



*Only our best will do!*

## **Albrighton Primary School Computing Policy**

This policy is based on the new Computing programmes of study: Key stage 1 and 2 National curriculum in England document; see Appendix 2.

### **Our Vision**

Through effective teaching and integration of computing technologies within Albrighton Primary School, we are aiming to provide all children with quality first hand experiences which will enhance their learning and prepare them for life in a world where they will encounter technology every day.

### **Aims of the new computing curriculum**

The aim of the New Computing Curriculum is to develop both skills and knowledge. Children are to be taught computer science and learn about programming, data, algorithms and networks. This enables children to develop an understanding of the principles of computer science. They develop computational thinking. There is also a focus on problem solving: using logic and ideas about systems, patterns (and pattern languages), abstraction and decomposition.

The new curriculum puts a clearer emphasis on three areas of learning:

**Computer science (CS)** - how computers work and how to write algorithms and solve problems to eventually create a computer program.

**Information technology (IT)**- how data is represented and managed on computers.

**Digital literacy (DL)**- how to understand digital information and interact with it safely and appropriately.

The following table is taken from the Computing at School/NAACE publication for Primary Schools and provides guidance as to what will be covered under each of the three areas.

	KS1	KS2
CS	<p>Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions</p> <p>Create and debug simple programs</p> <p>Use logical reasoning to predict the behaviour of simple programs</p>	<p>Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts</p> <p>Use sequence, selection, and repetition in programs; work with variables and various forms of input and output</p> <p>Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs</p> <p>Understand computer networks including the internet; how they can provide multiple services, such as the World Wide Web</p> <p>Appreciate how [search] results are selected and ranked</p>
IT	<p>Use technology purposefully to create, organise, store, manipulate and retrieve digital content</p>	<p>Use search technologies effectively</p> <p>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</p>
DL	<p>Recognise common uses of information technology beyond school</p> <p>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</p>	<p>Understand the opportunities [networks] offer for communication and collaboration</p> <p>Be discerning in evaluating digital content</p> <p>Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact</p>

The aim of the new curriculum is to build an understanding of how computers work and how they can be used in pupils' lives - both in their future employment and in enabling them to be good 21st Century global citizens.

At Albrighton Primary School, we are developing our teaching of computing and are working towards ensuring that pupils have the opportunity to use ICT within the class across a range of subjects. Children have access to both PCs. In the near future there will also be access to iPads. We are exploring the range of software and apps to ensure that children are given the best possible opportunity to access the new computing curriculum.

An important part of the teaching and learning is for pupils to find out and share how to keep themselves safe on line.

## **EYFS and Key Stage 1**

As in other areas of the curriculum, the new computing curriculum will require that children have knowledge and experience in a range of technologies. We must ensure that our teaching and learning continues and enhances prior knowledge and experience that children will bring from home. We will do this through the sharing of planning, having discussions regarding resources and software, regular communication and meetings.

## **Non Negotiables for teaching staff**

The following are minimum expectations for staff when planning and delivering computing lessons and when using information technology across the curriculum. There are also a set of guidelines for pupils which outline the expectations of them when using technology in school.

- **Clear objectives should be set using subject specific vocabulary.**
- **Lessons should build on pupils' existing learning and rehearse familiar concepts with the pupils before they move on to something new.**
- **Computing lessons should be planned to include lots of hands-on activities with a real purpose**
- **Sessions should include 'off screen activities' to support pupil learning.**
- **Pupil to pupil learning should be planned and valued.**
- **Pupils should be given the opportunity to make mistakes and try things out.**
- **E-safety should be integral to all work in computing.**

## **Further explanation and rationale of non-negotiables**

**Clear objectives should be set using subject specific vocabulary.** *Computing has a subject-specific vocabulary just like other areas of the curriculum. Some of these words will be totally new perhaps, like 'debug' and others might not be new, but have different meanings in the context of computing.*

**Lessons should build on pupils' existing learning and rehearse familiar concepts with the pupils before they move on to something new.** *Familiarity and reinforcement are an important part of using technology enabling pupils to ensure that they are using the correct and most efficient procedures.*

**Computing lessons should be planned to include lots of hands-on activities with a real purpose** – *for example creating games for friends to play or creating a website to show/share research and learning.*

**Sessions should include 'off screen activities' to support pupil learning.** *It is especially important for pupils to understand how technology relates to the real world and how it fits into "real life". Giving children concrete examples.*

**Pupil to pupil learning should be planned and valued.** *Children should be given the opportunity to find things out for themselves and share their knowledge. Using freely available software means that pupils can download it and practice their skills at home.*

**Pupils should be given the opportunity to make mistakes and try things out. Trial and error is a major way that pupils learn to program.** *Pupils should be given the opportunity to self and*

*peer evaluate and this should be part of overall teacher assessment. When pupils are debugging ready-made code blocks this is a powerful form of learning – it develops the crucial skill of problem solving. This is bit like taking apart a model to understand how it is made and it means that pupils can make changes to the code, like changing the distance a sprite travels or costumes, and see the effect immediately.*

**E-safety should be integral to all work in computing.** Discussions and exploration of e-safety issues should be planned into computing/PHSE sessions. Refer to the 'SMART' acronym used on the Childnet website.

**Non Negotiables for pupils (based on the pupil developed Acceptable Use Policy and updated in the light of the new curriculum).**

- Keep any passwords safe at all times
- Log off properly and ensure machines are switched off where necessary. Headphone wires should be wrapped around headsets and stored correctly.
- Treat equipment with respect – report any breakages to an adult. Be honest!
- Don't say anything electronically that you wouldn't be prepared to say to someone's face
- Mobile phones are not allowed in school (can be left at the school office).
- Don't use images of other people unless you have permission.
- Keep personal information private.
- Only add people you know to your list of friends.
- Only use appropriate games or websites.
- If anything bad happens tell an adult. Use the Hector Protector to cover a screen first of all.
- Children are not allowed to use school computers unsupervised.

### **Long Term Planning**

Due to the introduction of the new curriculum in September 2014, our long term planning is currently under review. We are in the process of introducing new technologies, use of equipment, apps and software into both our dedicated computing teaching and all areas of the curriculum.

### **Assessment and Progression**

With the removal of levels from the National Curriculum, we will use "I can" progression statements to assess children's knowledge and understanding. The outline of progression given below (*taken from Computing in the national curriculum: A guide for primary teachers*) will form the basis for our assessments and this is an on-going area for development as we introduce the new curriculum.

<b>Computing Science</b>	<b>Information Technology</b>	<b>Digital Literacy</b>
Understand what algorithms are Create simple programs	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 Recognise common uses of information technology beyond school
Understand that algorithms are implemented as programs on digital devices 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
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 Design and create content Present information	Use technology responsibly Identify a range of ways to report concerns about contact
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 Understand how computer networks can provide multiple services, such as the World Wide Web 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 Collect data Present data	Understand the opportunities computer networks offer for communication Identify a range of ways to report concerns about content Recognise acceptable/unacceptable behaviour
Solve problems by decomposing them into smaller parts Use selection in programs Work with variables Use logical reasoning to explain how some simple algorithms work Use logical reasoning to detect and correct errors in algorithms Understand computer	Combine a variety of software to accomplish given goals Select, use and combine software on a range of digital devices Analyse data Evaluate data Design and create systems	Understand the opportunities computer networks offer for collaboration Be discerning in evaluating digital content

networks, including the internet Appreciate how search results are ranked		
--	--	--

**Time Allocation**

Children will receive one hour weekly dedicated computing teaching time focussing on skills development. In addition to this, staff are expected to plan for the use of new technologies in English, Maths and Science lessons and across the curriculum.

**Monitoring and Development**

Computing will be monitored in line with the school's monitoring policy and the school development plan.

### **Staff Development**

It is recognised that computing is an area of the curriculum that requires detailed technical knowledge and a large investment of time in learning and developing knowledge of new technologies. Where possible, the ICT coordinator and visits of the computing technician will support and develop staff knowledge. This will include the provision of after-school training sessions for teaching staff and teaching assistants and the inclusion of a computing element in PD days where appropriate.

Details of courses will be made available to all staff and funding provided in accordance with the school priorities as outlined in the school development plan.

Regular e-safety training for staff will be provided so that all staff members are knowledgeable and confident in supporting pupils in staying safe on-line.

### **Equal Opportunities**

All children are offered equal opportunities to participate in all areas of the computing curriculum.

**This Policy was updated on February 2015.**

**Due to be reviewed: February 2017**

## Computing programmes of study: key stages 1 and 2

### National curriculum in England

#### Purpose of study

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

#### Aims

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

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

#### Attainment targets

By the end of each key stage, pupils are expected to know, apply and understand the matters, skills and processes specified in the relevant programme of study.

**Schools are not required by law to teach the example content in [square brackets].**

# Subject content

## Key stage 1

Pupils should be taught to:

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- create and debug simple programs
- use logical reasoning to predict the behaviour of simple programs
- use technology purposefully to create, organise, store, manipulate and retrieve digital content
- recognise common uses of information technology beyond school
- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

## Key stage 2

Pupils should be taught to:

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

© Crown copyright 2013

Reference: DFE-00171-2013 10

## Appendix 2: Glossary of terms

This is a glossary of terms in the computer science context – the words may sometimes (but not always) have different meanings in other contexts.

- **Abstraction** Reducing complexity by focusing on the essential features of an algorithm or data representation and omitting unnecessary detail.
- **Algorithm** A precise step by step method for solving a problem. Named after Iranian mathematician Muhammad ibn Musa al-Khwarizmi.
- **Application** A self-contained program that performs a specific function for end users.
- **Boolean logic / digital logic** A system of reasoning with truth values, true and false, using logical operations such as and, or, and not. Named after British mathematician George Boole.
- **Browser** A piece of software that enables a user to locate, retrieve and display information on the world wide web.
- **Complexity** The way that a solution to a problem scales as the size of the input increases, considering both the number of computational steps and the memory space required.
- **Computation** Performing a calculation by executing the instructions of a program on a computational device.
- **Computational device / digital device** A computer or other programmable device that performs computation.
- **Computational thinking** The ability to analyse ways to solve problems using appropriate algorithms and data representations, taking account of the complexity of possible solutions.
- **Computer science** The scientific study of computation, applied to both hardware and software, covering both theoretical and practical concerns.
- **Data** Information which can be stored, retrieved and manipulated in digital form using digital devices.
- **Debug** To find, remove and / or change errors in computer code.
- **Digital artifacts / digital content** Images, videos, text or data, or a combination of these, which are made on a digital device.
- **Digital literacy** The ability to access and manipulate digital content and understand the implications of its creation and distribution.
- **Data representation** The various ways data can be represented as words, numbers and pictures in the memory of a computational device.
- **E-safety** Understanding and applying rules to mitigate against the risks to personal safety and privacy of personal information in using digital devices of all kinds.
- **Function** A small section of computational code that performs a specific operation. In particular, a function takes inputs, or arguments, and returns outputs, or results.
- **Hardware** Physical items of computing kit such as desktop hard drives, printers and scanners
- **Input** The data that feeds into a computation. A stimulus to which a real-time application will respond.
- **Internet** A global network of computers which are linked, allowing the exchange of data. It uses various data transfer protocols which can be seen in the top bar of a browser, such as Hyper Text Transfer Protocol (http). 11

- **Logic** A systematic approach to reasoning. The rules that underlie an algorithm used for an application. Can also refer to digital components in computer hardware.
- **Network** A number of computational devices connected together, allowing sharing of resources and cooperation between devices in the solution of a problem. Also the hardware used to establish connections between devices on a network.
- **Operating system** The program that enables the computer to start and access different sorts of software on the computer, examples include Microsoft Windows and iOS for Mac.
- **Output** The data that results from a computation. A response generated by a real-time application to stimulus.
- **Procedure** A small section of computational code that performs a specific operation. Unlike a function, a procedure does not return a result but may be to change the data stored in objects in computer memory.
- **Program / code (verb)** To create or modify a program.
- **Program / code (noun)** A sequence of instructions for a computational device, written in an appropriate programming language, for implementing an algorithm that manipulates appropriate data representations for solving a problem.
- **Programming language** A formal language for representing statements, or commands, and data values used in a program. A programming language has a precise syntax that defines the valid ways for combining the symbols used to denote variables and data values. Examples used on schools include Scratch, Python and SmallBASIC.
- **Search technologies** Algorithms used by applications known as search engines to trawl the internet for digital content matching search terms given by a user. Results are normally presented in the form of links to relevant content.
- **Sequence** A number of program statements, to be executed one after another.
- **Software** The programs that enable computers to undertake specific functions.
- **Variable** A symbol in the code for a program that represents a data value or data object that can be changed during the course of computation.