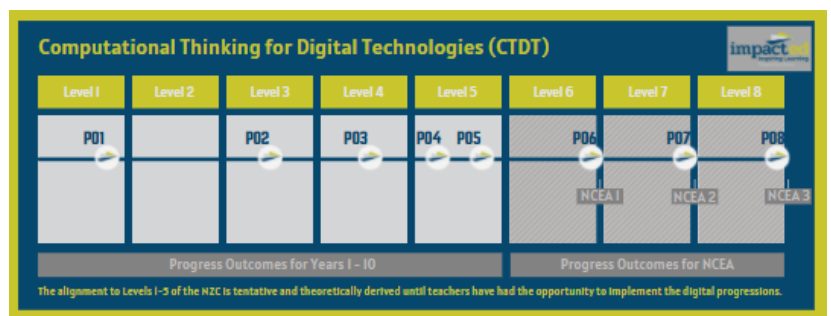


Computational Thinking (CT)

Progress Outcomes

Always:

- Take account of the end-users
- Be used in an authentic context



Key Elements	Terminology
Progress Outcome 1 (PO1)	
<ul style="list-style-type: none"> • Non-computerised • Break down tasks into step-by-step instructions • Give instructions, identify errors & fix (debug) 	<p>End-users: who will use or see this?</p> <p>Algorithm: set of instructions to solve a problem or complete a task</p> <p>Algorithmic thinking: ordered instructions</p> <p>Debug: find & remove errors</p>
Progress Outcome 2 (PO2)	
<ul style="list-style-type: none"> • Non-computerised & computerised • Give, follow & debug simple algorithms • Use algorithms to create simple programs with outputs & sequencing 	<p>Outputs: what end-user will see or use</p> <p>Program: a series of instructions put into a computer</p> <p>Sequencing: An action, or event, leads to the next ordered action</p>
Progress Outcome 3 (PO3)	
<ul style="list-style-type: none"> • Decompose problems into step-by-step instructions to create algorithms • There can be more than one algorithm for each problem • Develop and debug simple programs • Use inputs, outputs, sequence & iteration • Data is stored in 2 states shown by binary digits 	<p>Inputs: how you put data into a device</p> <p>Iteration: Repeating part of an algorithm with a loop</p> <p>Binary: a number system that only uses two digits; 1 and 0</p> <p>Bits: Binary digits</p>
Progress Outcome 4 (PO4)	
<ul style="list-style-type: none"> • Decompose problems to create simple algorithms • Program: using sequence, selection & iteration • Use algorithms to create programs that use inputs, outputs, sequence, comparative operators & iteration • Debug, fix and explain errors in algorithms & programs • Evaluate the efficiency of algorithms & user interfaces <p>Digital Devices Knowledge:</p> <ul style="list-style-type: none"> • Represent data with binary digits • Have ways to find errors in storage & transmission 	<p>Selection: In a selection structure, a question is asked, and depending on the answer, the program takes one of two courses of action,</p> <p>Comparative operators: used in conditional statements, especially loops, where the result decides whether execution should proceed</p>
Progress Outcome 5 (PO5)	
<ul style="list-style-type: none"> • Independently decompose problems into algorithms • Use algorithms to create programs with inputs, outputs, sequence, selection using comparative & logical operators, different data types & iteration • Determine when to use different control structures • Document, test & debug programs • Understand how computers store complex data types • Develop programs that consider human-computer interaction (HCI) heuristics 	<p>Logical operators: used to determine the logic between variables or values.</p> <p>Control structures: a block of programming that analyses variables and chooses a direction in which to go based on given parameters.</p> <p>Heuristics: A heuristic is a mental shortcut that allows people to solve problems and make judgments quickly and efficiently.</p>

