







Christ Church CE Primary School Computing Assessment 2020-21

Achieving excellence, opening minds, inspiring dreams, creating futures



<p>Key Stage: 2 Year Group: 4</p>	<p>Unit: 4.1 Coding</p>	
<p>Core Curriculum content</p>		
<p>For '<i>working towards</i>', the knowledge/ understanding is:</p>	<ul style="list-style-type: none"> - With support, children can turn a real-life situation into an algorithm for a program that has cause and effect and use their algorithm to write simple programs using 2Code. - They can identify errors within their programs and make logical attempts to fix it. - Pupils attempt to introduce repetition and selection into their code using timers and simple 'if statements'. - Children's use of these structures is experimental; they cannot always predict the outcome accurately or anticipate the structures required when planning their code. - They have a developing idea that a variable can be used to store information in a program, in lesson 5 they can follow the examples but might struggle when applying this with their own ideas. - Children's designs for their programs, show that they are thinking of the structure of a simple program in logical, achievable steps. - Children can make good attempts to 'read' code and predict what will happen in a program which can help them to correct errors in their code. 	
<p>Pupils:</p>		
<p>For '<i>expected</i>' the knowledge/ understanding is:</p>	<ul style="list-style-type: none"> - Children can turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Children's design shows that they are thinking of the required task and how to accomplish this in code. - Children can identify an error within a program that prevents it following the desired algorithm and then fix it (debug), they apply these techniques to their own code. - Children's use of timers to achieve repetition effects are becoming more logical and are integrated into their program designs. - They understand 'if statements' for selection. - Their design demonstrates their understanding of when a coded solution will require repetition. - They can explain the new command 'Repeat Until'. - Children's designs for their programs, show that they are thinking of the structure of a simple program in logical, achievable steps with attention to specific events that initiate specific actions. - Children can read others' code and predict what will happen which helps them to correct errors. - Most children can create programs which accomplish a specific goal utilizing a variety of media such as images, sounds and animation effects. Children can manipulate graphics in the design view of 2Code to achieve the desired look for the program. - Children can interpret the flowcharts used to represent if/else (lesson 2) and create their own when planning their programs. 	
<p>Pupils:</p>		
<p>For '<i>greater depth</i>', the knowledge/ understanding is:</p>	<ul style="list-style-type: none"> - Children are attempting to turn increasingly complex real-life situations into algorithms for a program by deconstructing the situation into manageable parts (lessons 5 and 6). - Children's design shows that they are thinking of the required task and how to accomplish this in code using structures for selection and repetition and variables. - Children make intuitive attempts to debug their own programs. - Pupils realise the constraints of creating programs and intuitively grasp the concepts of selection, repetition and variables. Children like to challenge themselves to combine these with other coding structures to achieve the effects that they design. - Their designs are ambitious but logical and achievable. Children's designs for their programs, show that they are absorbing new knowledge of coding structures such as 'if' statements, repetition and variables to think of their programs in logical, achievable steps. - Children can 'read' others' code and predict what will happen in a program which helps them to correct errors. They can also make good attempts to fix their own bugs as their coding becomes more complex. 	
<p>Pupils:</p>		