



## YEAR 12 A LEVEL COMPUTER SCIENCE SPRING TERM 2 – PAPER 2

‘An ambitious curriculum that meets the needs of all’

### Medium Term Planning – Python Programming Techniques

Curriculum Intent	Pupils will be taught the following National Curriculum guidelines this term:
Skills/Assessment Objective Links	<p><b>At the end of this Unit all students should be able to:</b></p> <ul style="list-style-type: none"><li>▪ use functions and library subroutines including random number generation</li><li>▪ know how to define and call a subroutine (procedure or function) with parameters</li><li>▪ construct algorithms using one-dimensional arrays</li><li>▪ describe what is meant by recursion</li><li>▪ define the OOP terms<ul style="list-style-type: none"><li>○ Class</li><li>○ Object</li><li>○ Method</li><li>○ Attribute</li><li>○ Inheritance</li><li>○ Encapsulation</li><li>○ Polymorphism</li></ul></li><li>▪ draw an inheritance diagram</li></ul> <p><b>Most students will be able to:</b></p> <ul style="list-style-type: none"><li>▪ use structured programming techniques and write their own subroutines with parameters</li><li>▪ construct algorithms using two-dimensional arrays</li><li>▪ use local and global variables in subroutines</li><li>▪ trace through a recursive algorithm</li><li>▪ compare iterative and recursive algorithms for solving a problem</li><li>▪ complete given pseudocode for an object-oriented program</li></ul> <p><b>Some students will be able to:</b></p> <ul style="list-style-type: none"><li>▪ write complex algorithms involving data structures, subroutines, and file-handling</li><li>▪ interpret complex algorithms and determine the output</li><li>▪ explain why using local variables makes a program easier to maintain</li><li>▪ distinguish between passing parameters by value and by reference</li><li>▪ write a recursive algorithm to solve a problem</li><li>▪ use object-oriented programming techniques to solve problems</li></ul>
Numeracy	
Literacy	<p><b>Vocabulary Tier 3:</b> Integrated Development Environment (IDE), syntax errors, logic errors, debug, watch, breakpoint, trace algorithm, structured programming, data type, variables, constants, assignment, arithmetic operations, Boolean operators, sequence, selection, branching, definite and indefinite iteration, top down design, modular programming, subroutine, procedure, function, parameter, argument, pass by value, pass by reference, global and local variables, recursion, object oriented programming, class, object, method, attribute, inheritance, encapsulation, polymorphism</p> <p><b>Vocabulary Tier 2:</b> method, data, logic</p> <p><b>Reading:</b> Worksheets, presentations, answer sheets, exam questions, mark scheme, further reading for homework, conduct research for NEA</p> <p><b>Writing:</b> Answer on the worksheet via word, complete NEA</p> <p><b>Oracy:</b> Listening and using tier 3 words</p>
Becoming future ready	<p><b>Careers/Employability:</b></p> <p>Understand the grade requirements at universities and the topics that can be applied for. Explore apprenticeship opportunities with a range of industries.</p> <ul style="list-style-type: none"><li>▪ Software Architect.</li><li>▪ Data Scientist.</li></ul>

	<ul style="list-style-type: none"> <li>Machine Learning Engineer.</li> <li>Blockchain Developer</li> <li>Cybersecurity Engineer.</li> <li>Cloud Solutions Architect.</li> <li>AI Research Scientist.</li> <li>Full-Stack Developer.</li> </ul>
<b>Adaptation</b>	Throughout this topic, quality first teaching will provide differentiation:
<b>QFT/SEND Provision</b>	<p><b>By product:</b> Learners are asked to present outcomes writing different code, not all code will be equal in style and sophistication, all code will work with teachers input, top end programmers will be set challenges on how to extend code and be expected to conduct a level of independent research</p> <p><b>By resource:</b> Worksheets are well presented and accessible. Instructions are clearly outlined and separate from the information so that pupils know where to begin and end. Handouts are differentiated by outcome. Resources used will appeal to the range of preferred learning styles of pupils e.g. visual, auditory or kinesthetic learners. Scaffolding of tasks – word frames.</p> <p><b>By Intervention:</b> By providing different levels of supervision and support depending on coding ability</p> <p><b>By Progressive Questioning:</b> Exploring pupils' understanding of programming by setting adaptive challenges</p> <p><b>By Grouping:</b> According to coding ability, prior attainment, gender, social preference, preferred learning style.</p> <p><b>By Task:</b> Pupils identify targets which are meaningful via level of coding ability</p> <p><b>By Offering Optional Activities:</b> In class or as homework, to extend learning.</p> <p>This QFT/SEND provision will be explicit within the lesson by lesson schemes of work.</p>
<b>Implementation Curriculum Delivery</b>	To be able to:
<b>Learning Outcomes (Knowledge)</b>	<p><b>Topic 1 Arrays, Tuples and Records</b> Be familiar with the concept of a data structure Understand how data is represented and stored within different structures including</p> <ul style="list-style-type: none"> <li>Arrays up to three dimensions</li> <li>Tuples</li> <li>Records</li> <li>Use 1- 2- and 3-dimensional arrays in the design of solutions to simple problems</li> </ul> <p><b>Topic 2 Subroutines</b> Be familiar with subroutines, their uses and advantages Use subroutines that return values to the calling routine Use arguments/parameters to pass data to subroutines by value and by reference Contrast the use of local and global variables</p> <p><b>Topic 3 Local and Global Variable by reference and value</b> Understand the difference between a local and global variable Understand the difference between passing as a reference or value Understand the storage impact of passing between reference and value Understand the validation impact of global and local variables.</p> <p><b>Topic 4 OOP Object Oriented Programming</b> Describe the features of an object-oriented language:</p> <ul style="list-style-type: none"> <li>Classes</li> <li>Object</li> <li>Method</li> <li>Attributes</li> <li>Inheritance</li> <li>Encapsulation</li> <li>Polymorphism</li> </ul> <p>Write pseudocode for a class definition Write pseudocode to instantiate an object and use its methods</p>



Draw inheritance diagrams  
Describe the advantages of an object-oriented approach to programming

**Topic 5 Recursion**  
Use recursion to solve simple problems  
Trace a recursive algorithm  
Compare recursion to an iterative approach

Programing end of unit assessment

Current learning to be developed in the future within:	NEA, data structures, algorithms and computational thinking.
Assessment	See assessment maps for formative and summative assessment opportunities.
Impact	<p>Review assessment results and target pupils that require further support via:-</p> <ul style="list-style-type: none"><li>• Learning conversation</li><li>• Changing seating plan</li><li>• Plan lessons to address areas of concern in assessment</li><li>• Targeted homework based on low performance areas identified in the assessment and marked pieces</li><li>• Stretch and challenge high ability pupils by identifying ambitious next steps to expand knowledge</li></ul> <p>Create a feedback sheet for each student Each student identifies areas of Green, Amber and Red using Mark Assessment on their feedback sheet Complete NOW task on areas identified as Amber and Red</p>