**Computing Scheme of Work 2014 –Year 1**

|  |  |  |
| --- | --- | --- |
| **TERM** | **COMPUTING OBJECTIVES** | **ACTIVITIES & RESOURCES** |
| Computer Science – Technology Around Us | Recognise common uses of information  technology beyond school  Understand what algorithms are;  Use logical reasoning to predict the behaviour of simple programs | **Somerset Resource Technology in Our Lives**  <https://slp.somerset.gov.uk/cypd/elim/somersetict/Computing_Curriculum_Primary/Technology%20in%20lives%20FS%20KS1.pdf> |
| **Nina and the Neurons-Go Digital**  [**http://www.bbc.co.uk/cbeebies/nina-and-the-neurons/**](http://www.bbc.co.uk/cbeebies/nina-and-the-neurons/)  This CBeebies show has recently shown episodes on how a mobile phone works and how a computer works. The episodes give ideas as to how you could deliver a lesson to your class in a practical and fun way. You may find the clips on You Tube.  <http://www.bbc.co.uk/cbeebies/games/nina-go-digital-game> |
| **How a supermarket works**  <http://www.code-it.co.uk/csplanning.html>  The children that they are going to model what really happens inside a supermarket. Not just what they see happening but also what happens behind the scenes on the tills and computers. |
| **KEY SKILLS**  Identify uses of technology in the classroom, at home and in the local area.  To know what an algorithm is.  To give instructions in the correct sequence.  <https://slp.somerset.gov.uk/cypd/elim/somersetict/Site%20Pages/Computing%20Curriculum%20Primary/Primary_Computing_home.aspx?PageView=Shared>  At KS1 there are many opportunities for pupils to understand algorithms and to create their own algorithms. For example, counting to 100 and learning the alphabet help with sequencing. Games like ‘Simon says’ help pupils to develop their use of instructions. Pupils can come up with their own sequences of instructions, for example how to get dressed or clean their teeth, and sorting sequences of instructions for simple tasks into their correct order.  As they break down the larger tasks into smaller instructions, they develop their understanding of using decomposition to solve a problem. The algorithms pupils create can often be implemented using programmable toys or with human robots.  Sequencing life cycles and stories. For example, asking children to sequence butterfly or frog life cycles and different food chains or sequencing the order of events in stories. | **Introducing Algorithms**  <http://www.computingatschool.org.uk/index.php?id=introducing-algorithms-to-children-in-reception-class> |
| **Computers All Around Us**  <http://www.thelifecloud.net/schools/WestHoveInfants/spaces/Classes/resources/ICT/Commonuse.page> |
| **SHERSTON SOFTWARE**  **Charlie Chimp’s Modelling Party -** Charlie Chimp is planning his annual party and needs your pupils to help him with his preparations. Are they up to the challenge? Will they be able to use their modelling and problem-solving skills to provide Charlie and his friends with the best Big Modelling Party yet? Sherston software.  **Little Brown Bear on the Farm** - Sherston software that develops essential ICT skills for early learners as they explore the fun-packed farmyard. With full narration, simple, bright graphics, enchanting musical introduction sequences and an intuitive design |
| **Songs and Games**  Musical algorithms (Hokey Cokey): Can pupils write the algorithm, sequence picture cards and create new dance routines? Learn dance routines that have actions, instruct a partner and teach them the dance. Just Dance Kids 2 - Five Little Monkeys Use Clip converter to download dances and action routines from You Tube  <http://www.clipconverter.cc/>  Play Simon Says, sing and perform songs with actions, Follow the Leader  Teach the class the Alligator Algorithm chant and have a class alligator hat. |
| **2Simple – 2 Do It Yourself**  Create your own interactive Flash resources, activities, games, puzzles, quizzes. With this software teachers and primary school children can create cross-curricular, personalised resources and use them on whiteboards, websites and even on Learning Platforms. There are plenty of opportunities for meaningful learning as children plan, design, create, publish and play.  Draw a teddy bear & dress him, create your own interactive screens, make jigsaws, sorting & grouping activities, picture quizzes & even create your own platform games. |
| **Sequencing games and activities**  [**http://www.turtlediary.com/kindergarten-games/ela-games/picture-sequencing.html**](http://www.turtlediary.com/kindergarten-games/ela-games/picture-sequencing.html)  [**http://www.superduperinc.com/handouts/pdf/167%20Sequencing.pdf**](http://www.superduperinc.com/handouts/pdf/167%20Sequencing.pdf)  **TES iboard – sequencing and sorting games**  [**http://www.iboard.co.uk/teacher/jlisaw8/1**](http://www.iboard.co.uk/teacher/jlisaw8/1) |

**Computing Scheme of Work 2014 –Year 2**

|  |  |  |
| --- | --- | --- |
| **TERM** | **COMPUTING OBJECTIVES** | **ACTIVITIES & RESOURCES** |
| Computer Science - Algorithms | Recognise common uses of information  technology beyond school  Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.  Use logical reasoning to predict the behaviour of simple programs | **Somerset Resource Technology in Our Lives**  <https://slp.somerset.gov.uk/cypd/elim/somersetict/Computing_Curriculum_Primary/Technology%20in%20lives%20FS%20KS1.pdf> |
| **Barefoot Computing Algorithm Lesson Plans**  Crazy Characters  Sharing Sweets  Spelling Rules  **http://barefootcas.org.uk/** |
| **Human Crane Algorithm**  <http://www.code-it.co.uk/csplanning.html>  Pupils create and test crane algorithms that move blocks from one bowl to another.  The author of this activity has a Power Point with some nice ideas for lessons on. Search KS1 Unplugged. |
| **KEY SKILLS**  <https://slp.somerset.gov.uk/cypd/elim/somersetict/Site%20Pages/Computing%20Curriculum%20Primary/Primary_Computing_home.aspx?PageView=Shared>  At KS1 there are many opportunities for pupils to understand algorithms and to create their own algorithms. For example, counting to 100 and learning the alphabet help with sequencing. Games like ‘Simon says’ help pupils to develop their use of instructions. Pupils can come up with their own sequences of instructions, for example how to get dressed or clean their teeth, and sorting sequences of instructions for simple tasks into their correct order.  As they break down the larger tasks into smaller instructions, they develop their understanding of using decomposition to solve a problem. The algorithms pupils create can often be implemented using programmable toys or with human robots.  Sequencing life cycles and stories. For example, asking children to sequence butterfly or frog life cycles and different food chains or sequencing the order of events in stories. | **Thinking for Myself**  <http://games.thinkingmyself.com/> |
| **iPad Apps**  Cato's hike, A.L.E.X, Move the turtle, Bee-Bot and Daisy Dino, Hopscotch |
| **Bubble Sort is** an example of an algorithm that a computer uses to sort information. What they need to know is that a computer can only compare two values (objects) that are next to each other, and swap them if necessary before moving onto the next two values. Here are some examples of Bubble Sorts with lots of links to numeracy and science:   * Use a balance scale to sort weights (use coins in an envelope for example). * Fill plastic bottles with e.g. sand, pasta, beads and sort loudest to quietest. * Sort coins in order of value. * Roll up different lengths of coloured paper and sort longest to shortest. * Put different textured materials in envelopes and sort roughest to smoothest. * Use toy animals and sort from least likely to most likely to win in a fight (or you could do a food chain).   <http://sencomputing.wikispaces.com/Computer+Science+KS1>  Teach your class the Bubble Sort dance to sort numbers  <http://bit.ly/tp-bubble> |
| **Rommy Robot**  <http://www.sandaigprimary.co.uk/fun/rommy_robot.html>  Rommy Robot is a screen programmable robot. More information from:-  <http://www.littlefishsw.co.uk/software/rommy/games.html> |
| **Musical Algorithms**  Musical algorithms - Can pupils write the algorithm, sequence picture cards and create new dance routines? You Tube - Just Dance Kids 2 - The Gummy Bear Song Use Clip converter to download dances and action routines from You Tube  <http://www.clipconverter.cc/>  Create and record their own verses to the Alligator Algorithm chant  Algorithms make you happy - You Tube video song clip based on Pharrell Williams - Happy |
| **Are computers clever?**  CAS resource – on the DVD  Introduction to algorithms – making the tea – Year 2 CAS resource (See DVD)  TESiboard  Programming games –  <http://www.iboard.co.uk/teacher/jlisaw8/2>  Mole Maze - <http://www.iboard.co.uk/search?keywords=maze> |
| **BBC Computer Science video clips**  [**http://www.bbc.co.uk/education/topics/z3tbwmn**](http://www.bbc.co.uk/education/topics/z3tbwmn) |

**Computing Scheme of Work 2014 –Year 3**

|  |  |  |
| --- | --- | --- |
| **TERM** | **COMPUTING OBJECTIVES** | **ACTIVITIES & RESOURCES** |
| Computer Science – Computational Thinking & Exploring Simulations | NC 2014 “A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world.”  Design, write and debug programs that accomplish specific goals.  Solve problems by decomposing them into smaller parts.  Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. | **Barefoot Computing – introduction to computational thinking concepts for teachers (Logic, algorithms, decomposition, patterns, abstraction, evaluation)**  [**http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/**](http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/) |
| **Logic and patterns – Barefoot Computing logical number sequences lesson plan**  In this activity pupils explain the rule for a number sequence and predict which number(s) comes next. In doing so, they extend their knowledge of simple rule based algorithms. They also use logical reasoning as they work out and explain their algorithms. The lesson plan given is for Year 5 but this can easily be adapted for use with Year 3 children by using mathematical pattern and guess my rule activities. For more challenging number pattern work look at the Nrich w  ebsite**<http://nrich.maths.org/frontpage>** |
| **KEY SKILLS**   |  | | --- | | **Logic**  I can use logical reasoning to explain  how simple algorithms work and to  detect and correct errors.  **Pattern Recognition**  Is this similar to a problem I’ve already solved?  How is it different?  How are the parts of the problem  connected?  **Decomposition**  Can I break this problem up into smaller parts?  Can I explain the different parts of this problem and solution? | | **Algorithm Design**  What do I need to think about to make  this happen?  What are the steps I will need to do  to solve this problem?  **Abstraction and Generalisation**  Which is the information I actually need? What don’t I need to know?  Have I made this more complicated  than I need to?  Will this work for other things?  **Evaluation**  I can use evaluation to check solutions  and make sure they do the job they are designed for and that they are fit for  purpose and | | **Algorithms – Jam Sandwich Algorithm**  [**http://www.code-it.co.uk/unplugged/jamsandwich.html**](http://www.code-it.co.uk/unplugged/jamsandwich.html)  Create an **algorithm** (a precise set of instructions or rules to achieve an outcome or solve a problem) to instruct a pretend robot (teacher) to make a jam sandwich. One of the steps towards writing a good algorithm, that can be converted into code, is precision. Precision doesn't come naturally to most humans. Another computational thinking skill is **evaluation** in this case is my algorithm fit for purpose. The robot teacher interpretation of their algorithm helps pupils to evaluate its effectiveness and make changes if needed |
| **Algorithm Dance (Part 1) – Decomposition (break the dance down into smaller parts to learn)**  Japanese Algorithm Dance - discuss the algorithm and perform the dance  [**http://www.thelifecloud.net/schools/WestHoveInfants/resources/ComputingCPD/ComputingCPD.page**](http://www.thelifecloud.net/schools/WestHoveInfants/resources/ComputingCPD/ComputingCPD.page)  [**https://www.youtube.com/watch?v=jdDd7Csl5vc**](https://www.youtube.com/watch?v=jdDd7Csl5vc)  **Video the children doing the dance and use the playback to cover the computational thinking concept ‘Evaluation’**  [**http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/evaluation/**](http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/evaluation/) |
| **Exchange Sort – Another version of the Bubble Sort algorithm taught in Year 2**  <http://www.code-it.co.uk/unplugged/sort/sortoverview.html> |
| **Year 3 Computing assembly**  Deliver an assembly about computing. This covers a brief history of computing, algorithms, the Japanese algorithm dance and a country dancing version of the Bubble sort algorithm. |
| **Abstraction**  I would introduce this concept during this term but I would probably deliver a lesson on it during the Internet topic term when children can use the Barefoot Computing resource – Modelling the Internet  [**http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/abstraction/**](http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/abstraction/)  **Simulations**  [**http://www.simonhaughton.co.uk/2012/05/year-3-exploring-simulations-lesson-2012.html**](http://www.simonhaughton.co.uk/2012/05/year-3-exploring-simulations-lesson-2012.html)  [**http://www.welbourneprimary.com/links/ict/Simulations/Simulations.htm**](http://www.welbourneprimary.com/links/ict/Simulations/Simulations.htm)  [**http://www.geffrye-museum.org.uk/kidszone/garden/**](http://www.geffrye-museum.org.uk/kidszone/garden/)  [**http://www.channel4.com/learning/microsites/R/robots/constructor.html**](http://www.channel4.com/learning/microsites/R/robots/constructor.html) |

**Computing Scheme of Work 2014 –Year 4**

|  |  |  |
| --- | --- | --- |
| **TERM** | **COMPUTING OBJECTIVES** | **ACTIVITIES & RESOURCES** |
| Computer Science – Computer Science Unplugged / Computing Theory | Design, write and debug programs that accomplish specific goals  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 | **Barefoot Computing – introduction to computational thinking concepts for teachers (Logic, algorithms, decomposition, patterns, abstraction, evaluation)** [**http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/**](http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/)  **The Thinkersmith Travelling Circuits**  The website contains simple activities to introduce computational thinking to K-12 children. [**http://travelingcircuits.blogspot.co.uk/**](http://travelingcircuits.blogspot.co.uk/) |
| **Computer Science Unplugged**  [**http://csunplugged.org/activities**](http://csunplugged.org/activities)  Data: Representing Information   1. **Image Representation (Colour by Numbers)** 2. **Error Detection (Card Flip Magic)** 3. **Information Theory (20 Guesses)** 4. **Text Compression (You can Say That Again)** |
| **KEY SKILLS**   |  | | --- | | **Logic**  I can use logical reasoning to explain  how simple algorithms work and to  detect and correct errors.  **Pattern Recognition**  Is this similar to a problem I’ve already solved?  How is it different?  How are the parts of the problem  connected?  **Decomposition**  Can I break this problem up into smaller parts?  Can I explain the different parts of this problem and solution? | | **Algorithm Design**  What do I need to think about to make  this happen?  What are the steps I will need to do  to solve this problem?  **Abstraction and Generalisation**  Which is the information I actually need? What don’t I need to know?  Have I made this more complicated  than I need to?  Will this work for other things?  **Evaluation**  I can use evaluation to check solutions  and make sure they do the job they are designed for and that they are fit for  purpose and | | **Algorithm Dance March (Part 2)**  Japanese Algorithm Dance - discuss the algorithm and perform the dance[**https://www.youtube.com/watch?v=jdDd7Csl5vc**](https://www.youtube.com/watch?v=jdDd7Csl5vc)  **Video the children doing the dance and use the playback to cover the computational thinking concept ‘Evaluation’**  [**http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/evaluation/**](http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/evaluation/) |
| **Introduce flowcharts** [**http://www.cimt.plymouth.ac.uk/projects/mepres/book8/bk8i1/bk8\_1i2.htm**](http://www.cimt.plymouth.ac.uk/projects/mepres/book8/bk8i1/bk8_1i2.htm)  **Playground Games flowcharts**<http://www.code-it.co.uk/unplugged/playgroundgames/playgroundoverview.html>  **Big Bang Theory – Friendship Algorithm** [**https://www.youtube.com/watch?v=jWWOM53Zh20**](https://www.youtube.com/watch?v=jWWOM53Zh20) |
| **Computer Science Unplugged – Algorithms**  **Lightest and Heaviest** Almost any list that comes out of a computer is sorted into some sort of order, and there are many more sorted lists inside computers that the user doesn't see. Many clever algorithms have been devised for putting values into order efficiently.In this activity students compare different algorithms to sort weights in order  [**http://csunplugged.org/sorting-algorithms**](http://csunplugged.org/sorting-algorithms) |
| **Magic Trick – Looking at a sequence of instructions, covers repeat commands**  <http://www.resources.digitalschoolhouse.org.uk/algorithms-a-programs/190-teaching-algorithms>  **Computing theory for 7-11 year olds**  <http://www.simonhaughton.co.uk/computing-theory-for-7-to-11-year-olds.html>  <http://www.simonhaughton.co.uk/2014/06/teaching-computer-science.html>  <http://www.simonhaughton.co.uk/2014/06/the-components-of-a-computer-system.html> |