LONG-TERM OVERVIEW

YEAR 12			
Term	Topics	Assessment	Notes
1	• Introduction to the course • SLR1 Programming basics (10 lessons) • SLR2 Programming next steps (8 lessons) • Plus 18 dedicated programming lessons	• Completed SLRs 1-2 form the basis for assessment • SLR 1-2 exam questions	• The Structured Learning Records included for capturing students' evidence of learning and assessment include both AS and Full A Level spec points. • For delivery of the AS course only, you can ignore any specification points written in gold text. • The following Structured Learning Records contain full A Level content ONLY and do not need to be delivered as part of an AS only course. o SLR05 Algorithms o SLR08 Classification of algorithms o SLR09 A model of computation o SLR22 TCP-IP and protocols o SLR23 Databases o SLR24 Big Data o SLR25 The functional programming paradigm o SLR26 Writing functional programs
2	• SLR3 Programming paradigms (3 lessons) • SLR4 Data structures (4 lessons) • SLR6 Abstraction and automation (10 lessons) • SLR7 Regular and context-free languages (2 lessons) • Plus 26 dedicated programming lessons	• Completed SLRs 3, 4, 6 and 7 form the basis for assessment • SLR 3, 4, 6 and 7 exam questions	
3	• SLR10 Number system and bases (9 lessons) • SLR11 Binary (5 lessons) • SLR12 Coding text and graphics (7 lessons) • Plus 15 dedicated programming lessons	• Completed SLRs 11-12 form the basis for assessment • SLR 10-12 exam questions	
4	• SLR13 Coding sound and music (6 lessons) • SLR14 Hardware and software (5 lessons) • SLR15 Programming languages and translators (5 lessons) • SLR16 Logic gates and Boolean algebra (4 lessons) • SLR17 Internal computer architecture (8 lessons)	• Completed SLRs 13-17 form the basis for assessment • SLR 13-17 exam questions	
5	• SLR18 Input and output devices (5 lessons) • SLR19 Moral, social, legal, cultural issues (2 lessons) • SLR20 Communication (4 lessons) • SLR21 Network and the internet (4 lessons) • SLR27 Aspects of software development (4 lessons) • Plus 11 dedicated revision lessons	• Completed SLRs 18-21 and 27 form the basis for assessment • SLR 18-21 and 27 exam questions	

The dedicated programming lessons are for students to engage in self-directed programming. We have hundreds of activities, worksheets and programming challenges for them to complete, available through your premium resources account.

For a detailed breakdown of which lessons to deliver week by week, see our Excel delivery calendar AQA AS-Level Only - 1-week model (delivery calendar).xlsx, which this SoL is based on.

SHORT-TERM SCHEME OF LEARNING

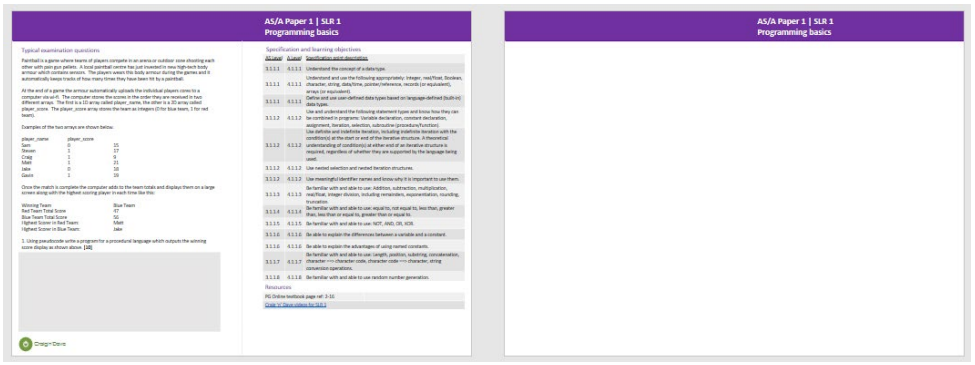
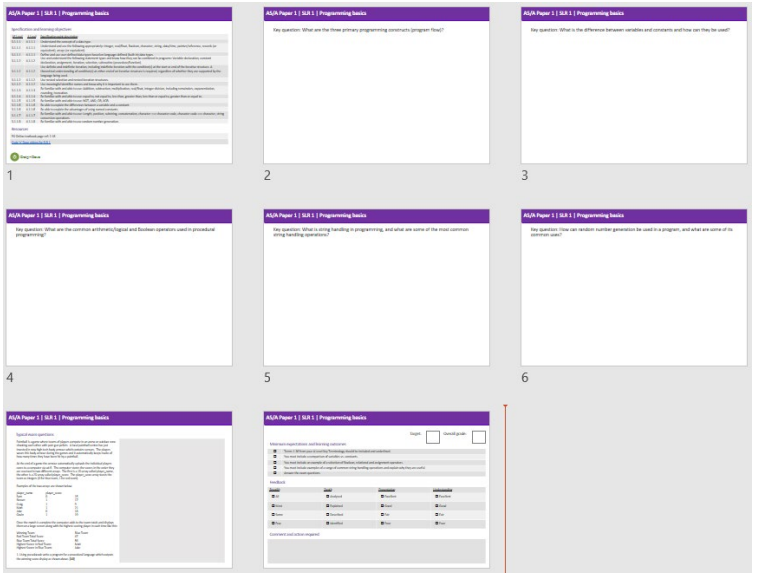
1. This lesson-by-lesson breakdown is based on a one-week calendar for the AS-Level course only. You will need to adapt it slightly to fit your school's delivery model.
2. The delivery method is flipped classroom, and homework is presented *before* the next lesson with a link to our YouTube videos hosted on student.craigndave.org.
3. A description of how a typical Craig 'n' Dave flipped classroom lessons can be structured is available here: craigndave.org/our-pedagogy/alevel-lesson/.
4. Along with the dedicated programming lessons in Year 12, students should reinforce their programming skills through regular practice in their own time. Opportunities for independent programming during lesson time are shown in green.
5. Dedicated lesson time for end-of-topic tests and student self-assessment are shown in blue.
6. Each topic in this SoL is presented as part of a Structured Learning Record, each structured learning record can be download from your premium account as a single zip file. When extracted they contain the following folders:

Contains all the activities for you to share with your students.
We often provide more activities than your students could reasonably complete in the time provided.
We constantly improve and add to our bank of activities for each SLR, so please check each year for the latest updates!
Pick and choose the most appropriate activities for your students as required.

Contains all the activities plus model answers.
For you to use as you see fit.

Contains the Structure Learning Records for your students to fill out as they carry out the activities above.
These provide your method of assessment. There is a video in this folder explaining how to get the most out of our SLRs.
Contains answers to the exam questions set in the SLRs.

7. Out structured learning records come in two formats. We would not expect a student to complete both formats, choose the one which is most appropriate for each of your students.

A3 Unscaffolded format	A4 Scaffolded format
	
• An **unscaffolded** format to allow students more freedom in how to demonstrate their knowledge and understanding in any way they see fit. • Provides minimal support on the cover page in terms of minimum expectations. • Provides an area for exam questions, assessment and feedback.	• A **scaffolded** format providing students with prompts in the form of questions which they need to answer in order to demonstrate their knowledge and understanding. • The question slides are referenced in the "Key question" column in the SoL. • Provides an area for exam questions, assessment and feedback.
YouTube Assessment with Craig'n'Dave – (AS/A Level)	

YEAR 12 - TERM 1

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
0	Introduction to the course	N/A	• Understand the course structure and appreciate how you will be taught and assessed in this subject. • Understand the importance of the flipped classroom approach.	What is "Computer Science"?	None	Introduction to programming Part 1 – Data types	
1	SLR1 Programming basics	3.1.1.1	• Understand the concept of a data type. • Understand and use the following appropriately: Integer, real/float, Boolean, character, string, date/time, pointer/reference, records (or equivalent), arrays (or equivalent) • Define and use user-defined data types based on language-defined (built-in) data types.	What are the various different common data types available in programming and what sort of data can they store? (SLR1 slide 2)	SLR1 Activities folder SLR1 Answers folder (Files starting 01 & 02)	Introduction to programming Part 2 – Basic constructs	Data type, Integer, Real/Float, Boolean, Character, String, Date/Time, Pointer/Reference, Record, Array/List, User-defined data type, Assignment, Subroutine, Sequence, Selection, Iteration, Count controlled loop, Condition controlled loop, Integer division, ==, !=, <, >, <=, >=, NOT, AND, OR, XOR Variables, Constant, String operations
2	SLR1 Programming basics	3.1.1.1 &2	• Use, understand and know how the following statement types can be combined in programs: Variable declaration, constant declaration, assignment, iteration, selection, subroutine (procedure/function) • Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure. A theoretical understanding of condition(s) at either end of an iterative structure is required, regardless of whether they are supported by the language being used. • Use nested selection and nested iteration structures. • Use meaningful identifier names and know why it is important to use them.	What are the three primary programming constructs (program flow)? (SLR1 slide 3)	SLR1 Activities folder SLR1 Answers folder (Files starting 03)	Introduction to programming Part 3 – Variables and constants	
3	SLR1 Programming basics	3.1.1.2	• Use, understand and know how the following statement types can be combined in programs: Variable declaration, constant declaration, assignment, iteration, selection, subroutine (procedure/function) • Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure. A	What is the difference between variables and constants and how can they be used? (SLR1 slide 4)	SLR1 Activities folder SLR1 Answers folder (Files starting 03)		

			theoretical understanding of condition(s) at either end of an iterative structure is required, regardless of whether they are supported by the language being used. • Use nested selection and nested iteration structures. • Use meaningful identifier names and know why it is important to use them.				
4	SLR1 Programming basics	3.1.1.2	• Use, understand and know how the following statement types can be combined in programs: Variable declaration, constant declaration, assignment, iteration, selection, subroutine (procedure/function) • Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure. A theoretical understanding of condition(s) at either end of an iterative structure is required, regardless of whether they are supported by the language being used. • Use nested selection and nested iteration structures. • Use meaningful identifier names and know why it is important to use them.	No new key questions for this lesson	SLR1 Activities folder SLR1 Answers folder (Files starting 03 & 05)	Introduction to programming Part 4 – Mathematical operators	
5	SLR1 Programming basics	3.1.1.3 &4	• Be familiar with and be able to use: Addition, subtraction, multiplication, real/float, integer division, including remainders, exponentiation, rounding, truncation • Be familiar with and be able to use: equal to, not equal to, less than, greater than, less than or equal to, greater than or equal to	What are the common arithmetic/logical and Boolean operators used in procedural programming? (SLR1 slide 5)	SLR1 Activities folder SLR1 Answers folder (Files starting 04)		
6	SLR1 Programming basics	3.1.1.5 &6	• Be familiar with and be able to use: NOT, AND, OR, XOR • Be able to explain the differences between a variable and a constant. • Be able to explain the advantages of using named constants.	No new key questions for this lesson	SLR1 Activities folder SLR1 Answers folder (Files starting 04)	Introduction to programming Part 5 – String handling	

7	SLR1 Programming basics	3.1.1.6 &7	• Be able to explain the differences between a variable and a constant. • Be able to explain the advantages of using named constants. • Be familiar with and be able to use: Length, position, substring, concatenation, character ==> character code, character code ==> character, string conversion operations	What is string handling in programming and what are some of the most common string handling operations? (SLR1 slide 6)	SLR1 Activities folder SLR1 Answers folder (Files starting 05 & 06)	Introduction to programming Part 6 – Random numbers	
8	SLR1 Programming basics	3.1.1.8	• Be familiar with and able to use random number generation.	How can random number generation be used in a program and what are some of its common uses? (SLR1 slide 7)	SLR1 Activities folder SLR1 Answers folder (Files starting 07)		
9	SLR1 Programming basics	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR1		
10	SLR1 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR1 Examination Questions (slide 8)		
11 to 18	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various		
19	SLR2 Programming next steps	3.1.1.9	• Be familiar with the concept of exception handling. • Know how to use exception handling in a programming language with which students are familiar.	How can exception handling help to improve the robustness of a program? (SLR2 slide 2)	SLR2 Activities folder SLR2 Answers folder (Files starting 01)	Subroutines, interfaces and parameters	Exception handling, Parameter, Procedure, Function, Local variable, Global variable
20	SLR2 Programming next steps	3.1.1.10	• Be familiar with subroutines and their uses. • Know that a subroutine is a named 'out of line' block of code that may be executed (called) by simply writing its name in a program statement. • Be able to explain the advantages of using subroutines in programs.	What is the difference between a procedure and a function and how do we use them to pass parameters by value? (SLR2 slide 3)	SLR2 Activities folder SLR2 Answers folder (Files starting 02)		
21	SLR2 Programming next steps	3.1.1.10 &11	• Be familiar with subroutines and their uses. • Know that a subroutine is a named 'out of line' block of code that may be executed (called) by simply writing its name in a program statement. • Be able to explain the advantages of using subroutines in programs. • Be able to describe the use of parameters to pass data within programs.	What is the difference between a procedure and a function and how do we use them to pass parameters by value? (SLR2 slide 3)	SLR2 Activities folder SLR2 Answers folder (Files starting 03)		

			• Be able to use subroutines with interfaces.				
22	SLR2 Programming next steps	3.1.1.11 &12	• Be able to describe the use of parameters to pass data within programs. • Be able to use subroutines with interfaces. • Be able to use subroutines that return values to the calling routine.	What is the difference between a procedure and a function and how do we use them to pass parameters by value? (SLR2 slide 3)	SLR2 Activities folder SLR2 Answers folder (Files starting 04)	Global and local variables	
23	SLR2 Programming next steps	3.1.1.13	• Known that subroutines may declare their own variables, called local variables, and that local variables: Exist only while the subroutine is executing, are accessible only within the subroutine • Be able to use local variables and explain why it is good practice to do so.	What is the difference between local and global variables and why are local variables preferred? (SLR2 slide 4)	SLR2 Activities folder SLR2 Answers folder (Files starting 05)		
24	SLR2 Programming next steps	3.1.1.14	• Be able to contrast local variables with global variables.	No new key questions for this lesson	SLR2 Activities folder SLR2 Answers folder (Files starting 05)		
25	SLR2 Programming next steps	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR2		
26	SLR2 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR2 Examination Questions (slide 7)		
27 to 36	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various		

YEAR 12 - TERM 2

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
						Characteristics of programming paradigms Structured approach to program design	
37	SLR3 Programming paradigms	3.1.2.1	• Understand the structured approach to program design and construction. • Be able to construct and use hierarchy charts when designing programs. • Be able to explain the advantages of the structured approach.	What do we mean by the term programming paradigm? (SLR3 slide 2) What is a hierarchy chart and how can it be used to help break down a problem? (SLR3 slide 3) What are the features of procedural programming? (SLR3 slide 4)	SLR2 Activities folder SLR2 Answers folder (Files starting 01 & 02)		Hierarchy chart, Structured approach
38	SLR3 Programming paradigms	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR3		
39	SLR3 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR3 Examination Questions (slide 6)		
40 to 42	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Data structures Arrays, records, lists and tuples	
43	SLR4 Data structures	3.2.1.1 &2	• Be familiar with the concept of data structures. • Use arrays (or equivalent) in the design of solutions to simple problems.	What are the characteristics of arrays, lists, tuples and records? (SLR4 slide 3)	SLR4 Activities folder SLR4 Answers folder (Files starting 01)	File handling	Text file, Binary (non-text file)
44	SLR4 Data structures	3.2.1.3	• Be able to read/write from/to a text file • Be able to read/write data from/to a binary(nontext) file	How do you read/write to/from a text file? (SLR4 slide 4)	SLR4 Activities folder SLR4 Answers folder (Files starting 02)		
45	SLR4 Data structures	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR4		

46	SLR4 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR4 Examination Questions (slide 16)		
47 to 54	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks*, *Programming challenges and Defold games tutorials.*		Various	Analysis and design of algorithms	
55	SLR6 Abstraction and automation	3.4.1.1	• Be able to develop solutions to simple logic problems. • Be able to check solutions to simple logic problems.	No key questions for this lesson	SLR6 Activities folder SLR6 Answers folder (Files starting 01)	Converting pseudocode to high-level code	Algorithm, Pseudocode, Test data, Abstraction, Procedural abstraction, Functional abstraction, Data abstraction, Problem abstraction, Decomposition, Composition, State transition diagram, FSM
56	SLR6 Abstraction and automation	3.4.1.2	• Understand the term algorithm. • Be able to express the solution to a simple problem as an algorithm using pseudo-code, with the standard constructs. • Be able to hand-trace algorithms. • Be able to convert an algorithm from pseudocode into high-level-language program code. • Be able to articulate how a program works, arguing for its correctness and its efficiency using logical reasoning, test data and user feedback.	What methods do we have to help us understand and express algorithms? (SLR6 slide 3)	SLR6 Activities folder SLR6 Answers folder (Files starting 02)	Hand tracing algorithms	
57	SLR6 Abstraction and automation	3.4.1.2	• Understand the term algorithm. • Be able to express the solution to a simple problem as an algorithm using pseudo-code, with the standard constructs. • Be able to hand-trace algorithms. • Be able to convert an algorithm from pseudocode into high-level-language program code. • Be able to articulate how a program works, arguing for its correctness and its efficiency using logical reasoning, test data and user feedback.	What methods do we have to help us understand and express algorithms? (SLR6 slide 3)	SLR6 Activities folder SLR6 Answers folder (Files starting 03 & 04)	The need for abstraction Nature of abstraction	
58	SLR6 Abstraction and automation	3.4.1.2 &3	• Understand the term algorithm. • Be able to express the solution to a simple problem as an algorithm using pseudo-code, with the standard constructs. • Be able to hand-trace algorithms. • Be able to convert an algorithm from pseudocode into high-level-language program code. • Be able to articulate how a program works, arguing for its correctness and its efficiency using logical reasoning, test data and user feedback.	What is meant by the term "abstraction" and how does it relate to information hiding? (SLR6 slide 4) How could abstraction be used in the production of different styles of maps? (SLR6 slide 5)	SLR6 Activities folder SLR6 Answers folder (Files starting 05 & 06)	Procedural, functional and data abstraction	

			• Be familiar with the concept of abstraction as used in computations and know that: • representational abstraction is a representation arrived at by removing unnecessary details • abstraction by generalisation or categorisation is a grouping by common characteristics to arrive at a hierarchical relationship of the 'is a kind of' type.				
59	SLR6 Abstraction and automation	3.4.1.4 &5	• Be familiar with the process of hiding all details of an object that do not contribute to its essential characteristics • Know that procedural abstraction represents a computational method.	What is meant by procedural abstraction? (SLR6 slide 6)	SLR6 Activities folder SLR6 Answers folder (Files starting 07 & 08)		
60	SLR6 Abstraction and automation	3.4.1.6 &7	• Know that for functional abstraction the particular computation method is hidden. • Know that details of how data is represented are hidden, allowing new kinds of data objects to be constructed from previously defined types.	What is meant by functional abstraction? (SLR6 slide 7) What is meant by data abstraction? (SLR6 slide 8)	SLR6 Activities folder SLR6 Answers folder (Files starting 09 & 10)	Problem decomposition Problem abstraction, reduction	
61	SLR6 Abstraction and automation	3.4.1.8 &9	• Know that details are removed until the problem is represented in a way that is possible to solve because the problem reduces to one that has already been solved. • Know that procedural decomposition means breaking a problem into several sub-problems so that each sub-problem accomplishes an identifiable task, which might itself be further subdivided.	What is meant by problem abstraction/reduction? (SLR6 slide 9)	SLR6 Activities folder SLR6 Answers folder (Files starting 11)	Composition Automation	
62	SLR6 Abstraction and automation	3.4.1.10 &11	• Know how to build a composition abstraction by combining procedures to form compound procedures. • Know how to build data abstractions by combining data objects to form compound data; for example, tree data structure. • Understand that automation requires putting models (abstraction of real-world objects/ phenomena) into action to solve problems. This is achieved by: • creating algorithms • implementing the algorithms in program code	What are decomposition and composition? (SLR6 slide 10) What is automation and how does it help us to solve problems? (SLR6 slide 11)	SLR6 Activities folder SLR6 Answers folder (Files starting 12 & 13)		

			(instructions) • implementing the models in data structures • executing the code.				
63	SLR6 Abstraction and automation	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR6		
64	SLR6 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR6 Examination Questions (slide 12)	Finite state machines	
65	SLR7 Regular and context-free languages	3.4.2.1	• Be able to draw and interpret simple state transition diagrams and state transition tables for FSMs with no output and with output (Mealy machines only).	What is a state transitions diagram and how does it allow us to represent a Finite State Machine? (SLR7 slide 3)	SLR7 Activities folder SLR7 Answers folder (Files starting 01)		
66	SLR7 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR7 Examination Questions (slide 9)		
67 to 78	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various		
79 to 84	This is the last week before Christmas. It has been left free in our delivery calendar as a buffer week.						

YEAR 12 - TERM 3

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
						Numbers	
85	SLR10 Number systems and bases	3.5.1.1 &2	• Be familiar with the concept of a natural number and the set ℕ of natural numbers (including zero). • Be familiar with the concept of an integer and the set ℤ of integers.	What are natural, rational, irrational, real and ordinal numbers? (SLR10 slide 3)	SLR10 Activities folder SLR10 Answers folder (Files starting 01)		Natural number, Rational number, Irrational number, Ordinal number, Decimal (base 10), Binary (base 2), Hexadecimal (base 16), bit, byte, Ki, Mi, Gi, Ti, k, M, G, T
86	SLR10 Number systems and bases	3.5.1.3 &4	• Be familiar with the concept of a rational number and the set ℚ of rational numbers, and that this set includes the integers. • Be familiar with the concept of an irrational number.	What are natural, rational, irrational, real and ordinal numbers? (SLR10 slide 3)	SLR10 Activities folder SLR10 Answers folder (Files starting 01)		
87	SLR10 Number systems and bases	3.5.1.5 &6	• Be familiar with the concept of a real number and the set ℝ of real numbers, which includes the natural numbers, the rational numbers and the irrational numbers. • Be familiar with the concept of ordinal numbers and their use to describe the numerical positions of objects.	What are natural, rational, irrational, real and ordinal numbers? (SLR10 slide 3)	SLR10 Activities folder SLR10 Answers folder (Files starting 01)		
88	SLR10 Number systems and bases	3.5.1.7	• Be familiar with the use of: • natural numbers for counting • real numbers for measurement.	What are natural, rational, irrational, real and ordinal numbers? (SLR10 slide 3)	SLR10 Activities folder SLR10 Answers folder (Files starting 01)	Base 2, 10 and 16 number systems	
89	SLR10 Number systems and bases	3.5.2.1	• Be familiar with the concept of a number base, in particular: • decimal (base 10) • binary (base 2) • hexadecimal (base 16). • Convert between decimal, binary and hexadecimal number bases • Be familiar with, and able to use, hexadecimal as a shorthand for binary and to understand why it is used in this way.	What is the relationship between the decimal (base 10), binary (base 2) and hexadecimal (base 16) number systems? (SLR10 slide 4)	SLR10 Activities folder SLR10 Answers folder (Files starting 02)	Bits, bytes and unit representation	
90	SLR10 Number systems and bases	3.5.3.1	• Know that: • the bit is the fundamental unit of information • a byte is a group of 8 bits. • Know that the 2n different values can be represented with n bits.	What are the various symbols, shorthand and storage values for the "bit" through to the "tebi/tera" using both the base-2 binary and base-10 decimal prefixes? (SLR10 slide 5)	SLR10 Activities folder SLR10 Answers folder (Files starting 03)		

91	SLR10 Number systems and bases	3.5.3.2	• Know that quantities of bytes can be described using binary prefixes representing powers of 2 or using decimal prefixes representing powers of 10; e.g., one kibibyte is written as 1KiB = 210 B and one kilobyte is written as 1 kB = 103 B. • Know the names, symbols and corresponding powers of 2 for the binary prefixes: • kibi, Ki - 2^{10} • mebi, Mi - 2^{20} • gibi, Gi - 2^{30} • tebi, Ti - 2^{40} • Know the names, symbols and corresponding powers of 10 for the decimal prefixes: • kilo, k - 10^{3} • mega, M - 10^{6} • giga, G - 10^{9} • tera, T - 10^{12}	What are the various symbols, shorthand and storage values for the "bit" through to the "tebi/tera" using both the base-2 binary and base-10 decimal prefixes? (SLR10 slide 5)	SLR10 Activities folder SLR10 Answers folder (Files starting 03)		
92	SLR10 Number systems and bases	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR10		
93	SL10 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR10 Examination Questions (slides 6-7)		
94 to 102	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Binary positive integers Minimum and maximum unsigned binary	
103	SLR11 Binary	3.5.4.1	• Know the difference between unsigned binary and signed binary • Know that in unsigned binary the minimum and maximum values for a given number of bits, n, are 0 and 2^{n} -1 respectively.	What is the difference between signed and unsigned binary numbers? (SLR11 slide 2)	SLR11 Activities folder SLR11 Answers folder (Files starting 01)	Unsigned binary arithmetic Two's complement	Unsigned binary, Signed binary, Two's complement, Fixed-point binary form, Floating-point binary form
104	SLR11 Binary	3.5.4.2 &3	• Be able to: • add two unsigned binary integers • multiply two unsigned binary integers • Know that signed binary can be used to represent negative integers and that one possible coding scheme is two's complement.	How do you add together two unsigned binary integers? (SLR11 slide 3) How do you multiply two unsigned binary integers? (SLR11 slide 4)	SLR10 Activities folder SLR10 Answers folder (Files starting 02)	Converting between binary, hex and decimal Floating-point representation	

			• Know how to: • represent negative and positive integers in two's complement • perform subtraction using two's complement • calculate the range of a given number of bits, n.	How do you carry out subtracting on binary numbers in twos' complement? (SLR11 slide 5)			
105	SLR11 Binary	3.5.4.3 &4	• Know how to: • represent negative and positive integers in two's complement • perform subtraction using two's complement • calculate the range of a given number of bits, n. • Know how numbers with a fractional part can be represented in: • fixed-point form in binary in a given number of bits • floating-point form in binary in a given number of bits. • Be able to convert for each representation from: • decimal to binary of a given number of bits • binary to decimal of a given number of bits.	How are numbers with a fractional part represented in binary? (SLR11 slide 6)	SLR10 Activities folder SLR10 Answers folder (Files starting 03)		
106	SLR11 Binary	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR11		
107	SLR11 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR11 Examination Questions (slide 11)	Character sets	
108	SLR12 Coding text and graphics	3.5.5.1 &2	• Differentiate between the character code representation of a decimal digit and its pure binary representation. • Describe ASCII and Unicode coding systems for coding character data and explain why Unicode was introduced.	How are character sets used to represent text? (SLR12 slide 2)	SLR12 Activities folder SLR12 Answers folder (Files starting 01)	Error checking and correction	Character coding, Character set, ASCII, Unicode, Parity bits, Majority voting, Checksums, Check digits, Bit pattern, ADC, DAC, Bitmap graphic, Resolution,
109	SLR12 Coding text and graphics	3.5.5.3	• Describe and explain the use of: • parity bits • majority voting • checksums • check digits.	How do the following error checking/correction methods work: parity bits, majority voting, checksums and check digits? (SLR12 slide 3)	SLR12 Activities folder SLR12 Answers folder (Files starting 02)	Bit patterns and data	
110	SLR12 Coding text and graphics	3.5.6.1	• Describe how bit patterns may represent other forms of data, including graphics and sound.	None for this lesson	SLR12 Activities folder SLR12 Answers folder (Files starting 03)	Bitmaps	

111	SLR12 Coding text and graphics	3.5.6.4	• Explain how bitmaps are represented.
			• Explain the following for bitmaps:
			  • resolution
			  • colour depth
			  • size in pixels.
				How are we able to store a bitmap graphic in binary and what effect does the resolution, colour depth and size in pixels have on the size of the file?
				(SLR12 slide 4)
					SLR12 Activities folder
					SLR12 Answers folder
					(Files starting 04)
						Bitmap metadata
						Calculating storage requirements for bitmaps
							Colour depth, Metadata

112	SLR12 Coding text and graphics	3.5.6.4	• Calculate storage requirements for bitmapped images and be aware that bitmap image files may also contain metadata. • Be familiar with typical metadata.	When talking about bitmap images what is metadata and how do we calculate the size of a bitmap image? (SLR12 slide 5)	SLR12 Activities folder SLR12 Answers folder (Files starting 04)		
113	SLR12 Coding text and graphics	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR12		
114	SLR12 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR12 Examination Questions (slide 8)		
115 to 120	Independent programming	N/A	Gain experience in practical programming Use our "Learning tasks", "Programming challenges" and "Programming theory PowerPoint"		Progress with learning tasks and challenges		

YEAR 12 - TERM 4

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
121-126	Term 1-3 assessment opportunity	The lessons in this first week back have been set aside to carry out a mock exam and marking with the students on the material covered in terms 1-3.					
						Analogue and digital data signals	
127	SLR13 Coding sound and music	3.5.6.2 &3	• Understand the difference between analogue and digital: • data • signals. • Describe the principles of operation of: • • an analogue to digital converter (ADC) • a digital to analogue converter (DAC).	How is digital sound represented? (SLR13 slide 3)	SLR13 Activities folder SLR13 Answers folder (Files starting 01)	Sample resolution and rate MIDI Calculating sound sample size	Sample resolution, Sampling rate, Nyquist theorem, MIDI, Lossy compression, Lossless compression, Run-length encoding, Dictionary-based encoding, Encryption, Caesar cipher, Vernam cipher
128	SLR13 Coding sound and music	3.5.6.5 &6	• Describe the digital representation of sound in terms of: • sample resolution • sampling rate and the Nyquist theorem. • Calculate sound sample sizes in bytes. • Describe the purpose of MIDI and the use of event messages in MIDI. • Describe the advantages of using MIDI files for representing music.	What is the significance of Nyquist theorem? (SLR13 slide 4) What is MIDI format for representing digital sound? (SLR13 slide 5)	SLR13 Activities folder SLR13 Answers folder (Files starting 02, 03 & 04)	Lossy vs lossless Length and dictionary coding	
129	SLR13 Coding sound and music	3.5.6.7	• Know why images and sound files are often compressed and that other files, such as text files, can also be compressed • Understand the difference between lossless and lossy compression and explain the advantages and disadvantages of each. • Explain the principles behind the following techniques for lossless compression: • run-length encoding (RLE) • dictionary-based methods.	What is the difference between lossless and lossy compression? (SLR13 slide 6)	SLR13 Activities folder SLR13 Answers folder (Files starting 05 & 06)	Encryption – Caesar cipher Encryption – Vernam cipher	
130	SLR13 Coding sound and music	3.5.6.7 &8	• Know why images and sound files are often compressed and that other files, such as text files, can also be compressed	What is run-length encoding and how does it work? (SLR13 slide 7)	SLR13 Activities folder SLR13 Answers folder (Files starting 05, 06, 07 & 08)		

			• Understand the difference between lossless and lossy compression and explain the advantages and disadvantages of each. • Explain the principles behind the following techniques for lossless compression: • run-length encoding (RLE) • dictionary-based methods. • Understand what is meant by encryption and be able to define it. • Be familiar with Caesar cipher and be able to apply it to encrypt a plaintext message and decrypt a ciphertext. Be able to explain why it is easily cracked. • Be familiar with Vernam cipher or one-time pad and be able to apply it to encrypt a plaintext message and decrypt a ciphertext. Explain why Vernam cipher is considered as a cypher with perfect security. • Compare Vernam cipher with ciphers that depend on computational security.	What is dictionary-based encoding and how does it work? (SLR13 slide 8) What are the Caesar and Vernam ciphers and how are they considered to be at the opposite ends of the encryption spectrum? (SLR13 slide 9)			
131	SLR13 Coding sound and music	3.5.6.8	• Understand what is meant by encryption and be able to define it. • Be familiar with Caesar cipher and be able to apply it to encrypt a plaintext message and decrypt a ciphertext. Be able to explain why it is easily cracked. • Be familiar with Vernam cipher or one-time pad and be able to apply it to encrypt a plaintext message and decrypt a ciphertext. Explain why Vernam cipher is considered as a cypher with perfect security. • Compare Vernam cipher with ciphers that depend on computational security.	What are the Caesar and Vernam ciphers and how are they considered to be at the opposite ends of the encryption spectrum? (SLR13 slide 9)	SLR13 Activities folder SLR13 Answers folder (Files starting 07 & 08)		
132	SLR13 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR13 Examination Questions (slide 10)	Hardware and software	
133	SLR14 Hardware and software	3.6.1.1 &2	• Understand the relationship between hardware and software and be able to define the terms: • hardware • software.	What is the difference between hardware and software? (SLR14 slide 2)	SLR14 Activities folder SLR14 Answers folder (Files starting 01 & 02)	The nature of applications	Hardware, Software, System software,

			• Explain what is meant by: • system software • application software. • Understand the need for, and attributes of, different types of software.	What is meant by the terms "system software" and "application software"? (SLR14 slide 3)		The need for operating systems Translators and utilities	Application software, Utility programs, Libraries, Translators, Compiler, Assembler, Interpreter, Operating system
134	SLR14 Hardware and software	3.6.1.2 &3	• Explain what is meant by: • system software • application software. • Understand the need for, and attributes of, different types of software. • Understand the need for, and functions of the following system software: • • operating systems (OSs) • utility programs • libraries • translators (compiler, assembler, interpreter).	What at the main functions of an operating system? (SLR14 slide 4) What are utility programs and libraries? (SLR14 slide 5)	SLR14 Activities folder SLR14 Answers folder (Files starting 03)		
135	SLR14 Hardware and software	3.6.1.4	• Understand that a role of the operating system is to hide the complexities of the hardware. • Know that the OS handles resource management, managing hardware to allocate processors, memories and I/O devices among competing processes.	How does the operating system deal with resource and memory management? (SLR14 slide 6)	SLR14 Activities folder SLR14 Answers folder (Files starting 04 & 05)		
136	SLR14 Hardware and software	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR14		
137	SLR14 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR14 Examination Questions (slide 7)	Types of programming languages Imperative languages, comparison of LLL and HLL	
138	SLR15 Programming languages and translators	3.6.2.1	• Show awareness of the development of types of programming languages and their classification into low-and high-level languages. • Know that low-level languages are considered to be: • machine-code • assembly language.	What is assembly language and how does it compare to high-level languages? (SLR15 slide 2)	SLR15 Activities folder SLR15 Answers folder (Files starting 01)		Low-level language, High-level language, Imperative high-level language,

			• Know that high-level languages include imperative high-level language. • Describe machine-code language and assembly language. • Understand the advantages and disadvantages of machine-code and assembly language programming compared with high-level language programming. • Explain the term 'imperative high-level language' and its relationship to low-level languages.				Machine-code language, Assembly language, Intermediate language, Byte code, Source code, Object (executable code)
139	SLR15 Programming languages and translators	3.6.2.1	• Show awareness of the development of types of programming languages and their classification into low-and high-level languages. • Know that low-level languages are considered to be: • machine-code • assembly language. • Know that high-level languages include imperative high-level language. • Describe machine-code language and assembly language. • Understand the advantages and disadvantages of machine-code and assembly language programming compared with high-level language programming. • Explain the term 'imperative high-level language' and its relationship to low-level languages.	What is the difference between source code and executable code? (SLR15 slide 3)	SLR15 Activities folder SLR15 Answers folder (Files starting 02)		
140	SLR15 Programming languages and translators	3.6.2.1	• Show awareness of the development of types of programming languages and their classification into low-and high-level languages. • Know that low-level languages are considered to be: • machine-code • assembly language. • Know that high-level languages include imperative high-level language. • Describe machine-code language and assembly language. • Understand the advantages and disadvantages of machine-code and assembly language	What is the difference between source code and executable code? (SLR15 slide 3)	SLR15 Activities folder SLR15 Answers folder (Files starting 03)	Translators Source code, bytecode and object executable code	

			programming compared with high-level language programming. • Explain the term 'imperative high-level language' and its relationship to low-level languages.				
141	SLR15 Programming languages and translators	3.6.3.1	• Understand the role of each of the following: • assembler • compiler • interpreter. Explain the differences between compilation and interpretation. Describe situations in which each would be appropriate. • Explain why an intermediate language such as bytecode is produced as the final output by some compilers and how it is subsequently used. • Understand the difference between source code and object (executable) code.	What are translators and how are a compiler, assembler and interpreter different? (SLR15 slide 4) What is an intermediate language and how is it used? (SLR15 slide 5)	SLR15 Activities folder SLR15 Answers folder (Files starting 04 & 05)		
142	SLR15 Programming languages and translators	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR15		
143	SLR15 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR15 Examination Questions (slide 6)	Define problems using Boolean logic Logic gates and truth tables	
144	SLR16 Logic gates and Boolean algebra	3.6.4.1	• Construct truth tables for the following logic gates: NOT, AND, OR, XOR, NAND, NOR • Be familiar with drawing and interpreting logic gate circuit diagrams involving one or more of the above gates. • Complete a truth table for a given logic gate circuit. • Write a Boolean expression for a given logic gate circuit. • Draw an equivalent logic gate circuit for a given Boolean expression.	What are the Boolean algebra logical operators and their associated logic gate symbols? (SLR16 slide 2)	SLR16 Activities folder SLR16 Answers folder (Files starting 01)	Writing Boolean expressions from logic diagrams	NAND, NOR, Boolean algebra, Truth table, Logic gate, Circuit diagram
145	SLR16 Logic gates and Boolean algebra	3.6.4.1	• Construct truth tables for the following logic gates: NOT, AND, OR, XOR, NAND, NOR	How do you translate a logic gate diagram into its associated truth table and Boolean expression and vice versa? (SLR16 slide 3)	SLR16 Activities folder SLR16 Answers folder (Files starting 02)	De Morgan's law	

			• Be familiar with drawing and interpreting logic gate circuit diagrams involving one or more of the above gates. • Complete a truth table for a given logic gate circuit. • Write a Boolean expression for a given logic gate circuit. • Draw an equivalent logic gate circuit for a given Boolean expression.				
146	SLR16 Logic gates and Boolean algebra	3.6.5.1	• Be familiar with the use of Boolean identities and De Morgan's laws to manipulate and simplify Boolean expressions.	What is De Morgan's law and how can be it used to help simplify Boolean expressions? (SLR16 slide 4)	SLR16 Activities folder SLR16 Answers folder (Files starting 05)		
147	SLR16 Logic gates and Boolean algebra	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR16		
148	SLR16 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR16 Examination Questions (slide 7)	ALU, CU, registers and buses	
149	SLR17 Internal computer architecture	3.7.1.1	• Have an understanding and knowledge of the basic internal components of a computer system. • Understand the role of the following components and how they relate to each other: processor, main memory, address bus, data bus, control bus, I/O controllers.	What are the components of a CPU, and what do they do? (SLR17 slide 4)	SLR17 Activities folder SLR17 Answers folder (Files starting 01)	Von Neumann and Harvard architecture	Processor, Main memory, Address bus, Data bus, Control bus, I/O controllers, Von Neumann architecture, Harvard architecture, ALU, Control unit, Clock, General-purpose register, PC, CIR, MAR, MBR, Status register, Fetch-execute cycle, Processor
150	SLR17 Internal computer architecture	3.7.1.1	• Understand the need for, and means of, communication between components. In particular, understand the concept of a bus and how address, data and control buses are used. • Be able to explain the difference between von Neumann and Harvard architectures and describe where each is typically used.	How does a CPU work? (SLR17 slide 5)	SLR17 Activities folder SLR17 Answers folder (Files starting 01 & 02)		
151	SLR17 Internal computer architecture	3.7.1.1	• Understand the concept of addressable memory. • Be able to describe the stored program concept: machine code instructions stored in main memory are fetched and executed serially by a processor that performs arithmetic and logical operations.	What do we mean by addressable memory? (SLR17 slide 6)	SLR17 Activities folder SLR17 Answers folder (Files starting 03)	The processor and its major components	
152	SLR17 Internal computer architecture	3.7.1.1	• Explain the role and operation of a processor and its major components:	What are the components of a CPU, and what do they do? (SLR17 slide 3)	SLR17 Activities folder SLR17 Answers folder (Files starting 01)	Fetch decode execute cycle	

			• arithmetic logic unit • control unit • clock • general-purpose registers • dedicated registers, including: • program counter • current instruction register • memory address register • memory buffer register • status register.	How does a CPU work? (SLR17 slide 4)		CISC vs RISC	instruction set, Opcode, Operand, Immediate addressing, Direct addressing, Mnemonic, Bitwise shirt, Multiple cores, Cache memory, Clock speed, Word length, Address bus width, Data bus width
153	SLR17 Internal computer architecture	3.7.3.2 &3	• Explain how the Fetch-Execute cycle is used to execute machine code programs including the stages in the cycle (fetch, decode, execute) and details of registers used. • Understand the term 'processor instruction set' and know that an instruction set is processor-specific.	How the fetch-decode-execute cycle works and how does it affect the various registers? (SLR17 slide 7)	SLR17 Activities folder SLR17 Answers folder (Files starting 01 & 04)	Opcodes and operands Addressing memory	
154	SLR17 Internal computer architecture	3.7.3.3 &4	• Know that instructions consist of an opcode and one or more operands (value, memory address or register). • Understand and apply immediate and direct address modes.	How is an instruction made up, and what are the different ways it can be addressed? (SLR17 slide 8)	SLR17 Activities folder SLR17 Answers folder (Files starting 05)	Basic machine code operations Performance of the CPU	
155	SLR17 Internal computer architecture	3.7.3.5 &6	• Understand and apply the basic machine-code operations of: • load • add • subtract • store • branching (conditional and unconditional) • compare • logical bitwise operators (AND, OR, NOT, XOR) • logical • shift right • shift left • halt. • Use the basic machine-code operations above when machine-code instructions are expressed in	What are machine code operations and what can you do with them? (SLR17 slide 9) How is the performance of a CPU determined? (SLR17 slide 10)	SLR17 Activities folder SLR17 Answers folder (Files starting 06 & 08)		

			mnemonic form- assembly language, using immediate and direct addressing. • Explain the effect on processor performance of: • multiple cores • cache memory • clock speed • word length • address bus width • data bus width.				
156	SLR17 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR17 Examination Questions (slide 12)		

YEAR 12 - TERM 5

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
						Input and output devices	
157	SLR18 Input and output devices	3.7.4.1	• Know the main characteristics, purposes and suitability of the devices and understand their principles of operation.	How are input, output and storage devices used in typical applications of Computer Science? (SLR18 slide 2)	SLR18 Activities folder SLR18 Answers folder (Files starting 01)	Magnetic, optical and flash storage Comparing capacity and speed of storage media	Input device, output device, Secondary storage, Hard disk, Optical storage, Solid-state storage
158	SLR18 Input and output devices	3.7.4.2	• Explain the need for secondary storage within a computer system. • Know the main characteristics, purposes, suitability and understand the principles of operation of the following devices: • hard disk • optical disk • solid-state disk (SSD). • the capacity and access speed of various media and make a judgement about their suitability for different applications.	Why does a computer system need secondary storage? (SLR18 slide 3) How does the following category of devices work: hard disks, optical disk and solid-state disks (SSD)? (SLR18 slide 4)	SLR18 Activities folder SLR18 Answers folder (Files starting 02, 03 & 04)		
159	SLR18 Input and output devices	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR18 (Also activity 05)		
160	SLR18 Input and output devices	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR18		
161	SLR18 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR18 Examination Questions (slide 5)	Moral, social and ethical issues part 1 Moral, social and ethical issues part 2	

						Moral, social and ethical issues part 3 Moral, social and ethical issues part 4 Moral, social and ethical issues part 5	
162	SLR19 Moral, social, legal, cultural issues	3.8.1.1	• Show awareness of current individual (moral), social (ethical), legal and cultural opportunities and risks of computing. • Understand that: •developments in computer science and digital technologies have dramatically altered the shape of communications and information •flows in societies, enabling massive transformations in the capacity to: •monitor behaviour •amass and analyse personal information •distribute, publish, communicate and disseminate personal information •computer scientists and software engineers, therefore, have power, as well as the responsibilities that go with it, in the algorithms that they devise and the code that they deploy •software and their algorithms embed moral and cultural values •the issue of scale, for software the whole world over, creates the potential for individual computer scientists and software engineers to produce great good, but with it comes the ability to cause great harm. • Be able to discuss the challenges facing legislators in the digital age.	For each of the following slides consider the topic listed at the top of the slide and answer the following questions: 1) What are the moral issues and opportunities of this topic? 2) What are the social issues and opportunities of this topic? 3) What are the ethical issues and opportunities of this topic? 4) What are the legal issues and opportunities of this topic? 5) What are the cultural issues and opportunities of this topic? (SLR19 slides 2-5)	SLR19 Activities folder SLR19 Answers folder (Files starting 00 & 01)		Moral issues, Social issues, Legal issues, Cultural issues
163	SLR19 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR19 Examination Questions (slide 6)	Data transmission basics	

164	SLR20 Communication	3.9.1.1	• Define serial and parallel transmission methods and discuss the advantages of serial over parallel transmission. • Define and compare synchronous and asynchronous data transmission. • Describe the purpose of start and stop bits in asynchronous data transmission.	How does serial and parallel data transmission work? (SLR20 slides 2) How do synchronous and asynchronous data transmission differ? (SLR20 slides 3)	SLR20 Activities folder SLR20 Answers folder (Files starting 01 & 02)	Bit rate, baud rate, bandwidth and latency	Serial transmission, Parallel transmission, Synchronous transmission, Asynchronous transmission, Start and stop bits, Baud rate, Bit rate, Bandwidth, Latency, Protocol

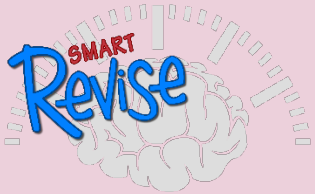
165	SLR20 Communication	3.9.9.1 &2	• Define serial and parallel transmission methods and discuss the advantages of serial over parallel transmission.
			• Define and compare synchronous and asynchronous data transmission.
			• Describe the purpose of start and stop bits in asynchronous data transmission.
			• Define:
			  • baud rate
			  • bit rate
			  • bandwidth
			  • latency
			  • protocol.
			• Differentiate between baud rate and bit rate.
			• Understand the relationship between bit rate and bandwidth.
				How do synchronous and asynchronous data transmission differ?
				(SLR20 slides 3)
				In asynchronous data transmission, what are start and stop bits used for?
				(SLR20 slides 4)
					SLR20 Activities folder
					SLR20 Answers folder
					(Files starting 02)		

166	SLR20 Communication	3.9.9.2	• Define:
			  • baud rate
			  • bit rate
			  • bandwidth
			  • latency
			  • protocol.
			• Differentiate between baud rate and bit rate.
			• Understand the relationship between bit rate and bandwidth.
				In asynchronous data transmission, what are start and stop bits used for?
				(SLR20 slides 4)
				What is the relationship between bit rate and bandwidth and what is meant by the latency of data transmission?
				(SLR20 slides 5)
					SLR20 Activities folder
					SLR20 Answers folder
					(Files starting 03)		

167	SLR20 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR20 Examination Questions (slide 6)	Networking topologies Client-sever and peer-to-peer	

168	SLR21 Network and the internet	3.9.2.1 &2	• Understand:
			  • physical star topology
			  • logical bus network topology and:
				What does a physical and logical bus network topology look like?	SLR21 Activities folder
					SLR21 Answers folder
						Introduction to Wi-Fi	Physical star topology, Logical

			• differentiate between them • explain their operation. • Explain the following and describe situations where they might be used: • peer-to-peer networking • client-server networking.	(SLR21 slides 3) Should you use a client-server or peer-to-peer network topology? (SLR21 slides 4)	(Files starting 01 & 02)		bus network topology, Peer-to-peer, Client-server, Wi-Fi, CSMA/CA, RTS/CTS, SSID
169	SLR21 Network and the internet	3.9.2.3	• Explain the purpose of Wi-Fi. • Be familiar with the components required for wireless networking. • Be familiar with how wireless networks are secured.	How does wireless networking work and how can you make it secure? (SLR21 slides 5)	SLR21 Activities folder SLR21 Answers folder (Files starting 03)	CSMA/CS and RTS/CTS	
170	SLR21 Network and the internet	3.9.2.3	• Explain the wireless protocol Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with and without Request to Send/Clear to Send (RTS/CTS). • Be familiar with the purpose of the Service Set Identifier (SSID).	How does the wireless protocol CSMA/CA with and without RTS/CTS work? (SLR21 slides 6)	SLR21 Activities folder SLR21 Answers folder (Files starting 04)		
171	SLR21 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR21 Examination Questions (slide 13)	Introduction to software development Requirements Design Implementation	
172	SLR27 Aspects of software development	3.3.1.1 &2	• Be aware that before a problem can be solved, it must be defined, the requirements of the system that solves the problem must be established and a data model created. Requirements of the system must be established by interaction with the intended users of the system. The process of clarifying requirements may involve prototyping/ agile approach. • Be aware that before constructing a solution, the solution should be designed and specified; for example, planning data structures for the data model, designing algorithms, designing an appropriate modular structure for the solution and designing the human user interface.	What are the various stages of the software development lifecycle and what happens in each one? (SLR27 slides 2)	SLR27 Activities folder SLR27 Answers folder (Files starting 01 & 02)	Testing Test strategies Test data and user feedback	Analysis, Prototyping, Agile development, Requirements specification, Design, HCI, Implementation, Testing, Normal test data, Boundary test

173	SLR27 Aspects of software development	3.3.1.3 &4	• Be aware that the models and algorithms need to be implemented in the form of data structures and code (instructions) that a computer can understand. • Be aware that the implementation must be tested for the presence of errors, using selected test data covering normal (typical), boundary and erroneous data.	When selecting test data what sort of data should you consider? (SLR27 slides 5)	SLR27 Activities folder SLR27 Answers folder (Files starting 03 & 05)	Evaluation	data, Erroneous test data, Acceptance testing, Evaluation
174	SLR27 Aspects of software development	3.3.1.5	• Know the criteria for evaluating a computer system.	Why is the evaluation stage such an important part of the software development process? (SLR27 slides 7)	SLR27 Activities folder SLR27 Answers folder (Files starting 06)		
175	SLR27 – End-of-topic test	End-of-topic test Students to self-assess and mark each other's questions to become familiar with examining mark schemes.			SLR27 Examination Questions (slide 8)		
176	Revision	This period is given over to revision. We have many resources to help with revision, including: • A dedicated FREE site for students with all our videos and downloadable cheat sheets: student.craigndave.org • A series of videos on exam technique, including how to understand command words and answer extended questions: student.craigndave.org/videos/exam-technique  We also have a dedicated revision app called Smart Revise with over a thousand questions. It has a pin-sharp focus on the specification and every single bullet point is covered. • For a summary and to share with your colleagues, visit smartrevise.craigndave.org • To get started with a free trial, visit www.smartrevise.online • To check out our overview videos, visit our YouTube channel https://youtu.be/YQDLfcy7xSM					
177	Revision						
178	Revision						
179	Revision						
180	Revision						
181	Revision						
182	Revision						
183	Revision						
184	Revision						
185	Revision						
186	Revision						