



# Year 11 Spring 2 Keywords:

## Computational Thinking

### Topic Title: Computational Thinking

Keyword	Definition
Abstraction	Removing unimportant parts of a problem in order to concentrate on those that are important
Decomposition	Breaking down a problem into smaller more manageable ones
Algorithmic thinking	An approach to solving problems by the use of algorithms (sequences of steps that lead to a solution)
Structure diagram	A hierarchical diagram that shows how a problem is broken down into sub-sections/sub-tasks
Binary search	This only works on a sorted list The middle item of the list is first checked If the item searched for is less than this item the right of the list is discarded, and a binary search is carried out on the left of the list
Linear search	Each item in the list is checked against the search item in order
Sorting algorithms	<ul style="list-style-type: none"> <li>• Bubble sort</li> <li>• Insertion sort</li> <li>• Merge sort</li> </ul> <p>Choice of algorithm - Merge sort is generally faster to sort lists, so would be the recommended algorithm</p>
Flowchart Symbols	<p><b>Input / Output</b></p> <p><b>Process</b> – Maths operations and assignment of variables</p> <p><b>Line</b> – shows direction of flow</p> <p><b>Terminal</b> – for start and stop</p> <p><b>Decision</b> – change flow based on a decision</p> <p><b>Sub program</b> – call a different function or procedure</p> <p> PG ONLINE</p>

Data Types	<table border="1"> <thead> <tr> <th>Data type</th> <th>Description</th> <th>Example</th> </tr> </thead> <tbody> <tr> <td>INTEGER</td> <td>A whole number</td> <td>1475, 0, -5</td> </tr> <tr> <td>REAL</td> <td>A number with a decimal point</td> <td>56.75, 6.0, -2.456, 0.0</td> </tr> <tr> <td>BOOLEAN</td> <td>Either TRUE or FALSE</td> <td>TRUE, FALSE</td> </tr> <tr> <td>CHARACTER</td> <td>A single alphabetic or numeric character</td> <td>'a', 'K', '4', '@', '%'</td> </tr> <tr> <td>STRING</td> <td>A sequence of one or more characters</td> <td>"Jo Hobson", "123"</td> </tr> </tbody> </table>	Data type	Description	Example	INTEGER	A whole number	1475, 0, -5	REAL	A number with a decimal point	56.75, 6.0, -2.456, 0.0	BOOLEAN	Either TRUE or FALSE	TRUE, FALSE	CHARACTER	A single alphabetic or numeric character	'a', 'K', '4', '@', '%'	STRING	A sequence of one or more characters	"Jo Hobson", "123"																											
Data type	Description	Example																																												
INTEGER	A whole number	1475, 0, -5																																												
REAL	A number with a decimal point	56.75, 6.0, -2.456, 0.0																																												
BOOLEAN	Either TRUE or FALSE	TRUE, FALSE																																												
CHARACTER	A single alphabetic or numeric character	'a', 'K', '4', '@', '%'																																												
STRING	A sequence of one or more characters	"Jo Hobson", "123"																																												
Boolean operators and programming symbols	<table border="1"> <thead> <tr> <th>Symbol / keyword</th> <th>Meaning</th> <th>Symbol / keyword</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>&lt;</td> <td>Less than</td> <td>+</td> <td>Concatenation</td> </tr> <tr> <td>&lt;=</td> <td>Less than or equal to</td> <td>if elseif else</td> <td>Branch depending on condition</td> </tr> <tr> <td>&gt;</td> <td>Greater than</td> <td>switch case default</td> <td>Branch depending on case</td> </tr> <tr> <td>&gt;=</td> <td>Greater than or equal to</td> <td>input()</td> <td>Get user input</td> </tr> <tr> <td>==</td> <td>Equal to</td> <td>print()</td> <td>Output to the user</td> </tr> <tr> <td>=</td> <td>Assignment</td> <td>for</td> <td>Repeat a set number of times</td> </tr> <tr> <td>!=</td> <td>Not equal to</td> <td>while</td> <td>Repeat while a condition is true</td> </tr> <tr> <td>*</td> <td>Multiply</td> <td>do until</td> <td>Do a loop until a condition is true</td> </tr> <tr> <td>^</td> <td>Exponent</td> <td>str()</td> <td>Convert to a string</td> </tr> <tr> <td>+</td> <td>Addition</td> <td>int()</td> <td>Convert to an integer</td> </tr> </tbody> </table>	Symbol / keyword	Meaning	Symbol / keyword	Meaning	<	Less than	+	Concatenation	<=	Less than or equal to	if elseif else	Branch depending on condition	>	Greater than	switch case default	Branch depending on case	>=	Greater than or equal to	input()	Get user input	==	Equal to	print()	Output to the user	=	Assignment	for	Repeat a set number of times	!=	Not equal to	while	Repeat while a condition is true	*	Multiply	do until	Do a loop until a condition is true	^	Exponent	str()	Convert to a string	+	Addition	int()	Convert to an integer	
Symbol / keyword	Meaning	Symbol / keyword	Meaning																																											
<	Less than	+	Concatenation																																											
<=	Less than or equal to	if elseif else	Branch depending on condition																																											
>	Greater than	switch case default	Branch depending on case																																											
>=	Greater than or equal to	input()	Get user input																																											
==	Equal to	print()	Output to the user																																											
=	Assignment	for	Repeat a set number of times																																											
!=	Not equal to	while	Repeat while a condition is true																																											
*	Multiply	do until	Do a loop until a condition is true																																											
^	Exponent	str()	Convert to a string																																											
+	Addition	int()	Convert to an integer																																											
Trace Tables	<p>Trace tables are used to help find errors in a program. Variable names and outputs are put in columns. The programmer traces through the program line by line. updating the values of variables and outputs. A row is used for each iteration.</p>																																													
Syntax error	<p>An error caused by not following the rules of the language e.g missing brackets or quotemarks.</p>																																													
Logical error	<p>The logic of the program is incorrect – e.g. wrong values used to create a total.</p>																																													
Boolean Functions	<p>AND, OR and NOT are Boolean operators A computer can calculate the results of <b>A AND B</b>, <b>A OR B</b>, or <b>NOT A</b></p>																																													
Truth Tables	<p>A <b>truth table</b> shows the output from all possible combinations of inputs from a Boolean expression</p>																																													
Logic Gates	<p>A logic gate is a device that acts as a building block for digital circuits. They perform basic logical functions that are fundamental to digital circuits.</p>																																													
Logic Diagrams	<p>Diagrams to represent digital circuits and logic gates.</p>																																													
Truth Table	<p>It shows all possible combinations of inputs and the outputs they create.</p>																																													
Input validation	<p>Checking input meets certain rules, e.g. the type of data</p>																																													
Anticipating misuse	<p>Preventing too many entries of a password to make it harder for hackers to guess</p>																																													
Authentication	<p>Entering data twice or checking from an alternative source</p>																																													
Syntax errors	<p>A syntax error is one where the code written doesn't conform to the rules of the language</p>																																													

Logic Error	The program will run, but it won't work as the programmer intended
Machine Code	Instructions that computers can understand e.g. binary
Assembly language	Allows a programmer to create programs more easily than writing in machine code
High level languages	High-level languages – programming language such as Python that generally have statements that look a bit like English or Maths.