


<b>Year 8 Computer Science</b>	<b>Curriculum Intent:</b> Computing consists of three strands: Computational Thinking, Computer Science and Digital Literacy. Within the Computer Science and Computational Thinking streams we will develop the student’s knowledge and understanding of what Binary is and how Networks operate. In addition, we will examine the logic behind instructions and how they are processed. Therefore, providing opportunities for students to develop their own software using a text-based programming language.			
	<p style="text-align: center;"><b>Term 1</b>  <b>Introduction to Programming &amp; Developing          Practical Programming Skills</b></p>	<p style="text-align: center;"><b>Term 2</b>  <b>Flowol and G-Develop</b></p>	<p style="text-align: center;"><b>Term 3</b>  <b>Microbits and Networking</b></p>	
<b>Topic Titles (in          order of delivery)</b>	<p><b>Algorithms</b></p> <ul style="list-style-type: none"> <li>• Computational logic</li> <li>• Flow charts</li> <li>• Pseudo Code</li> </ul> <p><b>Introduction to programming</b></p> <ul style="list-style-type: none"> <li>• Variables</li> <li>• Input, output, storage</li> <li>• Sequence</li> <li>• Selection</li> <li>• Iteration</li> <li>• Operators</li> </ul> <ul style="list-style-type: none"> <li>• Practical Programming Skills</li> </ul>	<p><b>Flowol</b></p> <ul style="list-style-type: none"> <li>• Control flowchart symbols</li> <li>• Developing a control solution</li> <li>• Decision symbols</li> <li>• Control solution with sensors</li> <li>• Control solution with subroutines</li> <li>• Actuators and Variables</li> </ul> <p><b>Game Development with G-Develop</b></p> <ul style="list-style-type: none"> <li>• Be able to change the properties of a particle emitter object</li> <li>• Create instances of an object</li> <li>• Alter the orientation of an object</li> <li>• Game Objectives</li> <li>• Characters and Sprites</li> <li>• Collision Detection</li> <li>• Falling and Spawning</li> <li>• Rewards and Hazards</li> </ul>	<p><b>Physical Computing/Microbits</b></p> <ul style="list-style-type: none"> <li>• Introduction to Micro bit</li> <li>• Built in – Inputs and Outputs</li> <li>• Creating a program</li> <li>• Pair programming</li> <li>• Physical computing project</li> <li>• Refining and refining project</li> </ul> <p><b>Networks</b></p> <ul style="list-style-type: none"> <li>• Network Protocols</li> <li>• Network Hardware</li> <li>• Wired and Wireless Networks</li> <li>• Network Topologies</li> </ul>	

<p><b>Key knowledge / Retrieval topics</b></p>	<ul style="list-style-type: none"> <li>• Algorithms are a set of instructions to solve a problem / perform a task.</li> <li>• How to use flowcharts to describe / define an algorithm</li> <li>• Know the main features of a flowchart.</li> <li>• A Variable is a 'named location in memory' the contents of which can change while the program is running.</li> <li>• How to assign values to variables</li> <li>• How to use variables to produce output, store input and perform calculations</li> <li>• Sequence is arranging instructions in a logical order.</li> <li>• Selection is deciding which instructions to execute based on a Boolean decision.</li> <li>• Iteration is the execution of a set of instruction several times based upon a Boolean decision.</li> </ul>	<ul style="list-style-type: none"> <li>• Operate systems automatically i.e. traffic lights.</li> <li>• Gather data automatically i.e. weather stations.</li> <li>• Computer to control the action or movement of an object i.e. robot cleaners!</li> <li>• An actuator is often part of a computer control system.</li> <li>• The actuator is a mechanical device or motor which carries out the action or decision made by the control system.</li> <li>• What do symbols mean?</li> <li>• Start symbol, process symbol and output symbol.</li> </ul>	<p>Microbits</p> <ul style="list-style-type: none"> <li>• Describe microbit</li> <li>• List the micro bit's input and output devices</li> <li>• write a Python program for the micro: bits</li> </ul> <p>Design</p> <ul style="list-style-type: none"> <li>• Write programs that use the microbit's built-in input and output devices</li> <li>• Communicate with other devices</li> <li>• Write programs that use GPIO pins to generate output and receive input</li> <li>• Write programs that communicate with other devices by sending and receiving messages wirelessly</li> </ul> <p>Network Protocols</p> <ul style="list-style-type: none"> <li>• Email servers</li> <li>• World Wide Web</li> </ul> <p>Network Hardware</p> <ul style="list-style-type: none"> <li>• Cables</li> <li>• Hub</li> <li>• Routers</li> <li>• Servers</li> </ul> <p>Wired and Wireless Networks</p> <ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• Wifi</li> <li>• 3G</li> <li>• 4G</li> <li>• Wired and Wireless Devices</li> </ul> <p>The Internet</p>
--	---	--	---

<p style="text-align: center;"><b>Understanding / Sequence of delivery</b></p>	<p>Programming techniques</p> <p>Variables</p> <ul style="list-style-type: none"> <li>• State what a variable is (A named location in memory that can change whilst the program is running)</li> <li>• Explain what a variable is (to store data while a program is running)</li> <li>• Perform basic mathematical or logical functions on variables</li> </ul> <p>Input, output, storage</p> <ul style="list-style-type: none"> <li>• Explain and show how input may be captured and assigned to a variable for use / storage within a program</li> <li>• Explain and show how to output text or movement on a screen</li> <li>• Be able to use a range of data types including: <ul style="list-style-type: none"> <li>▪ Integers</li> <li>▪ Real numbers (floats)</li> <li>▪ Text</li> <li>▪ Boolean</li> <li>▪ Lists / arrays (one dimensional) or equivalent</li> </ul> </li> </ul> <p>Sequence</p> <ul style="list-style-type: none"> <li>• Understand that instructions are executed in the sequence they are written</li> <li>• Write programs with instructions in the correct order</li> </ul>	<ul style="list-style-type: none"> <li>• Control Systems</li> <li>• What is a control system.</li> <li>• Identify Control system using flow charts.</li> <li>• Identify set of sequence</li> <li>• Impact of control systems on everyday life.</li> <li>• Creating a flow chart to demonstrate control systems</li> <li>• Identify subroutines</li> <li>• Creating a game</li> <li>• Changing the properties of a particle emitter object</li> <li>• Creating instances of an object</li> <li>• Altering the orientation of an object</li> <li>• Design sprites for a platform game</li> <li>• Be able to create and edit collision masks</li> <li>• Test and refine the playability of a game</li> <li>• The main character sprite falling off the screen</li> <li>• The main character sprite spawning back at the start</li> <li>• The main character reaching the end of the game</li> <li>• Be able to use variables to store a score</li> <li>• Update a score based on an event – such as a coin being collected</li> <li>• Be able to create hazard objects that return a character to the start of the game</li> </ul>	<ul style="list-style-type: none"> <li>• IP Addresses</li> <li>• TCP/IP</li> <li>• Packets</li> </ul> <p>Internet Services</p> <ul style="list-style-type: none"> <li>• Email</li> <li>• VoIP</li> </ul> <p>Network Topologies</p> <ul style="list-style-type: none"> <li>• Client Server Network Model</li> <li>• Peer to Peer Network Model</li> <li>• Bus Topology</li> <li>• Star Topology</li> <li>• Mesh</li> <li>• Ring</li> </ul> <ul style="list-style-type: none"> <li>• Physical computing components</li> </ul> <p>Design a physical computing artifact purposefully, keeping in mind the problem at hand, the needs of the audience involved, and the available resources</p> <ul style="list-style-type: none"> <li>• Decompose the functionality of a physical computing system into simpler features</li> </ul> <p>Physical computing project</p> <ul style="list-style-type: none"> <li>• Implement a physical computing project, while following, revising, and refining the project plan</li> </ul>
--	--	---	--

	<ul style="list-style-type: none"> <li>• Be able to identify errors in the order of a sequenced set of steps</li> </ul> <p>Selection</p> <ul style="list-style-type: none"> <li>• Explain and identify how programs can be made to execute code based on a choice (true or false)</li> <li>• Write programs that use selection <ul style="list-style-type: none"> <li>▪ If</li> <li>▪ Else</li> <li>▪ Else it (elif)</li> </ul> </li> </ul> <p>Iteration</p> <ul style="list-style-type: none"> <li>• Understand what is meant by a loop</li> <li>• Use a loop to execute statements multiple times</li> <li>• State the difference between a for loop and a while loop</li> </ul> <p>Operators</p> <ul style="list-style-type: none"> <li>• Use common arithmetic operators in a program (+-*/)</li> <li>• Use common Boolean operators in a program (==, !=, &lt;,&gt;)</li> </ul> <p>Comments</p> <ul style="list-style-type: none"> <li>• Explain why comments in code are useful</li> <li>• Show examples of commenting in code</li> </ul>		<ul style="list-style-type: none"> <li>• Computer Networks and Protocols</li> <li>• Network Hardware</li> <li>• Wired and Wireless Network</li> <li>• The Internet</li> <li>• Internet Services</li> <li>• Network Topologies</li> </ul>
	<p>Algorithms:</p> <ul style="list-style-type: none"> <li>• Computational logic</li> </ul> <p>Identify the success criteria of a problem</p> <ul style="list-style-type: none"> <li>• Flow charts</li> </ul>		

	<ul style="list-style-type: none"><li>• Produce algorithms using flow charts</li><li>• Use and be familiar with the flow chat shapes</li><li>▪ Start / Stop (End)</li><li>▪ Process</li><li>▪ Input / Output</li></ul>		
<b>Assessments</b>	Assessment on the above content	Assessment on the above content	Assessment on the above content