hi-impact consultancy

Fazakerley Primary School
Formosa Drive, Liverpool, L10 7LD

Skills Progression: Computer Science

EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
CSF.1	**CS1.1**	**CS2.1**	**CS3.1**	**CS4.1**	**CS5.1**	**CS6.1**
Be able to control a range of ‘toys’ using remote controls.	Be able to say what an 'algorithm' is.	Be able to give control devices instructions that contain numerical data (e.g. move 2 steps etc).	To sequence a list of commands/blocks to produce an output e.g. a light comes on or a robot follows a defined route.	Design, test and amend programs to achieve an intended objective, including controlling an external output.	Can use decomposition when solving problems (break the code/problem into smaller parts).	Be able to use logical operations (not, or, and) to alter and control the outcome of a series of commands.
CSF.2	**CS1.2**	**CS2.2**	**CS3.2**	**CS4.2**	**CS5.2**	**CS6.2**
Be able to explore the language needed to direct a floor robot, to achieve a specific outcome.	Be able to use the appropriate keys or commands to make a virtual or floor robot go forward, backward, left and right.	Can use the repeat command (loops) to program more efficiently.	Is able to use 'repeat' and 'repeat until' loops when appropriate.	Be able to use nested loops to increase the efficiency of a program.	Can explain what happens when a variable changes and can use this within a computer program to manipulate data.	Can use variables efficiently. Be able to create their own variable and use this within a computer program to manipulate data.
CSF.3	**CS1.3**	**CS2.3**	**CS3.3**	**CS4.3**	**CS5.3**	**CS6.3**

©hi-impact Consultancy Ltd 2022

ving Success Toge

Skills Progression: Computer Science

Be able to use step by step commands to program a floor or virtual robot to move.	Be able to program a bot or sprite by giving simple sequences of commands with an immediate outcome.	Can use logical reasoning to predict the outcome of a sequence of instructions and test the sequence, amending if necessary.	Can use simple conditional statements (if and when commands) and understands the importance of time within a program (e.g. using wait), with support.	Can use conditionals including 'if' and 'else'.	Show an understanding of when to use 'while', 'repeat until' and 'forever if' loops to make programs shorter and more efficient and can use them appropriately (understanding the differences between them).	Can demonstrate an understanding of what subroutines (e.g. functions and procedures) are, and be able to create them within a computer program to store and retrieve data.
CSF.4	**CS1.4**	**CS2.4**	**CS3.4**	**CS4.4**	**CS5.4**	**CS6.4**
Begin to use basic symbols to represent and record directional instruction.	Can use basic symbols to record directional instruction and attempt to identify a bug in their code.	Is able to make use of simple events e.g. mouse clicks/tap on screen.	Can make use of an input 'event' within a simple program e.g. when the start button is clicked.	Understands a wider range of 'events' such as sprite interactions and button presses, and can use them within programs.	Can use and change a pre-written function as part of a longer program or sequence.	Be able to use a wider range of events (such as broadcasts) and use them efficiently within programs to start and stop scripts.
	CS1.5	**CS2.5**	**CS3.5**	**CS4.5**	**CS5.5**	**CS6.5**
	Be able to use a developing range of language and styles of control e.g. tilt and turn/instructional to direct a robot.	Be able to find a bug in simple code and attempt to debug errors.	Be able to find errors in a simple program, and successfully debug to make the program work.	Be able to find errors in a program of their own design, and successfully debug to achieve a specific goal.	Be able to use a greater range of conditionals (selection) including "whilst", "if else", "repeat until".	When debugging, can use abstraction to filter out extraneous detail and debug the program.

©hi-impact Consultancy Ltd 2022