



Curriculum Map: Year 11 Computer Science

Topic	Key Knowledge <i>What will all students KNOW by the end of the topic?</i>	Key Skills <i>What key skills will be learnt/developed by the end of the topic? What will all students be able to DO by the end of the topic?</i>	Assessment Opportunities <i>What are the key pieces of assessment? How will students be assessed?</i>
Basic Programming concepts	Students will be able to: Know the different variable data types. Understand the difference between arrays and records. Know the arithmetic, logical and comparison operators. Understand what a variable is and when to use one. Understand what a constant is and when to use one. Understand how to get input from the keyboard. Understand what assignment is. Know the 3 programming constructs: Sequence, Selection, Iteration/Repetition Understand the use of nested selection and iteration structures. Understand that programs can be structured using procedures and functions.	Students will: Continue to program in Python following the Try, Investigate, Make and Evaluate challenges and Defold 2D games tutorials. Know how to convert from one data type to another. Know how to create, append to, and delete from a list. Know how to generate and customise outputs to the screen.	Students will be assessed by: SLR 7 Student workbook SLR 7 End of topic test Paper 2 exam revision activities
Advanced programming concepts	Students will be able to: Understand and be able to use a range of string manipulation functions. Understand how to use random number generation. Understand how to use the math module. Understand how to use the time module to pause the execution of programs.	Students will: Continue programming in Python following the Try, Investigate, Make and Evaluate challenges and Defold 2D games tutorials. Generate random numbers. Draw shapes with Turtle.	Students will be assessed by: SLR 8 Student workbook SLR 8 End-of-topic test Paper 2 exam revision activities

Curriculum Map: Year 11 Computer Science

	<p>Understand how to use the Python turtle module, including use of movement and pen operations.</p> <p>Understand how to use subprograms.</p> <p>Explain the advantages of using subprograms.</p> <p>Understand the difference between a procedure and a function.</p> <p>Describe the user of parameters to pass data within programs.</p> <p>Understand that subroutines can return values to the calling routine.</p> <p>Understand how to use subprograms.</p> <p>Understand how to use local variables.</p> <p>Explain the importance of using local variables.</p> <p>Understand how to use global variables.</p> <p>Understand how to pass variables between parts of a programs.</p>	<p>Open, read from and write to files.</p> <p>Pass data to programs.</p>	
<p>Robust and secure programming</p>	<p>Students will be able to:</p> <p>Know four reasons why a program should be tested.</p> <p>Know what syntax, logic error and runtime errors are.</p> <p>Understand what we mean by robust software.</p> <p>Understand the purpose of an audit trail.</p> <p>Understand the purpose of a code review.</p> <p>Know why creating easy to read code is important with large projects.</p> <p>Understand what programmers can do to make their code more readable.</p> <p>Understand how to refine algorithms to make them more robust.</p> <p>Understand why data validation is necessary.</p> <p>Know a range of validation techniques that can be used to write simple data validation routines.</p>	<p>Students will:</p> <p>Continue programming in Python following the Try, Investigate, Make and Evaluate challenges and Defold 2D games tutorials.</p> <p>Know how to identify and fix errors.</p> <p>Create easy to read code.</p> <p>Be able to write simple authentication routines.</p>	<p>Students will be assessed by:</p> <p>SLR 9 Student workbook – part 2</p> <p>SLR 9 End-of-topic test – part 2</p> <p>Paper 2 exam revision activities</p>

Curriculum Map: Year 11 Computer Science

	<p>Understand some authentication techniques a programmer may choose to use to protect their program from misuse.</p>		
<p>Algorithms and computational logic</p>	<p>Students will be able to:</p> <ul style="list-style-type: none"> Be able to define the term 'algorithm'. Know what is meant by the term 'abstraction'. Know the advantages of decomposition when applied to programming. Know an example of problem decomposition. Know the flow diagram symbols. Understand how to write pseudocode. Know how to make a flow diagram. Understand how to interpret algorithms. Understand how to correct algorithms. Be able to identify inputs, processing, and outputs from an algorithm. Understand how to use visual inspection. Understand how to use trace tables. Understand the linear search algorithm. Understand the binary search algorithm. Know the special condition of the list of items for the binary search to work. Understand which searching algorithm is quicker. Understand the bubble sort algorithm. Understand that certain algorithms can be more efficient (quicker) at solving a problem than another. Know what is meant by in-place and out-of-place algorithms. 	<p>Students will:</p> <ul style="list-style-type: none"> Continue programming in Python following the Try, Investigate, Make and Evaluate challenges and Defold 2D games tutorials. Solve computational problems by a systematic approach to problem solving. Create flow diagrams. Write a program from a flow diagram and pseudocode. Understand how to construct truth tables from logic gates NOT AND OR. Be able to interpret the results of simple truth tables. Know how to work out the best-case and worse-case scenarios for a linear search and binary search. Improve algorithm code to make it more efficient. Understand how to create, complete, or edit truth tables for given scenarios. 	<p>Students will be assessed by:</p> <ul style="list-style-type: none"> SLR 10 Student workbook SLR 10 End-of-topic test Paper 2 exam revision activities

Curriculum Map: Year 11 Computer Science

<p>Classification of programming languages</p>	<p>Students will be able to:</p> <p>Know the characteristics of low-level and high-level programming languages. Explain the main differences between low-level and high-level languages. Understand the advantages and disadvantages of low-level language programming compared with high-level language programming. Understand the relationship between machine code and low-level languages. Explain the difference between machine code and low-level languages. Know what a translator does. Understand the differences between compilers and interpreters. Understand when it would be appropriate to use each type of translator.</p>	<p>Students will:</p> <p>Continue programming in Python following the Try, Investigate, Make and Evaluate challenges and Defold 2D games tutorials.</p> <p>Carry out translations. Investigate the use of prime numbers in an emulator.</p>	<p>Students will be assessed by:</p> <p>SLR 11 Student workbook SLR 11 End-of-topic test Paper 2 exam revision activities</p>
<p>Exam Revision</p>	<p>Further programming practice, completion of programming challenges, exam technique, exam questions, recall activities, Isaac computing, Seneca Learning, Smart Revise tests.</p>		