3. Year 11 Curriculum Plan - TEMPLATE

YEAR 11	HALF TERM 3	SUBJECT Computer Science	LEARNING

TOPIC	LEARNING OBJECTIVES	KEY VOCABULARY	LEARNING SEQUENCE	LINKED LEARNING	HOME LEARNING
.			In pupil friendly language. Headings in pupils' books should match text here.	Links to: • Year 6 work (for Year 7 plans) • Other topics • Other subjects. (concise and most important links only).	
SLR 2.3 Producing robust programs					2.3 Defensive design considerations part 1
SLR 2.3 – Lesson 1, Input validation	• Know what is meant by the term "defensive design considerations" when writing programs. • Understand why input validation is necessary. • Know a range of validation techniques that can be used to write a robust program. KEY QUESTION: What issues should a programmer consider to ensure a program caters for all likely input values?	**Slides 200-217** Defensive design, Anticipating misuse, Authentication, Input validation, Maintainability, Naming conventions, Indentation, Commenting, Testing, Iterative testing, Final/terminal testing, Syntax error, Logical error, Test data, Test data: Normal, Test data: Boundary, Test data: Invalid, Test data: Erroneous	SLR 2.3 Workbook Complete slide 2 Write a program that asks the user to enter a date in the format dd/mm/yyyy The program should validate the data in the following ways, and output which type of check was failed or that the date is valid:		2.3 Defensive design considerations part 2
SLR 2.3 – Lesson 2, Defensive design consideration	• Know what is meant by the term "defensive design considerations" when writing programs.		SLR 2.3 Workbook Complete slides 3-4		2.3 Maintainability 2.3 Refining algorithms to make them more robust

	• Know a range of potential problems that can occur when a program is running, especially if it requires communication to servers, peripherals, data in files and arithmetic. • Understand some authentication techniques a programmer may choose to use to protect their program from misuse. KEY QUESTION: What issues should a programmer consider to ensure a program caters for all likely input values?		Continue the date validation program. Write a program to validate an email address: must contain a @. Must be sanitised to lower case. Dot cannot be a first or last character. Double dots are not permitted. SUPER CHALLENGE: Research what makes a strong password. Create a program to validate a secure password.		
SLR 2.3 – Lesson 3, Maintainability and refining algorithms	• Know why creating easy to read code is important with large projects. • Understand what programmers can do to make their code more readable. • Understand how to refine algorithms in order to make them more robust. KEY QUESTION: What does code maintainability mean?		SLR 2.3 Workbook Complete slides 5-6 Continue the date validation program. Continue the validate email address program. Add sanitisation and validation to the greatest common factors program on slide 5.		2.3 The purpose and types of testing 2.3 How to identify syntax and logic errors
SLR 2.3 – Lesson 4, Types if testing and errors	• Know four reasons why a program should be tested. • Know what iterative testing is. • Know what final/terminal testing is. • Know what a syntax error is. • Know what a logic error is. KEY QUESTION: What are the different types of errors that can occur in a program?		SLR 2.3 Workbook Complete slide 7-12 Continue the date validation program. Continue the validate email address program. Continue the greatest common factors program. Create the program shown in the starter to output the factorial of a number. Include suitable input sanitisation and validation for the program.		2.3 Suitable test data

SLR 2.3 – Lesson 5, Suitable test data	• Understand that because a program works, it doesn't mean it works for all inputs.
	• Understand that suitable test data for a program needs to include:
	  ○ Normal data
	  ○ Boundary data
	  ○ Invalid data
	  ○ Erroneous data
	KEY QUESTION:
	What are the features of good testing strategy?
			SLR 2.3 Workbook
			Complete slides 13-15
			
			Complete any outstanding programs.
			Write a program to simulate an input tweet of up to 280 characters. It should allow the user to enter text and output the number of characters that were remaining after the input. Inputs of more than 280 characters are rejected with the number of characters over shown as a negative number.
			
			SUPER CHALLENGE: Can you allow the user to enter a multi-line tweet using a list to store each line input, terminating when the line contains no characters? Extend the program so it only stops when a tweet of 0 characters is entered.
					None for this lesson

SLR 2.3 – Lesson 6, Complete outstanding programs	• Understand how robust programs are made. KEY QUESTION: What makes a robust program?		SLR 2.3 Workbook Complete any outstanding workbook pages Complete any outstanding programs: Date validation program. Secure password validation program. Greatest common factors program. Factorial numbers program. Twitter validation program. SUPER CHALLENGE: Write a program to convert a decimal into a fraction. E.g.		Revise what you have learned in this unit

			0.5 = ½, 0.75 = ¾. You will need to use your greatest common factors program and research a suitable algorithm.		
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 2.3 – End of topic test	End of topic test		Test - SLR 2.3		None for this lesson
SLR 2.3 – 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