

2. Year 11 Curriculum Plan - TEMPLATE

YEAR 11	HALF TERM 2	SUBJECT Computer Science	LEARNING
----------------	--------------------	---------------------------------	-----------------



TOPIC	LEARNING OBJECTIVES	KEY VOCABULARY	LEARNING SEQUENCE	LINKED LEARNING	HOME LEARNING
SLR 2.1 Algorithms			In pupil friendly language. Headings in pupils' books should match text here.	Links to: <ul style="list-style-type: none"> Year 6 work (for Year 7 plans) Other topics Other subjects. (concise and most important links only).	
SLR 2.1 – Lesson 6, Algorithmic thinking	<ul style="list-style-type: none"> Understand how to solve computational problems by applying algorithmic thinking. KEY QUESTION: What do we mean by “thinking algorithmically”?	Slides 134-151 Computational Thinking, Abstraction, Decomposition, Algorithmic thinking, Problem inputs, Problem processes, Problem outputs, Structure diagram, Pseudocode, Flowchart, Trace table, Searching algorithms, Binary search, Linear search, Sorting algorithm, Bubble sort, Merge sort, Insertion sort	Continue working on problems from either slide 10 or slide 12		2.1 Linear search
SLR 2.1 – Lesson 7, Linear search	<ul style="list-style-type: none"> Understand the linear search algorithm. Understand it is not an efficient algorithm, but it is easier to program than alternatives and does not require the items to be in any order. KEY QUESTION: How does a linear search work?		SLR 2.1 Workbook Complete slide 12 Write a program to perform a linear search to output the latitude and longitude of a given capital city		2.1 Binary search
SLR 2.1 – Lesson 8, Binary search	<ul style="list-style-type: none"> Understand the binary search algorithm. 		SLR 2.1 Workbook Complete slides 13-14		2.1 Bubble sort

	<ul style="list-style-type: none"> • Know the special condition of the list of items for the binary search to work. • Understand which searching algorithm is quicker. <p>KEY QUESTION: How does a binary search work?</p>		Write a program to perform a binary search on a list of items		
SLR 2.1 – Lesson 9, Bubble sort	<ul style="list-style-type: none"> • Understand the bubble sort algorithm. <p>KEY QUESTION: How does a bubble sort work?</p>		SLR 2.1 Workbook Complete slide 15	Write a program to perform a bubble sort on a list of items	2.1 Merge sort 2.1 Insertion sort
SLR 2.1 – Lesson 10, Merge sort and insertion sort	<ul style="list-style-type: none"> • Understand the merge sort algorithm. • Understand the insertion sort algorithm. <p>KEY QUESTION: How does a merge sort work? How does an insertion sort work?</p>		SLR 2.1 Workbook Complete slides 16-18	Continue working on programs you have already started in this unit.	2.1 How to produce algorithms using pseudocode and flow diagrams
			There is no need to learn how to program the merge sort as it requires some A'level knowledge.		
			You could have a go at programming an insertion sort on a list if you wanted another super challenge		
SLR 2.1 – Lesson 11, How to produce algorithms	<ul style="list-style-type: none"> • Know the flow diagram symbols. • Know that flow diagrams are also called flowcharts. • Know how to make a flow diagram. • Understand how to construct a program from a flow diagram. • Know what is meant by the term pseudocode. • Understand how to write pseudocode. • Understand the OCR reference language. 		SLR 2.1 Workbook Complete slides 19-21	Create the program on slide 21	None for this lesson

	<p>KEY QUESTION: How can algorithms be described without ambiguity?</p>				
SLR 2.1 – Lesson 12, How to produce algorithms	<ul style="list-style-type: none"> • Know how to make a flow diagram. • Understand how to write pseudocode. • Understand how to write a program from a flow diagram and pseudocode. • Understand the OCR reference language. <p>KEY QUESTION: How can algorithms be described without ambiguity?</p>		<p>SLR 2.1 Workbook Complete slides 22-23</p> <p>Write the program described on slide 23</p>		None for this lesson
SLR 2.1 – Lesson 13, Interpret, correct or complete algorithms	<ul style="list-style-type: none"> • Understand how to interpret algorithms. • Understand how to correct algorithms. • Understand the OCR reference language. <p>KEY QUESTION: How do you express algorithms using the exam board reference language?</p>		<p>SLR 2.1 Workbook Complete slides 24-25</p> <p>Write the program described on slide 25</p>		None for this lesson
SLR 2.1 – Lesson 14, How to produce algorithms	<ul style="list-style-type: none"> • Know how to make a flow diagram. • Understand how to write pseudocode. • Understand how to write a program from a flow diagram and pseudocode. • Understand the OCR reference language. <p>KEY QUESTION: How can algorithms be described without ambiguity?</p>		<p>SLR 2.1 Workbook Complete slides 26-27</p> <p>Write the program described on slide 27</p>		2.1 Identifying errors and suggesting fixes
SLR 2.1 – Lesson 15, Identifying common errors and suggesting fixes	<ul style="list-style-type: none"> • Know what a syntax error is. • Know what a logic error is. • Know how identify simple syntax and logic errors in high-level code and the OCR reference language. 		<p>SLR 2.1 Workbook Complete slides 28-30</p> <p>Complete or enhance any programs from this unit</p>		2.1 Trace tables

	<ul style="list-style-type: none"> Understand how to suggest fixes to code by spotting syntax and logic errors. <p>KEY QUESTION: What are the different types of errors that can occur when programming?</p>				
SLR 2.1 – Lesson 16, Trace tables	<ul style="list-style-type: none"> Know what a trace table is. Understand how trace tables can be useful for debugging. Understand how to complete a trace table. <p>KEY QUESTION: How and why do programmers use a trace table?</p>		SLR 2.1 Workbook Complete slide 31 Complete or enhance any programs from this unit		Revise what you have learned in this unit
Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the “Exam revision unit” folder		None for this lesson
Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the “Exam revision unit” folder		None for this lesson
SLR 2.1 – End of topic test	End of topic test		Test - SLR 2.1		None for this lesson
SLR 2.1 – Action	Action / Response lessons		Chance for students to respond to feedback, improve workbooks, correct misunderstandings		None for this lesson
Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the “Exam revision unit” folder		None for this lesson
Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the “Exam revision unit” folder		None for this lesson

Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the "Exam revision unit" folder		None for this lesson
Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the "Exam revision unit" folder		None for this lesson
Dedicated paper 2 exam revision lesson	Gain experience in answering computational thinking, algorithms and programming questions for component J277/02 using our dedicated exam Revision unit.		Progress with activities in the "Exam revision unit" folder		None for this lesson