LONG-TERM OVERVIEW

YEAR 12		
Term	Topics	Assessment
1	• Introduction to the course • SLR1 Programming basics (9 lessons) • SLR2 Programming next steps (9 lessons) • SLR3 Programming paradigms (5 lessons) • SLR4 Data structures (4 lessons) • Plus 9 dedicated programming lessons	• Completed SLRs 1-3 form the basis for assessment • SLR 1-3 exam questions
2	• SLR4 Data structures (9 lessons) • SLR6 Abstraction and automation (10 lessons) • SLR7 Regular and context-free languages (6 lessons) • SLR10 Number systems and bases (8 lessons) • Plus 15 dedicated programming lessons	• Completed SLRs 4,6,7 and 10 form the basis for assessment • SLR 4,6,7 and 10 exam questions
3	• SLR11 Binary (8 lessons) • SLR12 Coding text and graphics (9 lessons) • Plus 19 dedicated programming lessons	• Completed SLRs 11 and 12 form the basis for assessment • SLR 11 and 12 exam questions
4	• SLR13 Coding sound and music (7 lessons) • SLR14 Hardware and software (4 lessons) • SLR15 Programming languages and translators (5 lessons) • SLR16 Logic gates and Boolean algebra (5 lessons) • SLR17 Internal computer architecture (6 lessons) • Plus 9 dedicated programming lessons	• Completed SLRs 13-16 form the basis for assessment • SLR 13-16 exam questions
5	• SLR17 Internal computer architecture (5 lessons) • SLR19 Moral, social, legal, cultural issues (2 lessons) • SLR20 Communication (4 lessons) • SLR27 Aspects of software development (6 lessons) • Time set aside for Year 12 Mock	• Completed SLRs 17,19,20 and 27 form the basis for assessment • SLR 17,19,20 and 27 exam questions
6	• SLR18 Input and output devices (4 lessons) • SLR21 Network and the internet (11 lessons) • Plus 27 dedicated project lessons	• Completed SLRs 18 and 21 form the basis for assessment • SLR 18 and 21 exam questions

YEAR 13		
Term	Topics	Assessment
1	• SLR5 Algorithms (7 lessons) • SLR8 Classification of algorithms (5 lessons) • SLR9 A model of computation (3 lessons) • Plus 15 dedicated project lessons	• Completed SLRs 5,8 and 9 form the basis for assessment • SLR 5,8 and 9 exam questions
2	• SLR22 TCP IP and protocols (11 lessons) • SLR23 Databases (8 lessons) • Plus 21 dedicated project lessons	• Completed SLRs 22 and 23 form the basis for assessment • SLR 22 and 23 exam questions
3	• SLR24 Big Data (3 lessons) • SLR25 Functional programming paradigms (5 lessons) • SLR26 Writing functional programs (3 lessons) • Plus 19 dedicated project lessons	• Completed SLRs 24-26 form the basis for assessment • SLR 24-26 exam questions
4	• Plus 30 dedicated project lessons •	
5	• 25 dedicated revision lessons •	

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 A-Level Linear - 1-week model (delivery calendar).xlsx" which this SoL is based on.

SHORT-TERM SCHEME OF LEARNING

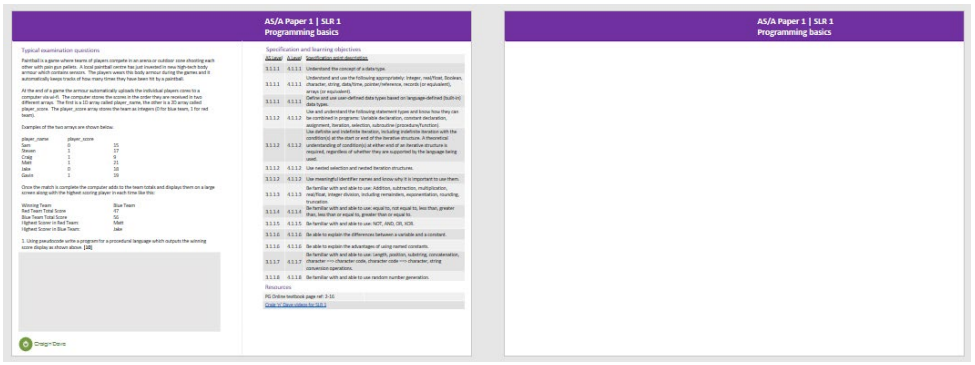
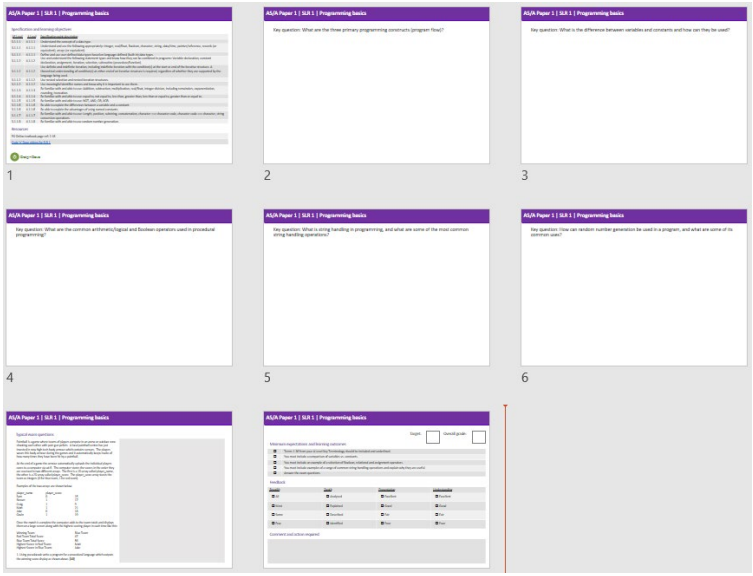
1. This lesson-by-lesson break down is based on the one-week linear calendar for the full A Level course. You will need to adapt it slightly to fit your school's delivery model.
2. The delivery method is flipped classroom, and homework is presented *before* the next lesson with a link to our YouTube 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. Dedicated lesson time set aside for the A Level project is shown in purple.
7. 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.

8. 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	4.1.1.1	• 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 types, 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, Variable, Constant, String operations
2	SLR1 Programming basics	4.1.1.1 & 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 4 – Mathematical operators	
3	SLR1 Programming basics	4.1.1.2	• 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	4.1.1.2	• 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)		
5	SLR1 Programming basics	4.1.1.3 & 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)	Introduction to programming Part 3 – Variables and constants	
6	SLR1 Programming basics	4.1.1.5 & 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	4.1.1.6 & 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	4.1.1.8	• 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 – 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)		
10 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	Exception handling	
19	SLR2 Programming next steps	4.1.1.9	• 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, Stack frame, Return address, Recursion
20	SLR2 Programming next steps	4.1.1.10	• 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	4.1.1.10 & 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. • Be able to use subroutines with interfaces.	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)		

22	SLR2 Programming next steps	4.1.1.11 & 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	4.1.1.13	• 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	4.1.1.14	• 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)	Use of stack frames with subroutines	
25	SLR2 Programming next steps	4.1.1.15	• Be able to explain how a stack frame is used with subroutine calls to store: Return address, parameters, local variables	How is a stack frame used when a program calls a subroutine? (SLR2 slide 5)	SLR2 Activities folder SLR2 Answers folder (Files starting 06)	Recursion	
26	SLR2 Programming next steps	4.1.1.16	• Be familiar with the use of recursive techniques in programming languages (general and base cases and the mechanism for implementation). • Be able to solve simple problems using recursion.	What is recursion and how does it differ from using an iterative approach? (SLR2 slide 6)	SLR2 Activities folder SLR2 Answers folder (Files starting 07)		
27	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)	Characteristics of programming paradigms Structured approach to program design	
28	SLR3 Programming paradigms	4.1.2.1	• Understand the characteristics of the procedural- and object-oriented programming paradigms and have experience of programming in each.	What do we mean by the term programming paradigm? (SLR3 slide 2) What are the features of object-oriented languages? (SLR3 slide 5)	SLR3 Activities folder SLR3 Answers folder (Files starting 01 & 03)		Procedural languages, OOP paradigms, Hierarchy chart, Structured approach, Class,

29	SLR3 Programming paradigms	4.1.2.2	• 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)	Object-oriented programming – part 1 Object-oriented programming – part 2	Object, Instantiation, Encapsulation, Inheritance, Aggregation, Composition, Polymorphism, Overriding, Class diagram
30	SLR3 Programming paradigms	4.1.2.3	• Be familiar with the concepts of: class, object, instantiation, encapsulation, inheritance, aggregation, composition, polymorphism, overriding • Know why the object-oriented paradigm is used • Be aware of the following object-oriented design principles: Encapsulate what varies, favour composition over inheritance, program to interface, not implementation • Be able to write object-oriented programs. • Be able to draw and interpret class diagrams	What are the features of object-oriented languages? (SLR3 slide 5)	03. A-Level - Object-orientated languages activities	Object-oriented programming – part 3 Object-oriented programming – part 4 Object-oriented programming – part 5	
31	SLR3 Programming paradigms	4.1.2.3	• Be familiar with the concepts of: Class, object, instantiation, encapsulation, inheritance, aggregation, composition, polymorphism, overriding • Know why the object-oriented paradigm is used • Be aware of the following object-oriented design principles: Encapsulate what varies, favour composition over inheritance, program to interface, not implementation • Be able to write object-oriented programs. • Be able to draw and interpret class diagrams	What are the features of object-oriented languages? (SLR3 slide 5)	SLR3 Activities folder SLR3 Answers folder (Files starting 03)		
32	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)	Arrays, records, lists and tuples	
33	SLR4 Data structures	4.2.1.1 & 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), Queue, Stack, Graph, Tree, Hash
34	SLR4 Data structures	4.2.1.3	• Be able to read/write from/to a text file	How do you read/write to/from a text file?	SLR4 Activities folder	Data structures	

			• Be able to read/write data from/to a binary(nontext) file.	(SLR4 slide 4)	SLR4 Answers folder (Files starting 02)	Static and dynamic data structures	table, Dictionary, Vector, Static data structure, Dynamic data structure, Linear queue, Circular queue, Priority queue, Push, Pop, Peek/top
35	SLR4 Data structures	4.2.1.4	• Be familiar with the concept of a: Queue, stack, graph, tree, hash table, dictionary, vector. • Be able to distinguish between static and dynamic structures and compare their uses, as well as explaining the advantages and disadvantages of each. • Describe the creation and maintenance of data within: queues (linear, circular, priority), stacks, hash tables.	How do the operations push and pop work with a stack stored as an array? (SLR4 slide 5) How do the operations enqueue and dequeue work with a queue stored as an array? (SLR4 slide 6) What are the uses of stacks and queues in computer science? (SLR4 slide 7)	SLR4 Activities folder SLR4 Answers folder (Files starting 04 & 05)	Stacks and Queues – Part 1 Stacks and Queues – Part 2	
36	SLR4 Data structures	4.2.2.1 & 4.2.3.1	• Be able to describe and apply the following to linear queues, circular queues and priority queues: Add an item, remove an item, test for an empty queue, test for a full queue • Be able to describe and apply the following operations: Push, pop, peek or top, test for an empty stack, test for a full stack.	How do the operations push and pop work with a stack stored as an array? (SLR4 slide 5) How do the operations enqueue and dequeue work with a queue stored as an array? (SLR4 slide 6) What are the uses of stacks and queues in computer science? (SLR4 slide 7)	SLR4 Activities folder SLR4 Answers folder (Files starting 04 & 05)		

YEAR 12 - TERM 2

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
						Graphs	
37	SLR4 Data structures	4.2.4.1	• Be aware of a graph as a data structure used to represent more complex relationships. • Be familiar with typical uses for graphs. • Be able to explain: Graph, weighted graph, vertex/node, edge/arc, undirected graph, directed graph • Know how an adjacency matrix and an adjacency list may be used to represent a graph. • Be able to compare the use of adjacency matrices and adjacency lists.	What are trees, directed or undirected graphs, and how can they be represented using other data structures? (SLR4 slide 8) What is an adjacency matrix? (SLR4 slide 9)	SLR4 Activities folder SLR4 Answers folder (Files starting 06)		Weighted graph, Vertex/node, Edge/arc, Undirected graph, Directed graph, Adjacency matrix, Adjacency list, Rooted tree, Binary tree, Tree node, Child node, Collision, Rehashing, Vector addition, Scalar-vector multiplication, Dot/scalar product
38	SLR4 Data structures	4.2.4.1	• Be aware of a graph as a data structure used to represent more complex relationships. • Be familiar with typical uses for graphs. • Be able to explain: Graph, weighted graph, vertex/node, edge/arc, undirected graph, directed graph • Know how an adjacency matrix and an adjacency list may be used to represent a graph. • Be able to compare the use of adjacency matrices and adjacency lists.	What are trees, directed or undirected graphs, and how can they be represented using other data structures? (SLR4 slide 8) What is an adjacency matrix? (SLR4 slide 9)	SLR4 Activities folder SLR4 Answers folder (Files starting 06)	Trees and Binary trees Relationship between trees and graphs	
39	SLR4 Data structures	4.2.5.1	• Know that a tree is a connected, undirected graph with no cycles. • Know that a rooted tree is a tree in which one vertex has been designated as the root. A rooted tree has parent-child relationships between nodes. The root is the only node with no parent and all other nodes are descendants of the root. • Know that a binary tree is a rooted tree in which each node has at most two children. • Be familiar with typical uses for rooted trees.	How can binary trees be visualised using arrays or objects? (SLR4 slide 10) How do you input and delete data from binary trees? (SLR4 slide 11)	SLR4 Activities folder SLR4 Answers folder (Files starting 07)	Hash tables – Part 1 Hash tables – Part 2	
40	SLR4 Data structures	4.2.6.1	• Be familiar with the concept of a hash table and its uses. • Be able to apply simple hashing algorithms. • Know what is meant by a collision and how collisions are handled using rehashing.	How do hash tables, hashing functions and overflow work? (SLR4 slide 12)	SLR4 Activities folder SLR4 Answers folder (Files starting 08)	Dictionaries	

41	SLR4 Data structures	4.2.7.1	• Be familiar with the concept of a dictionary. • Be familiar with simple applications of dictionaries; for example, information retrieval, and have experience of using a dictionary data structure in a programming language.	What is a dictionary and how can it be used? (SLR4 slide 13)	SLR4 Activities folder SLR4 Answers folder (Files starting 09)	Vectors – Part 1, Overview and representation Vectors – Part 2, Visualising vectors and maths	
42	SLR4 Data structures	4.2.8.1	• Be familiar with the concept of a vector and the following notations for specifying a vector: • [2.0, 3.14159, -1.0, 2.718281828] • 4-vector over $\mathbb{R}$ written as $\mathbb{R}4$ • function interpretation • 0 ↦ 2.0 • 1 ↦ 3.14159 • 2 ↦ -1.0 • 3 ↦ 2.718281828 • ↦ means maps to, That all the entries must be drawn from the same field; e.g., $\mathbb{R}$. • Dictionary representation of a vector. • List representation of a vector. • 1-D array representation of a vector. • Visualising a vector as an arrow.	What are vectors and how do you represent them as an arrow, 1D arrow and as an arrow? (SLR4 slide 14)	SLR4 Activities folder SLR4 Answers folder (Files starting 10)	Vectors – Part 3, Convex combination Vectors – Part 4, Dot or scalar product Vectors – Part 5, Application of dot product	
43	SLR4 Data structures	4.2.8.1	• Vector addition and scalar-vector multiplication. • Convex combination of two vectors, u and v. • Dot or scalar product of two vectors. • Applications of dot product.	What is vector addition, scalar vector multiplication, the convex combination of two vectors and dot or scale product of two vectors? (SLR4 slide 15)	SLR4 Activities folder SLR4 Answers folder (Files starting 10 & 11)		
44	SLR4 Data structures	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR4		
45	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)		
46 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	4.4.1.1	• 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)	Hand tracing algorithms Converting pseudocode to high-level code	Algorithm, Pseudocode, Test data, Abstraction, Procedural

56	SLR6 Abstraction and automation	4.4.1.2	• 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 a 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)		abstraction, Functional abstraction, Data abstraction, Problem abstraction, Decomposition, Composition, State transition diagram, FSM
57	SLR6 Abstraction and automation	4.4.1.2	• 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	4.4.1.2 & 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. • 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.	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	
59	SLR6 Abstraction and automation	4.4.1.4 & 5	• Be familiar with the process of hiding all details of an object that do not contribute to its essential characteristics	What is meant by procedural abstraction? (SLR6 slide 6)	SLR6 Activities folder SLR6 Answers folder		

			• Know that procedural abstraction represents a computational method.		(Files starting 07 & 08)		
60	SLR6 Abstraction and automation	4.4.1.6 & 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	4.4.1.8 & 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	4.4.1.10 & 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 (instructions) • implementing the models in data structures • executing the 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)		
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)		
65 &	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Finite state machines	

66							
67	SLR7 Regular and context-free languages	4.4.2.1	• 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)	Maths for regular expressions, Part 1	Regular expression, Finite sets, Countable infinite sets, Cardinality of finite set, Cartesian product of sets, Countable set, Syntax diagram
68	SLR7 Regular and context-free languages	4.4.2.2	• Be familiar with the concept of a set and the following notations for specifying a set: A = {1, 2, 3, 4, 5 } or set comprehension: A = {x | x ∈ ℕ ∧ x ≥ 1 } where A is the set consisting of those objects x such that x ∈ ℕ and x ≥ 1 is true. Know that the empty set, {}, is the set with no elements. Know that an alternative symbol for the empty set is Ø. • Be familiar with the compact representation of a set, for example, the set {0n1n | n ≥ 1}. This set contains all strings with an equal number of 0 s and 1s. • Be familiar with the concept of: • finite sets • infinite sets • countably infinite sets • cardinality of a finite set • Cartesian product of sets.	What are finite sets, infinite sets, countably infinite sets, cardinality of a finite set and the cartesian product of sets? (SLR7 slide 4)	SLR7 Activities folder SLR7 Answers folder (Files starting 02)	Maths for regular expressions, Part 2	
69	SLR7 Regular and context-free languages	4.4.2.2 & 3	• Be familiar with the meaning of the term: • subset • proper subset • countable set. • Be familiar with the set operations: • membership • union • intersection • difference. • Know that a regular expression is simply a way of describing a set and that regular expressions allow particular types of languages to be described in a convenient shorthand notation	What are subsets, proper subsets and countable sets? (SLR7 slide 5) What do the following set operations do: membership, union, intersection and difference? (SLR7 slide 6)	SLR7 Activities folder SLR7 Answers folder (Files starting 03)	Regular expression Regular expression and FSM	

			• Be able to form and use simple regular expressions for string manipulation and matching. • Be able to describe the relationship between regular expressions and FSMs. • Be able to write a regular expression to recognise the same language as a given FSM and vice versa.				
70	SLR7 Regular and context-free languages	4.4.2.3 & 4	• Know that a regular expression is simply a way of describing a set and that regular expressions allow particular types of languages to be described in a convenient shorthand notation • Be able to form and use simple regular expressions for string manipulation and matching. • Be able to describe the relationship between regular expressions and FSMs. • Be able to write a regular expression to recognise the same language as a given FSM and vice versa. • Know that a language is called regular if it can be represented by a regular expression.	What are regular expressions and how can we use them to perform simple string manipulation and pattern matching? (SLR7 slide 7)	SLR7 Activities folder SLR7 Answers folder (Files starting 04 & 05)	BNF and syntax diagrams	
71	SLR7 Regular and context-free languages	4.4.3.1	• Be able to check language syntax by referring to BNF or syntax diagrams and formulate simple production rules. • Be able to explain why BNF can represent some languages that cannot be represented using regular expressions.	What is BNF and how does it relate to syntax diagrams? (SLR7 slide 8)	SLR7 Activities folder SLR7 Answers folder (Files starting 06)		
72	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)		
73 to 76	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Numbers	
77	SLR10 Number systems and bases	4.5.1.1 & 2	• Be familiar with the concept of a natural number and the set $\mathbb{N}$ of natural numbers (including zero). • Be familiar with the concept of an integer and the set $\mathbb{Z}$ 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),
78	SLR10 Number systems and bases	4.5.1.3 & 4	• Be familiar with the concept of a rational number and the set $\mathbb{Q}$ 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)		

79	SLR10 Number systems and bases	4.5.1.5 & 6	• Be familiar with the concept of a real number and the set $\mathbb{R}$ 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)		Hexadecimal (base 16), bit, type, Ki, Mi, Gi, Ti, k, M, G, T
80	SLR10 Number systems and bases	4.5.1.7	• 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	
81	SLR10 Number systems and bases	4.5.2.1	• 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	
82	SLR10 Number systems and bases	4.5.3.1	• 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)		
83	SLR10 Number systems and bases	4.5.3.2	• 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:	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)		

			• kilo, k - 10^3 • mega, M - 10^6 • giga, G - 10^9 • tera, T - 10^{12}				
84	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 (slide 6-7)		

YEAR 12 - TERM 3

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
85 to 90	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	
91	SLR11 Binary	4.5.4.1	• 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, Normalised floating-point form, Underflow, Overflow
92	SLR11 Binary	4.5.4.2 & 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. • 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 add together two unsigned binary integers? (SLR11 slide 3) How do you multiply two unsigned binary integers? (SLR11 slide 4) How do you carry out subtracting on binary numbers in twos' complement? (SLR11 slide 5)	SLR10 Activities folder SLR10 Answers folder (Files starting 02)	Converting between binary, hex and decimal Floating-point representation	
93	SLR11 Binary	4.5.4.3 & 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)	Range and precision of binary numbers	

94	SLR11 Binary	4.5.4.5	• Know and be able to explain why both fixed-point and floating-point representation of decimal numbers may be inaccurate.	Why are both fixed-point and floating-point representation of decimal numbers sometimes inaccurate? (SLR11 slide 7)	SLR11 Activities folder SLR11 Answers folder (Files starting 04)	Rounding, absolute and relative errors in binary	
95	SLR11 Binary	4.5.4.6 & 7	• Be able to calculate the absolute error of numerical data stored and processed in computer systems. • Be able to calculate the relative error of numerical data stored and processed in computer systems. • Compare absolute and relative errors for large and small magnitude numbers, and numbers close to one. • Compare the advantages and disadvantages of fixed-point and floating-point forms in terms of range, precision and speed of calculation.	What is meant by absolute and relative error of numerical data? (SLR11 slide 8)	SLR11 Activities folder SLR11 Answers folder (Files starting 05 & 06)	Floating-point normalisation Underflow and overflow	
96	SLR11 Binary	4.5.4.8 & 9	• Know why floating-point numbers are normalised and be able to normalise un-normalised floating-point numbers with positive or negative mantissas. • Explain underflow and overflow and describe the circumstances in which they occur.	Why is there a need to normalise floating-point numbers? (SLR11 slide 9) What are underflow and overflow, and how do they occur? (SLR11 slide 10)	SLR11 Activities folder SLR11 Answers folder (Files starting 07 & 08)		
97	SLR11 Binary	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR11		
98	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)		
99 to 108	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Character sets	
109	SLR12 Coding text and graphics	4.5.5.1 & 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,

110	SLR12 Coding text and graphics	4.5.5.3	• 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	Majority voting, Checksums, Check digits, Bit pattern, Bitmap graphic, Resolution, Colour depth, Metadata, Vector graphic

111	SLR12 Coding text and graphics	4.5.6.1	• 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	

112	SLR12 Coding text and graphics	4.5.6.4	• 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	

113	SLR12 Coding text and graphics	4.5.6.4	• 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)	Vector graphics	
114	SLR12 Coding text and graphics	4.5.6.5	• Explain how vector graphics represent images using lists of objects. • Give examples of typical properties of objects. • Use vector graphic primitives to create a simple vector graphic.	What are vector images and how are they made up? (SLR12 slide 6)	SLR12 Activities folder SLR12 Answers folder (Files starting 05 & 06)	Bitmap vs Vectors	
115	SLR12 Coding text and graphics	4.5.6.6	• Compare the vector graphics approach with the bitmapped graphics approach and understand the advantages and disadvantages of each. • Be aware of the appropriate uses of each approach.	What are the advantages and disadvantages of bitmaps vs vectors? (SLR12 slide 7)	SLR12 Activities folder SLR12 Answers folder (Files starting 07)		
116	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		
117	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)		
118 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
						Analogue and digital data signals	
121	SLR13 Coding sound and music	4.5.6.2 & 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). • Know that ADCs are used with analogue sensors • Know that the most common use for a DAC is to convert a digital audio signal to an analogue signal.	How is digital sound represented? (SLR13 slide 3)	SLR13 Activities folder SLR13 Answers folder (Files starting 01)	Sample resolution and rate Calculating sound sample size	ADC, DAC, Sample resolution, Sampling rate, Nyquist theorem, MIDI, Lossy compression, Lossless compression, Run-length encoding, Dictionary-based encoding, Encryption, Caesar cipher, Vernam cipher
122	SLR13 Coding sound and music	4.5.6.7	• Describe the digital representation of sound in terms of: • sample resolution • sampling rate and the Nyquist theorem. • Calculate sound sample sizes in bytes.	What is the significance of Nyquist theorem? (SLR13 slide 4)	SLR13 Activities folder SLR13 Answers folder (Files starting 02 & 03)	MIDI	
123	SLR13 Coding sound and music	4.5.6.8	• 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 MIDI format for representing digital sound? (SLR13 slide 5)	SLR13 Activities folder SLR13 Answers folder (Files starting 04)	Lossy vs lossless	
124	SLR13 Coding sound and music	4.5.6.9	• 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.	What is the difference between lossless and lossy compression? (SLR13 slide 6)	SLR13 Activities folder SLR13 Answers folder (Files starting 05)	Length and dictionary coding	
125	SLR13 Coding sound and music	4.5.6.9	• Explain the principles behind the following techniques for lossless compression: • run-length encoding (RLE) • dictionary-based methods.	What is dictionary-based encoding and how does it work? (SLR13 slide 8)	SLR13 Activities folder SLR13 Answers folder (Files starting 06)	Encryption – Caesar cipher Encryption – Vernam cipher	

126	SLR13 Coding sound and music	4.5.6.10	• 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)		
127	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)		
128 to 132	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Hardware and software	

133	SLR14 Hardware and software	4.6.1.1 & 2	• Understand the relationship between hardware and software and be able to define the terms:
			  • hardware
			  • software.
			• Explain what is meant by:
			  • system software
			  • application software.
			• Understand the need for, and attributes of, different types of software.
				What is the difference between hardware and software?
				(SLR14 slide 2)
				What is meant by the terms "system software" and "application software"?
				(SLR14 slide 3)
					SLR14 Activities folder
					SLR14 Answers folder
					(Files starting 01 & 02)
						The nature of applications
						The need for operating systems
						Translators and utilities
							Hardware, software, System software, Application software, Utility programs, Libraries, Translators, Compiler, Assembler, Interpreter, Operating system

134	SLR14 Hardware and software	4.6.1.2 & 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	4.6.1.4	• 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 – 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)		
137 & 138	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Types of programming languages Imperative languages, comparison of LLL and HLL	
139	SLR15 Programming languages and translators	4.6.2.1	• 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 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, Machine-code language, Assembly language, Intermediate language, Byte code, Source code, Object (executable) code
140	SLR15 Programming languages and translators	4.6.2.1	• 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:	What is the difference between source code and executable code? (SLR15 slide 3)	SLR15 Activities folder SLR15 Answers folder (Files starting 02)		

			• 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.				
141	SLR15 Programming languages and translators	4.6.2.1	• 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)	03. Imperative and high-level languages activities	Translators Source code, bytecode and object executable code	
142	SLR15 Programming languages and translators	4.6.3.1	• 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.	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)		

			• Understand the difference between source code and object (executable) code.				
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 and Answers		
144	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	Define problems using Boolean logic Logic gates and truth tables	
145	SLR16 Logic gates and Boolean algebra	4.6.4.1	• 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)	01. Construct truth tables for logic gates activity	Writing Boolean expressions from logic diagrams	AND, OR, NOT, NAND, NOR, XOR, Boolean algebra, Truth table, Logic gate, Circuit diagram, Half-adder, Full-adder, D-type flip-flop, De Morgan's law
146	SLR16 Logic gates and Boolean algebra	4.6.4.1	• 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.	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)	Half and full adders D type flip flops	
147	SLR16 Logic gates and Boolean algebra	4.6.4.1	• Recognise and trace the logic of the circuits of a half-adder and a full-adder. • Construct the circuit for a half-adder. • Be familiar with the use of the edge-triggered D-type flip-flop as a memory unit.	What is the purpose of the D-type flip flop and where are they used in a computer? (SLR16 slide 5) What are the purpose and function of a half-adder and full-adder circuit? (SLR16 slide 6)	SLR16 Activities folder SLR16 Answers folder (Files starting 03 & 04)	De Morgan's law	

148	SLR16 Logic gates and Boolean algebra	4.6.5.1	• 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)		
149	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)		
150	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	ALU, CU, registers and buses	
151	SLR17 Internal computer architecture	4.7.1.1	• 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 instruction set, Opcode, Operand, Immediate addressing, Direct addressing,
152	SLR17 Internal computer architecture	4.7.1.1	• 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)		
153	SLR17 Internal computer architecture	4.7.1.1 & 4.7.2.1	• 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	
154	SLR17 Internal computer architecture	4.7.3.1	• Explain the role and operation of a processor and its major components: • 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.	What are the components of a CPU, and what do they do? (SLR17 slide 3) How does a CPU work? (SLR17 slide 4)	SLR17 Activities folder SLR17 Answers folder (Files starting 01)	Fetch decode execute cycle CISC vs RISC	

155	SLR17 Internal computer architecture	4.7.3.2	• 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.	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)	Addressing memory CISC vs RISC Opcodes and operands	Mnemonic, Bitwise shift, Interrupt, ISR, Multiple cores, Cache memory, Clock speed, Word length, Address bus width, Data bus width
156	SLR17 Internal computer architecture	4.7.3.3 & 4	• Understand the term 'processor instruction set' and know that an instruction set is processor-specific. • 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)		

YEAR 12 - TERM 5

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
						Basic machine code operations	
157	SLR17 Internal computer architecture	4.7.3.4 & 5	• Understand and apply immediate and direct address modes. • 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 mnemonic form- assembly language, using immediate and direct addressing.	What are machine code operations and what can you do with them? (SLR17 slide 9)	SLR17 Activities folder SLR17 Answers folder (Files starting 06)	Interrupts	
158	SLR17 Internal computer architecture	4.7.3.6	• Describe the role of interrupts and interrupt service routines (ISRs); their effect on the Fetch-Execute cycle; and the need to save the volatile environment while the interrupt is being serviced.	What are interrupts and what role do they play? (SLR17 slide 11)	SLR17 Activities folder SLR17 Answers folder (Files starting 07)	Performance of the CPU	
159	SLR17 Internal computer architecture	4.7.3.7	• Explain the effect on processor performance of: • multiple cores • cache memory • clock speed • word length • • address bus width • data bus width.	How is the performance of a CPU determined? (SLR17 slide 10)	SLR17 Activities folder SLR17 Answers folder (Files starting 08)		
160	SLR17 Internal computer architecture	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR17		

161	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)		
162	Independent programming	N/A	Gain experience in practical programming Use our *T.I.M.E workbooks, Programming challenges and Defold games tutorials.*		Various	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	
163	SLR19 Moral, social, legal, cultural issues	4.8.1.1	• 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	For each of the following slide 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?	SLR19 Activities folder SLR19 Answers folder (Files starting 00 & 01)		Morale issues, Social issues, Legal issues, Cultural issues

			• 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.	5) What are the cultural issues and opportunities of this topic? (SLR19 slide 2-5)			
164	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	
165	SLR20 Communication	4.9.1.1	• 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
166	SLR20 Communication	4.9.1.1 & 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)		
167	SLR20 Communication	4.9.1.2	• Define: • baud rate • bit rate • bandwidth • latency • protocol. • Differentiate between baud rate and bit rate.	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 Activities folder SLR20 Answers folder (Files starting 03)		

			• Understand the relationship between bit rate and bandwidth.	(SLR20 slides 5)			
168	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)	Introduction to software development Requirements Design Implementation	
169	SLR27 Aspects of software development	4.13.1.1 & 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 data, Erroneous test data, Acceptance testing, Evaluation
170	SLR27 Aspects of software development	4.13.1.3 & 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	
171	SLR27 Aspects of software development	4.3.1.5	• 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)	Iterative design and development	
172	SLR27 Aspects of software development	4.13.1.2 & 3	• Be aware that design can be an iterative process involving a prototyping/agile approach. • Be aware that the final solution may be arrived at using an iterative process employing	What is the agile approach to software development? (SLR27 slides 3)	SLR27 Activities folder SLR27 Answers folder (Files starting 04)		

			prototyping/ an agile approach with a focus on solving the critical path first.	What is prototyping and why is it an important part of software development? (SLR27 slides 4)			
173	SLR27 Aspects of software development	4.13.1.4	• Be aware a system should also undergo acceptance testing with the intended user(s) of the system to ensure that the intended solution meets its specification.	What is acceptance testing? (SLR27 slides 6)	SLR27 Activities folder SLR27 Answers folder (Files starting 05)		
174	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)		
175-186	This is the period has been set aside for you to set and mark a year 12 mock exam on the theory and programming concepts taught in terms 1-5.						

YEAR 12 - TERM 6

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
187 to 189	Project intro	N/A	Activities can include: • Coming up with project ideas • Group discussions on what makes a for a good group • Practising programming skills ready for the project • Working on prototypes and proof of concepts				
190 to 204	Project analysis	4.14.3.1	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The analysis section is worth 9 marks. We have a playlist on our YouTube channel which, although titled OCR A Level: Unit 3 - General guidance, has several very useful videos to help get your AQA students started: • Video 1 – General: www.youtube.com/watch?v=ZUcjAoVFYWAandlist=PLCiOXwirraUDinzjsVmpx7yof8AE-LVgd • Video 2 – Project choice: www.youtube.com/watch?v=2O7cUBGWMe0andlist=PLCiOXwirraUDinzjsVmpx7yof8AE-LVgdandindex=2 • Video 3 – Keeping it real: www.youtube.com/watch?v=GNmLyj1L5UEandlist=PLCiOXwirraUDinzjsVmpx7yof8AE-LVgdandindex=3 We provide additional support via your premium resources account under the following tile: • "NEA – The Computing Practical Project – A level Only" When providing help and feedback to students, make sure you read the guidance document from JCQ: • Title: Instructions for conducting non-examination assessments (new GCE and GCSE specifications) If you students are interested in creating a computer game of some description for their A Level project they can use this time to complete our Defold game tutorial workshops. It includes 5 full games for them to develop: • Incoming • Landers • Pong • Spotter • Worms All the above resources are available for free under your premium subscription login.				
						Input and output devices Magnetic,	

						optical and flash storage Comparing capacity and speed of storage media	
205	SLR18 Input and output devices	4.7.4.1	• 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)		Input device, Output device, Secondary storage, Hard disk, Optical storage, Solid-state storage
206	SLR18 Input and output devices	4.7.4.2	• 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)		
207	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		
208	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)	Networking topologies Client-server and peer-to-peer	
209	SLR21 Network and the internet	4.9.2.1 & 2	• Understand: • physical star topology • logical bus network topology and: • differentiate between them • explain their operation. • Explain the following and describe situations where they might be used:	What does a physical and logical bus network topology look like? (SLR21 slides 3) Should you use a client-server or peer-to-peer network topology? (SLR21 slides 4)	SLR21 Activities folder SLR21 Answers folder (Files starting 01 & 02)	Introduction to Wi-Fi	Physical star topology, Logical bus network topology, Peer-to-peer, Client-server, Wi-Fi,

			• peer-to-peer networking • client-server networking.				CSMA/CA, RTS/CTS, SSID, Packet switching, Router, Gateway, Routing, URL, FQDN, Domain name, IP address, DNS, Firewall, Packet filtering, Proxy server, Stateful packet inspection, Symmetric encryption, Asymmetric encryption, Digital certificate, Digital signature, Worms, Trojans, Virus
210	SLR21 Network and the internet	4.9.2.3,4 & 5	• 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	
211	SLR21 Network and the internet	4.9.2.6 & 7	• 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)	Structure of the Internet Packets and packet switching	
212	SLR21 Network and the internet	4.9.3.1	• Understand the structure of the Internet. • Understand the role of packet switching and routers. • Know the main components of a packet.	How does the internet work? (SLR21 slides 11)	SLR21 Activities folder SLR21 Answers folder (Files starting 00 & 05)		
213	SLR21 Network and the internet	4.9.3.1	• Define: • router • gateway. Consider where and why they are used. • Explain how routing is achieved across the Internet.	How is routing achieved across the Internet? (SLR21 slides 9) What hardware is used to connect networks? (SLR21 slides 10)	SLR21 Activities folder SLR21 Answers folder (Files starting 00 & 05)	The Internet and how it works	
214	SLR21 Network and the internet	4.9.3.1	• Describe the term 'uniform resource locator' (URL) in the context of internetworking. • Explain the terms 'fully qualified domain name' (FQDN), 'domain name' and 'IP address'. • Describe how domain names are organised. • Understand the purpose and function of the domain service and its reliance on the Domain Name Server (DNS) system. • Explain the service provided by Internet registries and why they are needed.	How does the internet work? (SLR21 slides 11)	SLR21 Activities folder SLR21 Answers folder (Files starting 00 & 06)	How firewalls work	
215	SLR21 Network and the internet	4.9.3.2	• Understand how a firewall works (packet filtering, proxy server, stateful inspection).	What are the threats to network security and how can they be mitigated? (SLR21 slides 12)	SLR21 Activities folder SLR21 Answers folder (Files starting 07)	Symmetric and asymmetric encryption	

						Digital certificates and signatures	
216	SLR21 Network and the internet	4.9.3.2	• Explain symmetric and asymmetric (private/public key) encryption and key exchange. • Explain how digital certificates and digital signatures are obtained and used.	How do digital certificates and digital signatures help to authenticate data transmission? (SLR21 slides 7) What is the difference between symmetric and asymmetric encryption? (SLR21 slides 8)	SLR21 Activities folder SLR21 Answers folder (Files starting 08)	How to address worms, trojans and viruses Network security threats	
217	SLR21 Network and the internet	4.9.3.2	• Discuss worms, trojans and viruses, and the vulnerabilities that they exploit. • Discuss how improved code quality, monitoring and protection can be used to address worms, trojans and viruses.	What are the threats to network security and how can they be mitigated? (SLR21 slides 12)	SLR21 Activities folder SLR21 Answers folder (Files starting 09)		
218	SLR21 Network and the internet	Buffer lesson	This lesson is provided as a buffer, use it as you see fit.		Any unfinished activities from SLR21		
219	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)		
220 to 228	Project analysis	4.14.3.1	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The analysis section is worth 9 marks.				

YEAR 13 - TERM 1

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
0	The first week of Year 13 has been left free for INSET days, sixth-form registration and introducing the second year of the full A Level course.						
1 to 5	Project design introduction	4.14.3.2	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The design section is worth 12 marks.				
						Graphs traversal algorithms	
6	SLR5 Algorithms	4.3.1.1	• Be able to trace breadth-first and depth-first search algorithms and describe typical applications of both.	How does the breadth-first and depth-first searches work, how are they different? (SLR5 slides 2)	SLR5 Activities folder SLR5 Answers folder (Files starting 01, 02 & 03)	Tree traversal algorithms	Breadth-first traversal, Depth-first traversal, Pre-order tree-traversal, Post-order tree-traversal, In-order tree-traversal, Infix form, RPN, Linear search, Binary search, Bubble sort, Merge sort, Dijkstra's shortest path
7	SLR5 Algorithms	4.3.2.1	• Be able to trace the tree-traversal algorithms: • pre-order • post-order • in-order. • Be able to describe the uses of tree-traversal algorithms.	How do the different tree traversal algorithms work? (SLR5 slides 3)	SLR5 Activities folder SLR5 Answers folder (Files starting 04)	Reverse Polish Notation – Part 1 Reverse Polish Notation – Part 2	
8	SLR5 Algorithms	4.3.3.1	• Be able to convert simple expressions in infix form to Reverse Polish notation (RPN) form and vice versa. Be aware of why and where it is used.	What is Reverse Polish Notation and why is it useful? (SLR5 slides 4)	SLR5 Activities folder SLR5 Answers folder (Files starting 04)	Linear search Binary search	
9	SLR5 Algorithms	4.3.4.1, 2 & 3	• Know and be able to trace and analyse the complexity of the linear search algorithm. • Know and be able to trace and analyse the time complexity of the binary search algorithm. • Be able to trace and analyse the time complexity of the binary tree search algorithm.	How does linear search work? (SLR5 slides 5) How does a binary search work and how is it different from a linear search? (SLR5 slides 6)	SLR5 Activities folder SLR5 Answers folder (Files starting 05, 06 & 07)	Bubble sort Merge sort	
10	SLR5 Algorithms	4.3.5.1 & 2	• Know and be able to trace and analyse the time complexity of the bubble sort algorithm. • Be able to trace and analyse the time complexity of the merge sort algorithm.	How does a bubble sort work? (SLR5 slides 7) How does a merge sort work and how is it different from a bubble sort? (SLR5 slides 8)	SLR5 Activities folder SLR5 Answers folder (Files starting 08 & 09)	Dijkstra's shortest path	
11	SLR5 Algorithms	4.3.6.1	• Understand and be able to trace Dijkstra's shortest path algorithm.	What is Dijkstra's shortest path algorithm and how does it work?	SLR5 Activities folder SLR5 Answers folder		

			Be aware of applications of shortest path algorithm.	(SLR5 slides 9)	(Files starting 10)		
12	SLR5 – 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.			SLR5 Examination Questions (slide 10)		
13 to 20	Project design	4.14.3.2	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The design section is worth 12 marks.				
						Algorithmic complexity, efficiency and permutation Measures to determine efficiency of algorithms	
21	SLR8 Classification of algorithms	4.4.4.1	• Understand that algorithms can be compared by expressing their complexity as a function relative to the size of the problem. Understand that the size of the problem is the key issue. • Understand that some algorithms are more efficient: • time-wise than other algorithms • space-wise than other algorithms.	Why are execution time and space an important consideration when talking about the suitability of different algorithms? (SLR8 slides 3)	SLR8 Activities folder SLR8 Answers folder (Files starting 01)	Comparison of the complexity of algorithms Big O notation in practice	Constant time, Logarithmic time, Linear time, Polynomial time, Exponential time, Time-complexity, Tractable algorithms, Intractable algorithms, Hating problem
22	SLR8 Classification of algorithms	4.4.4.2 & 3	• Be familiar with the mathematical concept of a function as a mapping from one set of values, the domain, to another set of values, drawn from the co-domain; for example, $\mathbb{N} \rightarrow \mathbb{N}$. • Be familiar with the concept of: • a linear function; for example, y = 2x • a polynomial function; for example, y = 2x2 • an exponential function; for example, y = 2x • a logarithmic function; for example, y = log10 x. • Be familiar with the notion of permutation of a set of objects or values, for example, the letters of a word and that the number of permutations of n distinct objects is n factorial (n!). • Be familiar with Big-O notation to express time complexity and be able to apply it to cases where	What is meant by Big O notation? (SLR8 slides 4) How can we use Big O notation to compare the complexity of different algorithms? (SLR8 slides 5)	SLR8 Activities folder SLR8 Answers folder (Files starting 02 & 03)	Limits of algorithms	

			the running time requirements of the algorithm grow in: constant time, logarithmic time, linear time, polynomial time, exponential time. • Be able to derive the time complexity of an algorithm				
23	SLR8 Classification of algorithms	4.4.4.4 & 5	• Be aware that algorithmic complexity and hardware impose limits on what can be computed. • Know that algorithms may be classified as being either: • tractable – problems that have a polynomial (or less) time solution are called tractable problems. • intractable – problems that have no polynomial (or less) time solution are called intractable problems.	What are some of the limits on computation? (SLR8 slides 6)	SLR8 Activities folder SLR8 Answers folder (Files starting 04)	The Halting problem	
24	SLR8 Classification of algorithms	4.4.4.6 & 7	• Be aware that some problems cannot be solved algorithmically. • Describe the Halting problem (but not prove it), that is the unsolvable problem of determining whether any program will eventually stop if given a particular input. • Understand the significance of the Halting problem for computation.	What is the difference between computable and non-computable problems? (SLR8 slides 7) What is the Halting problem? (SLR8 slides 8)	SLR8 Activities folder SLR8 Answers folder (Files starting 05)		
25	SLR8 – 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.			SLR8 Examination Questions (slide 9-10)	The Turing machine	
26	SLR9 A model of computation	4.4.5.1 & 2	• Be familiar with the structure and use of Turing machines that perform simple computations. • Know that a Turing machine can be viewed as a computer with a single fixed program, expressed using: • a finite set of states in a state transition diagram • a finite alphabet of symbols • an infinite tape with marked-off squares • a sensing read-write head that can travel along the tape, one square at a time. One of the states is called a start state and states that have no outgoing transitions are called halting states.	What is a Turing machine and how does it operate? (SLR9 slides 2)	SLR9 Activities folder SLR9 Answers folder (Files starting 01)	Transition functions and state transition diagrams The importance of the Universal Turing machine	Turing machine, Transition function, State transition diagram, Universal Turing machine

27	SLR9 A model of computation	4.4.5.3, 4 & 5	• Understand the equivalence between a transition function and a state transition diagram. • Be able to: • represent transition rules using a transition function • represent transition rules using a state transition diagram • hand-trace simple Turing machines. • Be able to explain the importance of Turing machines and the Universal Turing machine to the subject of computation.	What is the relationship between a transition function and a state transition diagram? (SLR9 slides 3) What is the importance of the Universal Turing Machine to the subject of computation? (SLR9 slides 4)	SLR9 Activities folder SLR9 Answers folder (Files starting 02 & 03)		
28	SLR9 – 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.			SLR9 Examination Questions (slide 5)		
29 & 30	Project design	4.14.3.2	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The design section is worth 12 marks.				

YEAR 13 - TERM 2

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
						The TCP IP stack and MAC addresses Well known ports	
31	SLR22 TCP IP and protocols	4.9.4.1	• Describe the role of the four layers of the TCP/IP stack (application, transport, network, link). • Describe the role of sockets in the TCP/IP stack. • Be familiar with the role of MAC (Media Access Control) addresses. • Explain what the well-known ports and client ports are used for and the differences between them.	What are the features of a protocol? (SLR22 slides 3) How does the TCP/IP protocol work? (SLR22 slides 4)	SLR22 Activities folder SLR22 Answers folder (Files starting 01 & 02)	Introduction to protocols FTP	TCP/IP stack, Application layer, Transport layer, Network layer, Link layer, MAC, FTP, HTTP, HTTPA, POP3, SMPT, SSH, Email server, Web server, Subnets, Subnet masks, IPv4, IPv6, DHCP, NAT, Port forwarding, WebSocket, CRUD, REST, JSON, XML, Thin-client computing, Thick-client computing
32	SLR22 TCP IP and protocols	4.9.4.1 & 2	• Be familiar with the following protocols: • FTP (File Transfer Protocol) • HTTP (Hypertext Transfer Protocol) • HTTPS (Hypertext Transfer Protocol Secure) • POP3 (Post Office Protocol (v3)) • SMTP (Simple Mail Transfer Protocol) • SSH (Secure Shell). • Be familiar with FTP client software and an FTP server, regarding transferring files using anonymous and non-anonymous access.	How do the following protocols work: FTP, HTTP(S), POP3, SMTP, SSH? (SLR22 slides 5)	SLR22 Activities folder SLR22 Answers folder (Files starting 03 & 04)	SSH	
33	SLR22 TCP IP and protocols	4.9.4.2	• Be familiar with how SSH is used for remote management. • Know how an SSH client is used to make a TCP connection to a remote port and send commands to this port using application-level protocols such as GET for HTTP, SMTP commands for sending email and POP3 for retrieving email. • Be familiar with using SSH to log in securely to a remote computer and execute commands.	How do the following protocols work: FTP, HTTP(S), POP3, SMTP, SSH? (SLR22 slides 5)	SLR22 Activities folder SLR22 Answers folder (Files starting 05)	Email servers	
34	SLR22 TCP IP and protocols	4.9.4.2	• Explain the role of an email server in retrieving and sending email.	How do email and webservers work? (SLR22 slides 6)	SLR22 Activities folder SLR22 Answers folder (Files starting 06)	Web servers and web browsers The structure of IP addresses	

35	SLR22 TCP IP and protocols	4.9.4.2 & 3	• Explain the role of a web server in serving up web pages in text form. • web pages and web page resources and rendering these accordingly. • Know that an IP address is split into a network identifier part and a host identifier part.	How do email and webservers work? (SLR22 slides 6)	SLR22 Activities folder SLR22 Answers folder (Files starting 06 & 07)	IPv4 and IPv6 Routable and non-routable IP addresses	
36	SLR22 TCP IP and protocols	4.9.4.4 & 5	• Know that networks can be divided into subnets and know how a subnet mask is used to identify the network identifier part of the IP address. • Know that there are currently two standards of IP address, v4 and v6. • Know why v6 was introduced.	What is IPv4 and why is there a need for IPv6? (SLR22 slides 7)	SLR22 Activities folder SLR22 Answers folder (Files starting 07 & 08)	DHCP NAT and port forwarding	
37	SLR22 TCP IP and protocols	4.9.4.6, 7 & 8	• Distinguish between routable and non-routable IP addresses. • Understand the purpose and function of the DHCP system. • Explain the basic concept of NAT and why it is used.	How do DHCP and NAT work and why are they needed? (SLR22 slides 8)	SLR22 Activities folder SLR22 Answers folder (Files starting 00 & 07)	The client-server model	
38	SLR22 TCP IP and protocols	4.9.4.9 & 10	• Explain the basic concept of port forwarding and why it is used. • Be familiar with the client-server model.	What is port forwarding and how does it work? (SLR22 slides 9) What is the client-server model (SLR22 slides 10)	SLR22 Activities folder SLR22 Answers folder (Files starting 07 & 09)	The WebSocket protocol Web CRUD and applications at REST JSON and XML	
39	SLR22 TCP IP and protocols	4.9.4.10	• Be familiar with the WebSocket protocol and know why it is used and where it is used. • Be familiar with the principles of Web CRUD Applications and REST: • CRUD is an acronym for: • C – Create • R – Retrieve • U – Update • D – Delete. • REST enables CRUD to be mapped to database functions (SQL) as follows: • GET → SELECT • POST → INSERT • DELETE → DELETE • PUT → UPDATE. • Compare JSON (Java script object notation) with XML.	What are the principles of Web CRUD Applications and how is this associated with "REST"? (SLR22 slides 12) What are the advantages of JSON over XML? (SLR22 slides 13)	SLR22 Activities folder SLR22 Answers folder (Files starting 10,11 & 12)	Thin client and thick client computing	

40	SLR22 TCP IP and protocols	4.9.4.11	• Compare and contrast thin-client computing with thick-client computing.	What are the advantages and disadvantages of thin vs thick-client computing? (SLR22 slides 11)	SLR22 Activities folder SLR22 Answers folder (Files starting 13)		
41	SLR22 – 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.			SLR22 Examination Questions (slide 14-15)		
42 to 60	Project technical solution intro	4.14.3.3	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The technical solution section is worth 42 marks.				
						Databases	
61	SLR23 Databases	4.10.1.1	• Produce a data model from given data requirements for a simple scenario involving multiple entities. • Produce entity-relationship diagrams representing a data model and entity descriptions in the form: Entity1 (Attribute1, Attribute2, ...).	What are the key terms associated with databases? (SLR23 slides 2)	SLR23 Activities folder SLR23 Answers folder (Files starting 00 & 01)		ERD, Attribute, Primary key, Composite primary key, Foreign key, Normalisation, 0NF, 1NF, 2NF, 3NF, SQL, Relational database

62	SLR23 Databases	4.10.2.2	• Explain the concept of a relational database.
			• Be able to define the terms:
			  • attribute
			  • primary key
			  • composite primary key
			  • foreign key.
				What are the key terms associated with databases?
				(SLR23 slides 2)
				Can you explain the concepts of a relational database?
				(SLR23 slides 3)
					SLR23 Activities folder
					SLR23 Answers folder
					(Files starting 00 & 01)		

63	SLR23 Databases	4.10.2.2	• Explain the concept of a relational database.
			• Be able to define the terms:
			  • attribute
			  • primary key
			  • composite primary key
			  • foreign key.
				What are the key terms associated with databases?
				(SLR23 slides 2)
				Can you explain the concepts of a relational database?
				(SLR23 slides 3)
					SLR23 Activities folder
					SLR23 Answers folder
					(Files starting 00 & 01)
						Normalisation to 3NF	

64	SLR23 Databases	4.10.3.3	• Normalise relations to third normal form. • Understand why databases are normalised.	What is the purpose of normalisation? (SLR23 slides 4)	SLR23 Activities folder SLR23 Answers folder (Files starting 02)		
65	SLR23 Databases	4.10.3.3	• Normalise relations to third normal form. • Understand why databases are normalised.	What is the purpose of normalisation? (SLR23 slides 4)	SLR23 Activities folder SLR23 Answers folder (Files starting 02)	SQL	

66	SLR23 Databases	4.10.4.1	• Be able to use SQL to retrieve, update, insert and delete data from multiple tables of a relational database. • Be able to use SQL to define a database table.	How do you use the main keywords in SQL to create, return and delete data in a database? (SLR23 slides 5)	SLR23 Activities folder SLR23 Answers folder (Files starting 03)	Transaction processing	
67	SLR23 Databases	4.10.5.1	• Know that a client-server database system provides simultaneous access to the database for multiple clients. • Know how concurrent access can be controlled to preserve the integrity of the database.	How do databases seal with concurrent access and make sure that it preserves data integrity? (SLR23 slides 6)	SLR23 Activities folder SLR23 Answers folder (Files starting 04)		
68	SLR23 – 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.			SLR23 Examination Questions (slide 7-9)		
69 & 70	Technical solution	4.14.3.3	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The technical solution section is worth 42 marks.				

YEAR 13 - TERM 3

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
71 to 75	Technical solution	4.14.3.3	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The technical solution section is worth 42 marks.				
						What is Big Data Big Data and functional programming	
76	SLR24 Big Data	4.11.1.1	• Know that 'Big Data' is a catch-all term for data that won't fit the usual containers. Big Data can be described in terms of: • volume – too big to fit into a single server • velocity – streaming data, milliseconds to seconds to respond • variety – data in many forms such as structured, unstructured, text, multimedia. • Know that when data sizes are so big as not to fit on to a single server: • the processing must be distributed across more than one machine • functional programming is a solution because it makes it easier to write correct and efficient distributed code. Know what features of functional programming make it easier to write: • correct code • code that can be distributed to run across more than one server.	What is Big Data? (SLR24 slides 2) How do we handle Big Data given that it can't fit onto a single server? (SLR24 slides 3)	SLR24 Activities folder SLR24 Answers folder (Files starting 01 & 02)	Representing Big Data	Big Data, Fact-based model, Graph schema
77	SLR24 Big Data	4.11.1.1	• Be familiar with the: • fact-based model for representing data • graph schema for capturing the structure of the dataset • nodes, edges and properties in graph schema.	How does the fact-based model allow us to represent Big Data? (SLR24 slides 4) How does a graph scheme allow us to represent Big Data? (SLR24 slides 5)	SLR24 Activities folder SLR24 Answers folder (Files starting 03)		

78	SLR24 – 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.			SLR24 Examination Questions (slide 6)	
79 to 90	Technical solution	4.14.3.3	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The technical solution section is worth 42 marks.			
					Basics of functional programming	
91	SLR25 Functional programming paradigms	4.12.1.1 & 2	• Know that a function, f, has a function type f: A → B (where the type is A → B, A is the argument type, and B is the result type). Know that A is called the domain and B is called the co-domain. Know that the domain and co-domain are always subsets of objects in some data type. • Know that a function is a first-class object in functional programming languages and in imperative programming languages that support such objects. This means that it can be an argument to another function as well as the result of a function call.	What is functional programming, and how is it different from other programming paradigms? (SLR25 slides 2)	SLR25 Activities folder SLR25 Answers folder (Files starting 00, 01 & 02)	Function and partial function application	Domain, Co-domain, First-class object, Function application, Partial function application, Composition of functions
92	SLR25 Functional programming paradigms	4.12.1.3	• Know that function application means a function applied to its arguments.	What are the key terms associated with functional programming? (SLR25 slides 3)	SLR25 Activities folder SLR25 Answers folder (Files starting 00, 01 & 02)		
93	SLR25 Functional programming paradigms	4.12.1.4	• Know what is meant by partial function application for one, two and three-argument functions and be able to use the notations shown opposite.	What is partial function application and how does it work? (SLR25 slides 4)	SLR25 Activities folder SLR25 Answers folder (Files starting 00, 01 & 02, 03)	Composition of functions	
94	SLR25 Functional programming paradigms	4.12.1.5	• Know what is meant by the composition of functions.	What is meant by the composition of functions? (SLR25 slides 5)	SLR25 Activities folder SLR25 Answers folder (Files starting 00, 01 & 02, 04)		
95	SLR25 – 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.			SLR25 Examination Questions (slide 6)	Higher-order functions	

96	SLR26 Writing functional programs	4.12.2.1	• Show experience of constructing simple programs in a functional programming language. • Higher-order functions. • Have experience of using the following in a functional programming language: • map • filter • reduce or fold.	What is a higher-order function? (SLR26 slides 2) How do map, filter and reduce or fold work? Provide examples. (SLR26 slides 3)	SLR26 Activities folder SLR26 Answers folder (Files starting 00, 01 & 02)	List operations in functional programming Introduction to Haskell	Higher-order functions, Map, Filter, Reduce or fold
97	SLR26 Writing functional programs	4.12.3.1	• Be familiar with representing a list as a concatenation of a head and a tail. • Know that the head is an element of a list and the tail is a list. • Know that a list can be empty. • Describe and apply the following operations: • return head of list • return tail of list • test for an empty list • return length of list • construct an empty list • prepend an item to a list • append an item to a list. • Have experience writing programs for the list operations mentioned above in a functional programming language or a language with support for the functional paradigm.	How does a list work in functional programming? Provide examples. (SLR26 slides 4)	SLR26 Activities folder SLR26 Answers folder (Files starting 00, 01 & 03)		
98	SLR26 – 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.			SLR26 Examination Questions (slide 5)		
99 & 100	Technical solution	4.14.3.3	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The technical solution section is worth 42 marks.				

YEAR 13 - TERM 4

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
101 to 115	Technical solution	4.14.3.3	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The technical solution section is worth 42 marks.				
116 to 130	Project evaluation	4.14.3.6	These lessons are given over to the project. This is marked out of 75 and worth 20% of the full A Level qualification. The evaluation section is worth 4 marks.				

YEAR 13 - TERM 5

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
131	Revision						
132	Revision						
133	Revision						
134	Revision						
135	Revision						
136	Revision						
137	Revision						
138	Revision						
139	Revision						
140	Revision						
141	Revision						
142	Revision						
143	Revision						
144	Revision						
145	Revision						
146	Revision						
147	Revision						
148	Revision						
149	Revision						
150	Revision						
151	Revision						
152	Revision						
153	Revision						
154	Revision						
155	Revision						

This period is given over to revision.

We have many resources to help with revision.

We have a dedicated FREE site for students with all our videos and downloadable cheat sheets: student.craigndave.org

We have 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 which has a full coverage of the AQA AS/A Level course.

It will have a pin-sharp focus on the specification and every single bullet point 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: www.youtube.com/watch?v=XqJZNTZNa9Mandlist=PLCiOXwirraUBSzKbl4TaGvXkhP6NkzVgi
-

YEAR 13 - TERM 6

	Topic Focus	Spec ref	Specification Learning Outcomes	Key question	Activities	HW for next lesson	Key Terms
156	A Level Exam period	This is the period of A Level exams.					
157	A Level Exam period						
158	A Level Exam period	Most Year 13 students are on study leave during this period, so it has been left blank.					
159	A Level Exam period						
160	A Level Exam period						
161	A Level Exam period						
162	A Level Exam period						
163	A Level Exam period						
164	A Level Exam period						
165	A Level Exam period						
166	A Level Exam period						
167	A Level Exam period						
168	A Level Exam period						
169	A Level Exam period						
170	A Level Exam period						
171	A Level Exam period						
172	A Level Exam period						
173	A Level Exam period						
174	A Level Exam period						
175	A Level Exam period						