Progression of Disciplinary Knowledge[[1]](#footnote-1)[[2]](#footnote-2) in Computing – “Knowing How We Know”

|  |  |  |  |
| --- | --- | --- | --- |
| Substantive Knowledge[[3]](#footnote-3) | Disciplinary Knowledge | Declarative / Conceptual Knowledge | Procedural Knowledge |
| “Established facts” | “Knowing How We Know” | “Knowing That E.g. ‘I know that…’” | “Knowing How To” |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Examples of building disciplinary knowledge shown within the year group… for each activity the pupils/students will be able to demonstrate…** | | | | |
| **What we do in order to know…** | **Year 7** | **Year 8** | **Year 9** | **Year 10** | **Year 11** |
| **Abstraction -** The removal of unnecessary complexity. | Input > Process > Output  Diagrams to abstract complex systems | Caesar cipher as an abstraction of encryption.  Application of number systems - binary | App and Web design during the MTWABP project.  Data representation – Text/images | Python programming (both algorithms and the programming outputs) (CS)  Design of User Interfaces (IT)  Data models (IT) | Network Models |
| **Decomposition –** Breaking a large task into several smaller tasks, each of which being easier to solve in one go. | Flowcharts (Each item is one task/decision/input/output) | Text programming using sub-programs. | How to conduct a:   * Merge Sort * Binary Search   As ‘divide and conquer’ algorithms. | Functions and procedures in python programming. (CS)  Excel spreadsheets being separated into different worksheets (IT) | Libraries used in Python Programming  IT products being decomposed into separate sections. |
| **Algorithmic Thinking – Sequence -** Putting tasks in order | Sequence – Block-programming.  Limited / small contexts.  Familiar contexts. | Sequence – Text programming  Reducing the familiarity.  Increasing the size of the context. | Sequence – Formal Algorithms.  Unfamiliar contexts yet important to get correct. | Sequence –  Unfamiliar contexts.  Abstract / no context. (CS/IT) | Sequence –  Unseen contexts  Contexts that need to be abstracted first before the sequence can be determined. |
| **Algorithmic Thinking - Pattern recognition –** Finding ways to repeat tasks | Looping in block-based programming.  ‘Forever’ and ‘Repeat X’  Nested loops for complex pattern drawing.  Finding patterns in data - Small data sets. | Looping in text-based programming:  For and while loops.  Finding patterns in data – Large data sets. | Looping in text-based programming:  Nested Loops | Looping in text-based programming:  Increasing complexity.  Variety of loops within nests. (CS) | Problem solving using loops to solve the problems.  Unseen / complex patterns to be determined.  Variety of loops within nests. Including decisions within loops. |
| **Algorithmic Thinking – Decision -** Using logic and reasoning to determine the best path through a program. | Block-based programming using  If blocks  If..Else blocks  If…Else If… Else blocks | Text based programming  If  If..Else  If…Else If… Else  Multiple conditions that need to be compared before an outcome is found.  Boolean logic | Decisions in Text based programming  Nested selection involving Boolean Logic | Selection in text-based programming:  Increasing complexity.  Variety of selection within nests. (CS) | Problem solving using decisions to solve the problems.  Unseen / complex decisions to be determined.  Variety of decisions within nests. Including loops within decisions. |
| **Efficiency –** Completing more tasks in a shorter time frame or with less instructions. | “Robot Vacuum” – Comparison of algorithms. Which cleans best? Which is quicker?  Drawing a circle. | Formulas and functions in data handling using MS Excel. | Comparison of formal sorting and searching algorithms. E.g. Linear vs. binary search. Bubble vs. insertion sort. | Algorithm efficiency (Sorting and Searching) (CS)  Using Master slides/layouts and templates to reduce work.(IT) | Text-based programming – Functions and the reuse of program structures. |
| **Generalisation -** Reusing and repurposing previous work, therefore solving problems that have already been partly solved. | Shape drawing – E.g. “You’ve drawn a square, now change the code to draw a hexagon.”  Quiz – E.g. “You’ve made a quiz about capital cities, now make one about computer hardware.” | Spreadsheets  Word-processing  Presentations  The use of similar features of MS office software to do similar jobs. | Problems solving – “Have you solved other similar problems? How can you change those solutions to meet the needs of this problem?  Using the Pen Tool to create a logo. Then using the same tool for a different logo. | Using Functions and procedures. (CS) | Recycling code, modifying it to meet the needs of a new problem.  Using existing products and modifying them to solve new problems. |
| **Concurrency –**  Completing multiple tasks (seemingly) at the same time. | Using two sprites at the same time in Scatch, block-based programming. | Using 2/3 starts in the same Flowol program. E.g. separate starts for the heating and light in the ‘Greenhouse’ mimic. | Using 2/3 starts in the same Flowol program. E.g. separate starts for the window blind and garage door in the ‘Automatic Home’ mimic. | Knowledge of the Operating systems.(CS) | Memory management. |

1. Christine Counsell describes disciplinary knowledge as "what pupils learn ***about*** **how** that [substantive] knowledge was established, its degree of certainty and how it continues to be revised’" – [OfSTED Geography Review](https://www.gov.uk/government/publications/research-review-series-geography) [↑](#footnote-ref-1)
2. "Broadly, disciplinary knowledge introduces pupils ‘to specialised forms of knowledge, modes of thought and experience, which are the symbolic products of past human endeavours to better know the world and the people within it’." – [OfSTED History review](https://www.gov.uk/government/publications/research-review-series-history) [↑](#footnote-ref-2)
3. https://www.aidansevers.com/post/what-are-all-the-different-types-of-knowledge-part-1 [↑](#footnote-ref-3)