Computing Key Stage 3 Levels

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Systems** | **Development** | **Programming** | **Modelling** | **Analysis** |
| 3 | Understand that computer systems work step-by-step and can only do what we tell them. | Be able to create a sequence of instructions and improve it if necessary. | Be able to plan a sequence of instructions for something that you want to happen. | Be able to read a sequence of instructions and predict what the result will be. | Be able to describe the goals of a given problem. |
| 4 | Be able to explain why we must be accurate when working with computers. | Write sequences of instructions and data in a way that a computer will understand. | Use selection and repetition correctly in your programs. | Be able to trace instructions using variables, selection and repetition and predict what the result will be. | Understand what is meant by a computational problem. |
| 5 | Understand how data, such as numbers, sound and images are physically stored on a computer system. | Be able to plan, create, test and reflect on a solution to a problem that a computer could solve. | Correctly use variables, lists and simple procedures in your programs. | Be able to recognise similarities between simple problems and the ways in which they can be solved. | Be able to take a problem and divide it into its main sub-problems. |
| 6 | Understand how instructions are run inside a computer. | Be able to develop solutions for problems that are described to you by someone else. | Correctly use procedures and functions with parameters in your programs. | Be able to take solutions to one problem and adapt them for similar problems. | Be able to take a problem and divide it into all its sub-problems and show this as a diagram. |
| 7 | Understand how instructions can be written efficiently and be able to describe the efficiency of your programs. | Be able to test the different modules of your programs as you are developing them, reflect on the results and then improve them. | Be able to write programs in a text-based language like Python and be able to create your own data structures. | Be able to create a simple model for a complex problem. | Be able to define an outline of a solution in terms of functions and global values. |
| 8 | Be able to show how elements of real life can be represented in programs and the difficulties that sometimes exist when doing this. | Make sure that the programs you develop have been written so they are unlikely to crash or cause errors. | Be able to create your own relational databases and use them in your programs and be able to find, understand and use techniques for specific tasks. | Be able to create an accurate, detailed model for a complex problem. | Be able to analyse real world problems and develop low-level and high-level plans for a solution. |