

## YEAR 12 CURRICULUM PLAN

Term one	Monday 4 September to Friday 20 October 2023 (7 Weeks)		
WEEK Starting:	TUESDAY LESSONS (UNIT1 COMPUTER SYSTEMS)	THURSDAY LESSONS (UNIT 2 ALGORITHMS AND PROGRAMMING )	FRIDAY LESSONS (PRACTICALS)
4th Sept	Introduction to course, admin work, recap on GCSE topics	Introduction to course, admin work, recap on GCSE topics	Introduction to course, admin work, recap on GCSE topics
11th Sept	1.1.3. a)How different input output and storage devices can be applied to the solution of different problems	2.1.2 Thinking ahead (introduced)	Simple problem solving tasks (introduction to programming, including blocks)
18th Sept	b)The uses of magnetic, flash and optical storage device	2.2.1 Introduction to the e) Use of an IDE to develop/debug a program (Register with Repl.it)	Procedural/Imperative language IDE of Centre's choice
25th Sept	c)RAM and ROM d) Virtual storage		<b>BASELINE ASSESSMENT (Comp 01 Only)</b>
2nd Oct	1.2.3. Writing and following algorithms	2.2.1 b) Programming constructs: sequence, iteration, branching	Programming exercises involving sequence
9th Oct	1.1.1 a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR): How this relates to assembly language programs b) The Fetch-Decode-Execute Cycle	2.1.4 Thinking logically (introduced) 2.2.1 a) Programming constructs: sequence, iteration, branching	Programming exercises involving branching (IF, nested IF, SELECT/CASE)
16th Oct	1.2.1 a) The function and purpose of operating systems	2.2.1 b) Programming constructs: sequence, iteration, branching	Programming exercises involving iteration (FOR, WHILE, REPEAT)
<b>HALF TERM</b>	<b>Monday 23 October to Friday 27 October 2023.</b>	<b>Monday 23 October to Friday 27 October 2023.</b>	<b>Monday 23 October to Friday 27 October 2023.</b>
Term Two	Monday 30 October to Thursday 21 December 2023 (8 Weeks)		
30th Oct	1.2.1  b) Distributed, Embedded, Multi-tasking, Multi-user and Real Time operating systems c) BIOS d) Device driver	2.2.1 c) Programming constructs: sequence, iteration, branching	Programming exercises involving iteration (FOR, WHILE, REPEAT)
6th Nov	1.2.4 a) Procedural languages 1.4.1 a) Represent positive integers in binary	2.2.1 b) Recursion, how it can be used and compares to an iterative approach	Programming exercises demonstrating recursion (eg factorial)
13th Nov	b) Use of Sign and Magnitude and Two's Complement to represent negative numbers in binary	2.1.3 Thinking Procedurally (introduced) 2.2.1 c) Global and local variables.	Programming exercises involving functions, procedures and parameters
20th Nov	c) Addition and subtraction of binary integers d) Represent positive integers in hexadecima e) Representation and normalisation of floating point numbers in binary f) Floating point arithmetic, positive and negative numbers, addition and subtraction	d) Modularity, functions and procedures, parameter passing by value and by reference 2.2.2 a) Features that make a problem solvable by computational methods b) Problem Recognition	Programming exercises involving functions, procedures and parameters
27th Nov	g) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR h) How character sets (ASCII and UNICODE) are used to represent tex	c) Problem Decomposition d) Use of divide and conquer	
4th Dec	1.4.2 a) Arrays (of up to 2 dimensions)	1.4.2 a) Arrays (of up to 2 dimensions)	Programming using 2D arrays
11th Dec	1.4.2 b) The following structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table	1.2.3 a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development	Programming exercises including the algorithms for the main data structures
18th Dec <b>CHRISTMAS HOLIDAY</b>	<b>Friday 22 December 2023 to Friday 5 January 2024</b>	<b>Friday 22 December 2023 to Friday 5 January 2024</b>	<b>Friday 22 December 2023 to Friday 5 January 2024</b>

Missed Lessons

Mocks on 5th December

Term Three	Monday 8 January to Friday 9 February 2024. (5 Weeks)	Monday 8 January to Friday 9 February 2024. (5 Weeks)	Monday 8 January to Friday 9 February 2024. (5 Weeks)
8th Jan	c) How to create, traverse, add data to and remove data from the data structures mentioned above 1.4.3	2.3.1 a) Analysis and design of algorithms for a given situation	
15th Jan	a) Define problems using Boolean logic	b) The suitability of different algorithms for a given task and data set, in terms of execution time and space	
22nd Jan	b) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation 1.2.4	c) Algorithms for the main data structures, (Stacks, queues, trees, linked lists, depth-first (post-order) and breadth-first traversal of trees)	
29th Jan	b) Assembly language (including following and writing simple programs with the Little Man Computer instruction set)	d) Standard algorithms (Bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search)	
5th Feb	c) Modes of addressing memory (immediate, direct, indirect and indexed)		
<b>HALF TERM</b>	<b>Monday 12 February to Friday 16 February 2024.</b>	<b>Monday 12 February to Friday 16 February 2024.</b>	<b>Monday 12 February to Friday 16 February 2024.</b>
Term Four	Monday 19 February to Thursday 28 March 2024 (6 Weeks)	Monday 19 February to Thursday 28 March 2024 (6 Weeks)	Monday 19 February to Thursday 28 March 2024 (6 Weeks)
19th Feb	1.3.1 a) Lossy vs Lossless compression b) Run Length Encoding and dictionary coding for lossless compression	1.2.4 b) Assembly language (including following and writing simple programs with the Little Man Computer instruction set)	
26th Feb	c) Symmetric and asymmetric encryption d) Different uses of hashing 1.3.3	1.2.4 (continue) b) Assembly language (including following and writing simple programs with the Little Man Computer instruction set)	
4th Mar	a) The TCP/IP Stack b) Protocol layering c) LANs and WAN	1.3.2 a) Relational database, flat file, primary key, foreign key, secondary key, normalisation and indexing b) Normalisation to 3NF c) SQL - Interpret and modify (list of key words) d) Referential Integrity	
11th Mar	d) Packet and circuit switching e) Protocols f) Client-server and Peer to peer 1.3.4	e) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy	
18th Mar	a) HTML, CSS and JavaScript b) Search engine indexing	1.3.4 a) HTML, CSS and JavaScript	
25th Mar	c) PageRank Algorithm d) Server and client side processing	b) Search engine indexing c) PageRank Algorithm d) Server and client side processing	Practical HTML, CSS, JavaScript exercises
<b>EASTER HOLIDAY</b>	<b>Friday 29 March to Friday 12 April 2024.</b>	<b>Friday 29 March to Friday 12 April 2024.</b>	<b>Friday 29 March to Friday 12 April 2024.</b>
Term Five	Monday 15 April to Friday 24 May 2024 (6 weeks)	Monday 15 April to Friday 24 May 2024 (6 weeks)	Monday 15 April to Friday 24 May 2024 (6 weeks)
15th Apr	1.5.1 a) Data Protection Act b) Computer Misuse Act c) Copyright and Patents Act d) Regulation of Investigatory Powers Act .2.4	1.5.2 These include but are not limited to: a) Computers in the workforce b) Automated decision making	
22nd Apr	d) Object-oriented languages (using Java/C++ style pseudocode) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism	c) Artificial intelligence d) Environmental effects e) Censorship and the Internet	

29th Apr	.2.4 (continue) d) Object-oriented languages (using Java/C++ style pseudocode) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism	2.1.1 Thinking abstractly (introduced)	Practical OO pseudocode exercises
6th May (Bank Holiday)	1.1.2 a) The differences between and uses of CISC and RISC processors b) GPUs and their uses (including those not related to graphics) c) Multicore and Parallel systems	1.2.4 d) Object-oriented languages (using Java/C++ style pseudocode) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism	
13th May	1.1.1 c) The use of pipelining in a processor to improve efficiency	2.2.2 e) Use of abstraction	
20th May	1.2.1 b) Memory Management (paging, segmentation and virtual memory)) Interrupts d) Scheduling: Round Robin, First come first served, Multi-level feedback queues, shortest job first and shortest remaining time h) Virtual Machines	f) Candidates should apply their knowledge of • backtracking • data mining • heuristics	
<b>HALF TERM</b>	Monday 27 May to Friday 31 May 2024	Monday 27 May to Friday 31 May 2024	Monday 27 May to Friday 31 May 2024
Term Six	Monday 3 June to Wednesday 24 July 2024. (7 Weeks 3 days)	Monday 3 June to Wednesday 24 July 2024. (7 Weeks 3 days)	Monday 3 June to Wednesday 24 July 2024. (7 Weeks 3 days)
	1.2.2 d) Translators: Interpreters, compilers and assemblers e) Stages of compilation (Lexical Analysis, Syntax Analysis, Code Generation and Optimisation) f) Linkers and loaders	• performance modelling • pipelining • visualisation to solving problems	PRACTICALS PROJECT
3rd June		2.1.5 Thinking Concurrently (introduction)	3.1 Analysis
10th June			3.1.1 Problem identification
17th June			3.1.2 Stakeholders
24th June			3.1.3 Research the problem
1st July			3.1.4 Specify the proposed solution
8th July			3.2 Design
			3.2.1 Decompose the problem
			3.2.2 Describe the solution
			3.3.2 Describe the approach to testing
15th July 22nd July <b>SUMMER HOLIDAY</b>			<b>CONTINUE FROM 3.3 IN YEAR 13</b>