computing and D&T

Pupils should know how to program a computer to monitor changes in the environment and control their products. They should develop



Image courtesy of IET Lego FIRST League



an understanding of 'monitoring' as well as control and the idea of 'input' as well as 'output'. The application of pupils' learning from computing should focus on using input devices such as switches and sensors. They learn that it is possible to connect input devices such as light dependant resistors (LDRs), reed switches, push-to-make switches, pressure pads and toggle switches to an interface. By the introduction of an input which controls the output, pupils develop an understanding of selection, 'if...then' logic. They design algorithms using selection and implement these as they control and monitor products they have designed and made. Pupils need to keep track of state of their product, such as whether a bulb is on or off; in doing this they are introduced to the use of variables.

Early KS3 - computing and D&T

Pupils should understand how more advanced electrical and electronic systems can be powered and used in their products. They should develop their programming skills to control devices that they have constructed using kits, such as robots built with Lego, using more sophisticated routines including selection (if... then) using a graphical programming language. They should develop their ability to assemble systems on printed circuit boards (PCBs) using specialist electronic equipment such as soldering irons and incorporating programmable components such as PIC chips. They will build and develop products that incorporate systems that make use of for example, sensors to detect heat, light and sound, and control movement using simple actuators such as motors.

Late KS3 - computing and D&T

Pupils should develop their ability to assemble more complex systems on PCBs that they have designed and tested using electronic PCB design simulation software and assembled on boards they have produced themselves. They develop their ability to apply computing and use electronics to embed intelligence in products that respond to inputs, using microprocessors they program using text-based languages. They apply the concepts of feedback in systems and how to control outputs such as actuators and motors, including stepper motors, for instance in the programming of pre built models or robots constructed from kits such as the system produced by VEX.

Glossary of technical terms

- See: www.teachprimarycomputing.org.uk/glossary
- For D&T terminology see <https://www.data.org.uk/media/1123/annotated-programme-of-study.pdf>

Possible hardware and software for programming and physical control

- <https://www.data.org.uk/media/1446/resources-for-programming-and-control.pdf>



Image courtesy
of TTS

As the UK's national academy for engineering, we bring together the most successful and talented engineers from across the engineering sectors for a shared purpose: to advance and promote excellence in engineering.

We provide analysis and policy support to promote the UK's role as a great place from which to do business.

We take a lead on engineering education and we invest in the UK's world class research base to underpin innovation. We work to improve public awareness and understanding of engineering. We are a national Academy with a global outlook.

The Academy's work programmes for 2011 to 2015 are driven by four strategic challenges:

Drive faster and more balanced economic growth

Foster better education and skills

Lead the profession

Promote engineering at the heart of society



Royal Academy of Engineering
Prince Philip House, 3 Carlton House Terrace, London SW1Y 5DG

Tel: +44 (0)20 7766 0600
www.raeng.org.uk

Registered charity number 293074