# *Computer Programming for Key Stage 3*

## *Sample teaching unit for ICT - computer programming*

*The green text represents editorial notes and comments - please respond.*

*I am happy to accept comments; suggestions and resources off-list to J.Woollard@southampton.ac.uk and then incorporate before sending back to the group for consideration.*

*Please consider the following sections and suggest your words/text/alternatives:*

*Read the "rationale for computer programming" and modify the text so that you agree with it.*

*Look at the generic homework tasks and decide if you could draft up some exercises.*

*Glossary - do we need a definition/description of words to help teachers?*

*Appendix 1 - we need some good references to materials - no need to re-invent the wheel.*

*The three scenarios - I'd rather not have the gaming scenario - I'd like 3 focussing on coding but we need 3 teachers' work to celebrate - do you have ideas.*

*Read the Attainment Targets (Appendix 2) and be inspired to create computer programming activities that can enable pupils to attain at all levels.*

This document illustrates how the yearly objectives from the Framework for teaching ICT capability: Years 7, 8 and 9 can be grouped together and taught in a way that promotes and utilises knowledge and understanding of computing. Programming is a core activity of computing because it enables the user to access and release the potential of the computer they are using. Computer programming can be likened to playing chess - although there is a relatively small set of simple rules, it is the strategic and sustained application of those rules that can create interesting games between children or intellectual fights between grand masters. The same with programming, the first applications of the rules can produce the interesting results, fun play on graphics, numbers or words. But, there is no boundary preventing the learner moving all the way to being the grand master of computer programs. Once you can do it, the sky's the limit over what you can make computers do.

The following texts are direct quotes from the 2008 revision of the National Curriculum for ICT.

> *Capability - using a range of ICT tools in a purposeful way to tackle questions, solve problems and create ideas and solutions of value.*
>
> *Developing ideas - pupils should be able to test predictions and discover patterns and relationships, exploring, evaluating and developing models by changing their rules and values.*
>
> *Use ICT to make things happen by planning, testing and modifying a sequence of instructions, recognising where a group of instructions needs repeating, and automating frequently used processes by constructing efficient procedures that are fit for purpose.*
>
> *Pupils should be able to review, modify and evaluate work as it progresses, reflecting critically and using feedback.*
>
> *Scope the information flow: Represent a system and identify all its parts, including inputs, outputs and the processes used. (Processes could include manipulating data or information.)*
>
> *Developing an ICT-based model to meet particular needs: this should involve testing predictions and discovering relationships, exploring, evaluating and developing models by changing their rules and values.*

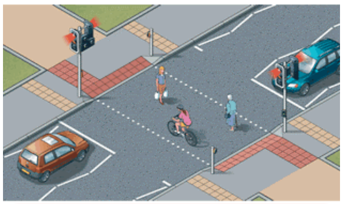
The aim of ***Computing for the Next Generation*** and this document is to promote the principles of "computing" into the key stage 3 curriculum. It also celebrates, through using them as example scenarios, the work of teachers currently engaged in teaching programming to pupils.

## *Types of computer programming*

A program is just a plan-of-action a machine can follow and everything a computer does is based upon a plan-of-action. There are different ways in which computer programming can be approached, there are different forms that computer programs can take and there are different resources/software available to create programs in particular forms.

In this basic example pupils are introduced to the concept of sequencing instructions that will be followed by the computer to control the lights at a pedestrian/cyclist crossing (Toucan).

```
IF INPUT 1 ON THEN
SWITCH OFF 3 [traffic green light]
SWITCH ON 2  [traffic amber light]
WAIT 3
SWITCH OFF 2 [traffic amber light]
SWITCH ON 1  [traffic red light]
WAIT 2
SWITCH OFF 4 [pedestrian/cyclist red light]
SWITCH ON 5  [pedestrian/cyclist green light]
WAIT 20
SWITCH OFF 5 [pedestrian/cyclist green light]
SWITCH ON  4 [pedestrian/cyclist red light]
WAIT 5
SWITCH ON 2  [traffic amber light]
WAIT 5
SWITCH OFF 1 [traffic red light]
SWITCH OFF 2 [traffic amber light]
SWITCH ON 3  [traffic green light]
ENDIF
```



Rule 25: Toucan crossings can be used by both cyclists and pedestrians

*Extend this example later to show procedures, setting initial states/values and concept of variables*

A good supporting resource is the Highway Code online
http://www.direct.gov.uk/en/TravelAndTransport/Highwaycode

*A point for discussion, development and writing on February 5th*

## *Rationale for computer programming in the curriculum*

Computer programming is more than the paid employment of computer programmers.

Computer programming is carried out by many people and is a hobby, pastime, leisure pursuit, interest, diversion, relaxation… For our pupils, it could be a way of enabling them to "enjoy and achieve" - an aim of the Every Child Matters agenda http://www.everychildmatters.gov.uk/aims

"*In some senses, computer programming itself is one of the best computer games of all. In the 'computer programming game', there are obvious goals and its easy to generate more. The 'player' gets frequent performance feedback (that is, in fact, often tantalizingly misleading about the nearness of the goal). The game can be played at many different difficulty levels, and there are many levels of goals available, both in terms of the finished product (whether it works, how fast it works, how much space it requires, etc.) and in terms of the process of reaching it (how long it takes to program, etc.). Self-esteem is crucially involved in the game, and there is probably the occasional emotional or fantasy aspects involved in controlling so completely, yet often so ineffectively, the behaviour of this responsive entity. Finally the process of debugging a program is perhaps unmatched in its ability to raise expectations about how the program will work, only to have the expectations surprisingly disappointed in ways that reveal the true underlying structure of the program*" (Malone, 1980).

Computer programming is also a vocational pursuit and may enable our pupils to "*achieve economic well-being*", another aim of Every Child Matters. Pupils discovering their proficiency in handling syntax, algorithm, logic and analysis may find they can enter an industry in which those skills are highly valued.

However, this teaching resource is designed to bring computer programming to every pupils' experience because it contributes in a powerful way to their ability to learn, conceptualise and understand. Computer programming exercises skills that are valuable in other aspects of learning, work and leisure. It also gives an insight into why computers behave as they do and therefore puts an understanding into the ICT curriculum.

*Is Jack's ladder properly represented?*

| | | |
|---|---|---|
| accuracy of expression | algorithm | visual representation |
| analysis | data structure | logic |

### *Accuracy of expression*

Through computer programming we can insist upon and, importantly, demonstrate the need for accuracy and precision in what we do.

At the ***character level*** it is akin to spelling in the English curriculum - color and colour are significantly different.

To change the colour of text to red use `<font color="#FF0000">` [HTML]

*Is HTML a programming language - it is a scripting language, is it the lack of iteration that stops it being a programming language?*

But the accuracy at character level is more than just an Americanisation of spelling; it is also "paying attention to detail" and realising that spaces, punctuation marks and the case of letters are important. In many areas, computer programming is case sensitive.

At the ***syntax level*** the pupils become aware of the structures of instructions with operands and operators and the need to match each with the other in much the same way as there is an object-verb relationship. For example, `FORWARD 10, WAIT 20, REPEAT 5, PRINT "hello world"`.

Structures such as the `IF THEN ELSE ENDIF` emphasise the importance of accuracy at a level higher than the individual character. In science, pupils would be expected to know the symbols of reactants in an experiment and use the correct syntax when representing them, for example,
`H2SO4 + 2NaHCO3 → Na2SO4 + 2H2O + 2CO2↑`

At the ***instruction level*** the pupils have to be aware of structure in the same way as the grammar of an English sentence has rules of structure. All sentences begin with a capital letter and end with a full stop, exclamation or question mark. They follow the rules of grammar and punctuation. Each sentence has a subject and a verb. In the same way, computer instructions have a precise structure with common rules called syntax. For example, in many languages the end of a line of code is indicated by a semi-colon.

```
    if (aName == bName) {System.out.println('== the same')};
```

At the ***modules, procedure or program level*** then the concept of wholeness is introduced. The computer program is complete within itself and usable. When pupils are asked in history to write about the reason for the rise of nationalism in the 1930s, they are expected to present their ideas as a sequence of connected statements that follow each other logically to build a sustained argument, frequently as a paragraph of text. In the same way, pupils develop the skills of coherent thought through sequencing the instructions, beginning by setting the context, carrying out the operation and concluding in a formal manner. Writing a complete module, procedure or program is like writing a formal essay, poster, recipe or invitation. The product has a wholesomeness or completeness.

For example, a simple program in BASIC to print out a “times table”,

```
    10 A=7
    20 M=12
    30 FOR K = 1 TO M
    40 P=K*A
    50 PRINT A;” times “; K;” equals “; P
    60 NEXT K
    70 END
```

Lines 10 and 20 set the context; 30 to 60 carry out the repeated operation to print 1 times 7 equals 7, 2 times 7 equals 14, etc. and the final line formally ends the program releasing the computer to do other things.

### *Algorithm*

Algorithm can be considered a sequence of instructions, a finite set of commands or a method of working. Algorithm can be considered synonymous with program but algorithm encompasses the whole domain of carrying out instructions in a predefined and accurate way. Algorithm is to computer program as writing is to story. It is not limited to programming. However, through computer programming, pupils can gain a better understanding of the value of predefined sequences of action to more efficiently and effectively achieve an outcome.

Two initial definitions for pupils are:

***an algorithm is a sequence of instructions to be carried out until an end point is reached;***

***algorithm is the rules, conditions or sequence by which the computer or we tackle a problem or situation.***

Other keywords to be use when discussing algorithm are: these words will need clarifying in prose

steps, instructions, commands

sequence, flow,

decisions, branches, jumps, if then, conditional, if then else, true false

repeat, until, condition, iteration

Algorithm takes many forms. It can be the rules by which you drive a car. It can be the way in which you eat from a buffet. It can be the way in which you carry out a science procedure.

“You are approaching a traffic queue: which lane do you take? Always going to the shortest line is a greedy algorithm. You might consider the shortest queue but always err to the right because you think that the fastest drivers are there. You may rely upon knowledge of the road and queues and make different decisions in different places. In the same way, we program the computers to obey a set of predetermined rules - the algorithm.

### *Visual representation sections to be written*

describe visual representation

flow diagram

dry run chart

coding sheet

### *Analysis*

describe analysis

what are the principles of analysis of a situation to prepare for computer programming?

### *Data structures*

describe data structure

science classification keys binary tree

indexing/sequencing of shopping magazines

car registration number plates

### *Logic*

describe logic

## *The research evidence*

*I am musing on the benefit of computer programming for thinking skills to which you allude but perhaps needs more prominence. Any research in this area? Simon H.*

"*If you can use technology to make things you can make a lot more interesting things. And you can learn a lot more by making them. We are entering a digital world where knowing about digital technology is as important as reading and writing. So learning about computers is essential for our students' futures BUT the most important purpose is using them NOW to learn about everything else*" (Papert, 1999).

*the Papert quote I got from:http://www.stager.org/articles/thecaseforcomputing.html, in the Bibliography he references 2 Papert papers from 1999, it is not clear which is the relevant one!*

*http://books.google.com/books?id=6KKpnpLevg4C&pg=PA359&lpg=PA359&dq=%22computer+programming%22+develop+%22thinking+skills%22&source=bl&ots=lSiUw05Gl-&sig=dO0AiUTl2pthiPiKLVDDk7P9EsI&hl=en&sa=X&oi=book_result&resnum=1&ct=result#PPA357,M1*

Thinking : the second international conference / edited by D.N. Perkins, Jack Lochhead, John Bishop. Papers presented at the 1984 International Conference on Thinking, held at the Harvard Graduate School of Education, Cambridge, *Mass. I hope to have the University of Southampton copy before February 5th*

A Study of the Development of Programming Ability and Thinking Skills in High School Students. by Kurland, D. Midian; et al: Journal of Educational Computing Research, v2 n4 p429-58 1986 Abstract: A study of high school students learning computer programming was conducted to determine the impact of programming on particular mathematical and reasoning abilities, the cognitive skills or abilities that best predict programming ability, and what students actually understand about programming after two years of study. (MBR) *I've made an interlibrary loan request, John.*

*Just found it! Document available from* http://hal.archives-ouvertes.fr/docs/00/19/05/39/PDF/A29_Kurland_etal_86.pdf

The influence of pupils' thinking skills: implementing computer programming instruction (LOGO) in elementary school in Taiwan Shun-der Su* Abstract A quasi-experimental research design was used to investigate the changes in pupils' mathematical problem-solving ability, logical and spatial reasoning ability, and the attitude toward operating computer within the computer programming language LOGO instruction for sixth grade students in Taiwan. Available at http://www.npue.edu.tw/adm/research/%BE%C7%B3%F8/13/9.pdf

*More references that need tracking down and using/eliminating.*

*Bishop, D. (1988.) BASIC Training. inCider, 6, 6, pp. 52-58.*

*Clements, D. (1986.) Effects of Logo and CAI environments on cognition and creativity. Journal of Educational Psychology, 78, 4, pp. 309-318.*

*Cumming, G. (1990.) K-Log: A tool for thinking, across the curriculum. Education, 110, 4, pp. 418-423.*

*Jonassen, D. (1996). Computers in the Classroom: Mindtools for Critical Thinking. Englewood Cliffs, N.J.: Prentice-Hall, Inc. A Simon & Schuster Company.*

*McCoy, L. (1990.) Literature relating critical skills for problem solving in mathematics and in computer programming. School Science & Mathematics, 90, 1, pp. 48-60.*

*Nixon, P. (1993.) Project report: The use of PILOT to teach information technology concepts. Computers & the Humanities, 27, 4, pp. 285-289.*

*Silvern, S. (1988.) Creativity through play with Logo. Childhood Education, 64, 4, pp. 220-224.*

*Tuovinen, J. & Hill, D. (1992.) Towards better strategies for thinking and programming. School Science & Mathematics, 92, 4, pp. 206-211.*

*Yelland, N. (1995.) Encouraging young children's thinking skills with Logo. Childhood Education, 71, 3, pp. 152-155.*

*A point for discussion, development and writing on February 5th*

## Scenarios

The way in which computer programming can be introduced in the classroom is illustrated through these scenarios. They are not comprehensive in nature but illustrative of good and successful practice in UK schools. Each scenario is described by the opportunities to support particular forms of computer programming. The outcomes of the pupils' activities are described in terms of the attainment targets at levels 4, 6, 8 and exceptional performance. The alternative resources that can be used are also described.

We need volunteers to send in what they are doing so that they can be represented in the style of a scenario - see below.

### About the ICT sample teaching units for Key Stage 3

The following section is adopted directly from a sample teaching unit. "*The ICT Framework recommends that schools offer one hour each week, or 38 hours per year, for discrete ICT lessons. The sample teaching units for a year, if taught without amendment, need less teaching time than 38 hours. This leaves time for lessons of your own design at suitable points*" (DfES, 2002a).

*'DFES 2002a'. Can you provide me with a link to these documents? It would be useful to have them to hand when we meet in Feb.*

*This document has been further updated to reflect 2008 changes l the National Curriculum for ICT.*

*19The National Strategies | Secondary*

*The Framework for secondary ICT: overview and learning objectives*

*This unit is similar in intention to one of a series illustrating how objectives from the Framework for teaching ICT capability: Years 7, 8 and 9 can be taught (DfES, 2002b).*

http://www.standards.dfes.gov.uk/secondary/keystage3/respub/ictframework/foreword

There is no requirement to use the units but they are used in many schools. They normally contain sample lesson plans that you can amend to suit your local circumstances and the needs of your pupils. This unit is different in that it present 3 alternative scenarios that can be taught with one of several different resources.

The units contain outline plans for lessons of 60 minutes. Each activity in the lessons has a guide time. This will help you to fit activities into lessons that are longer or shorter than 60 minutes.

### About sample teaching computer programming unit

This unit helps you to introduce some of the ICT Framework objectives for Year 7 in the theme 'Developing ideas and making things happen'. It focuses upon the National Curriculum (QCA, 2007) key processes of developing ideas, communicating information and evaluating. In particular, the activities support curriculum requirements that pupils should be able to:

2.2e use ICT to make things happen by planning, testing and modifying a sequence of instructions, recognising where a group of instructions needs repeating, and automating frequently used processes by constructing efficient procedures that are fit for purpose;

2.3c use technical terms appropriately and correctly;

2.4a review, modify and evaluate work as it progresses, reflecting critically and using feedback.

These statements are taken from the "new" National Curriculum introduced in September 2008 to Year 7 pupils.

Reflecting critically could include self-review, peer evaluation and user or audience feedback. Pupils should judge both the quality of their work and how effectively they have used ICT. In computer programming this can be reflected in the minimum use of code, the fastest processing time or the shortest development time.

Aspects of control and monitoring are taught in both science and design and technology. You might find it helpful to ask these departments what they have covered with pupils before you teach this unit. You could then refer to the work pupils have done in these other subjects at appropriate points in the lessons. For example, pupils may have created sequences of instructions in control software.

## *Keywords or Glossary shall we write a glossary?*

Keywords related to computer programming

automate/automatic, control, control loop, flow chart, input, input device, label, logo, model, output, output device, procedure, process, program, random number, repeat, repeated process, robot, rule, sequence of instructions, store, switch, subtask, system, template, variable (Drawn from Key Stage 3 Strategy materials, 2005)

steps, instructions, commands, sequence, flow, decisions, branches, jumps, if then, conditional, if then else, true false, repeat, until, condition, iteration, procedure, subroutine, call.

## *Is this a thinking/planning/writing activity for February?*

## *Homework tasks (generic)*

***A point for discussion, development and writing on February 5th***

6 tasks to highlight the 6 aspects of computer programming

“Well organised homework can play a vital role in raising standards of achievement” (DfES, 1999)

http://www.teachernet.gov.uk/docbank/index.cfm?id=12761

*Do a search for “homework” on the TeacherNet website* http://www.teachernet.gov.uk

*each of these needs expanding to give clarity to the task and a handout produced as a RTF file*

*we need to present each of these in two ways - one that would be for a class working at about level 5 enabling level 3 pupils to do something but level 6 to be challenged - the more difficult sheet would be for a class working at level 6 challenging the most able.*

*there needs to be a computer dependent task and a computer independent task for each.*

***Accuracy of expression*** - data entry form for name, date of birth, telephone number, post code - include issues of data security and how the teacher disposes of the physical paper.

***Algorithm*** exact sequencing of events, choose two from the list and describe accurately as a sequence of steps. In class afterwards the teacher goes through each event highlighting the key factors and the important sequence points and introduces iteration and conditional.

***Visual representation***. Pupils are given a revision/information sheet with the different symbols and then asked to label a complete diagram, complete a partial diagram and create a new diagram for a common event.

***Analysis*** Pupils are issued with a computer program and they have to complete a dry-run. They then have to consider which lines to change to carry out a different process.

***Data structures*** Activity sheet with partially completed tables for completion by pupils based on:

a spreadsheet-like grid of records and fields of well known musicians;

several truth tables;

two Venn diagrams;

statements that are answered true/false using radio buttons;

sorting a set of people into alphabetical and numeric order.

***Logic*** further development of Venn diagram, truth tables,

## *Scenario 1*

Please note: the following text is an example of how the activities of a scenario could be presented as a sequence of 6 lessons.

| Lesson Number | Learning objective<br>We are learning to<br>WALT | Learning outcomes<br>What I am looking for<br>WILF | Rationale<br>This is because<br>TIB | Prerequisite skills of pupil and teacher, resources and | NC, Strategy and QCA references |
|---|---|---|---|---|---|
| 1/6 | That a computer programme is a series of instructions programmed in to automate a sequence of events<br>or<br>that a computer programme is a set of statements that can be executed or interrogated | That you can recognise the correct sequence of a  set of traffic lights<br>That you can arrange a sequence of instructions for a set of traffic lights into the correct order<br>Start to recognise the advantages of ICT systems over manual systems | Importance of computer programming to make computers do what we wish them to do. | Keywords display<br>Teacher's dictionary. | |

Supporting text…

*To be discussed, developed and written on February 5th.*

*WALT We are learning to… should be the skills, knowledge and understanding of the lesson (perhaps also "attitudes");*

*WILF What I'm looking for… the assessment for learning or assessment for teaching statements*

*TIB This is because… the rationale for teaching, say,  how to copy using absolute cell referencing, the essence of ICT capability.*

*Scenario 2 - a thinking machine*

***Introductory lesson:***

talk about and show videos of artificial intelligence computers;

explore tic-tac-toe;

discussion - identify some real-world entertainments;

talk to Eliza;

discussion - identify some real-world applications;

demonstrate Prolog;

extension work - consider a scenario.

***Resources and advice:***

http://www.cs4fn.org/programming/noughtscrosses

first play tic-tac-toe on paper, "can anyone guarantee to always win?", "what is your strategy?"

the challenge is to stop the computer winning by designing the right sequence of responses.

| Lesson Number | Learning objective<br>We are learning to<br>WALT | Learning outcomes<br>What I am looking for<br>WILF | Rationale<br>This is because<br>TIB | Prerequisite skills of pupil and teacher, resources and | NC, Strategy and QCA references |
|---|---|---|---|---|---|
| 1/6 | That a computer programme is a series of instructions programmed in to automate a sequence of events<br>or<br>that a computer programme is a set of statements that can be executed or interrogated | That you can choose and justify your choice of order for instructions<br>That you can create an artificial intelligence program that rarely loses at tic-tac-toe.<br>Start to recognise the potential of ICT systems to think | Importance of computer programming to make computers do what we wish them to do and eventually "think" through and resolve problems that we cannot resolve. | Keywords display<br>AI artificial intelligence<br>declarative<br>sequence<br>order<br>Teacher's dictionary. | |

Supporting text…

*To be discussed, developed and written on February 5th.*

## *Scenario 3 - a game business*

### *Context*

This scheme of work was created by Liz Crane for Oaklands Roman Catholic School, Waterlooville, Hampshire. It combines the game authoring activities and business related skills and knowledge activities to form a 15-lesson structure that enables pupils to experience many aspects of ICT and focus upon those elements that interest them the most. The intention is to offer level 5 opportunities for all with extension opportunities for some.

Scheme of work is called GameBusiness.doc

### *Aims:*

Creating a game

Create a game that has a sequence, loop and different things happen depending on changes in variables. Make game more efficient through adding a loop, sub routine

Creating a cash flow

Create a spreadsheet model. Consider the layout and content of the model. Use a set of rules to predict values and solve problems 'what if questions'. Students can predict some of the effect of these changes.

Creating a Database

Students can, with guidance prepare a flat database, using a data capture as a planning tool and test the data set. Plan a data form. Create a data entry form.

Creating a Web page and an online form

Create a set of criteria to judge their own work and that of their peers. Design website to meet the needs of the audience and for a specific purpose. Structure, refine,and present information in different ways, including using email.

*To be discussed, developed and written on February 5th.*

*WALT We are learning to… should be the skills, knowledge and understanding of the lesson (perhaps also "attitudes");*

*WILF What I'm looking for… the assessment for learning or assessment for teaching statements*

*TIB This is because… the rationale for teaching, say, how to copy using absolute cell referencing, the essence of ICT capability.*

## *References*

DfES (2002b) *Key Stage 3 National Strategy Framework for teaching ICT capability: Years 7, 8 and 9* London, UK: Department for Education and Skills

Malone, TW (1980) What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games. Paper presented at the *Association for Computing Machinery Symposium on Small and Personal Computer Systems*, Pal Alto, California.

(Papert, 1999). *Need to find reference*

## *Appendix 1*
## *Resources that support computer programming*

*We need teacher reviews of these resources and references to reviews already carried out.*

Alice http://alice.org

Flash

Gamemaker

Greenfoot

Scratch

We need more items in this list.

## *Appendix 2*
## *Level descriptors for the National Curriculum ICT*

*We need teacher reviews of these resources and references to reviews already carried out.*

### *Level 4*

Pupils combine and refine different forms of information from various sources. Pupils understand the need for care in framing questions when collecting, finding and interrogating information. They interpret their findings, question plausibility and recognise that poor-quality information leads to unreliable results. They use ICT to present information in different forms and show they are aware of the intended audience and the need for quality in their presentations. They exchange information and ideas with others in a variety of ways, including using digital communication. They understand the risks associated with communicating digitally, including the security of personal information. **They plan and test sequences of instructions.** They use ICT-based models and simulations to explore patterns and relationships, and make predictions about the consequences of their decisions. They use ICT to organise, store and retrieve information. They compare their use of ICT with other methods and with its use outside school.

### *Level 5*

Pupils combine ICT tools within the overall structure of an ICT solution. They select the information they need for different purposes, check its accuracy and organise it in a form suitable for processing. They use ICT to structure, refine and present information in different forms and styles for specific purposes and audiences. They exchange information and ideas with others in a variety of ways, including using digital communications. **They create sequences of instructions and understand the need to be precise when framing and sequencing instructions.** They explore the effects of changing the variables in an ICT-based model. They use ICT to organise, store and retrieve information using logical and appropriate structures. They use ICT safely and responsibly. They discuss their knowledge and experience of using ICT and their observations of its use outside school. They assess the use of ICT in their work and are able to reflect critically in order to make improvements in subsequent work. They use appropriate evaluation criteria to critically evaluate the fitness for purpose of their work as it progresses.

### *Level 6*

Pupils plan and design ICT-based solutions to meet a specific purpose and audience, demonstrating increased integration and efficiency in their use of ICT tools. They develop and refine their work to enhance its quality, using a greater range and complexity of information. Where necessary, they use complex lines of enquiry to test hypotheses. They present their ideas in a variety of ways and show a clear sense of audience. **They develop, try out and refine sequences of instructions and show efficiency in framing these instructions, using sub-routines where appropriate.** They use ICT-based models to make predictions and vary the rules within the models. They assess the validity of these models by comparing their behaviour with information from other sources. They plan and review their work, creating a logically structured portfolio of digital evidence of their learning. They discuss the impact of ICT on society.

### *Level 7*

Pupils design and implement systems. They are able to scope the information flow required to develop an information system. They combine information from a variety of ICT-based and other sources for presentation to different audiences. They identify the advantages and limitations of different information-handling applications. They select and use information to develop systems suited to work in a variety of contexts, translating enquiries expressed in ordinary language into the form required by the system. **They develop, test and refine sequences of instructions as part of an ICT system to solve problems.** They design ICT-based models and procedures with variables to meet particular needs. They consider the benefits and limitations of ICT tools and information sources and of the results they produce, and they use these results to inform future judgements about the quality of their work. They make use of audience and user feedback to refine and enhance their ICT solutions. They take part in informed discussions about the use of ICT and its impact on society.

### *Level 8*

Pupils independently select appropriate information sources and ICT tools for specific tasks, taking into account ease of use and suitability. They design successful ways to collect and prepare information for processing. **They design and implement systems for others to use.** They take part in informed discussions about the social, economic, ethical and moral issues raised by ICT.

***Exceptional performance***

Pupils evaluate software packages and ICT-based models, analysing the situations for which they were developed and assessing their efficiency, ease of use and appropriateness. **They suggest refinements to existing systems and design, implement and document systems for others to use, predicting some of the consequences that could arise from the use of such systems.** When discussing their own and others' use of ICT, they use their knowledge and experience of information systems to inform their views on the social, economic, political, legal, ethical and moral issues raised by ICT.