KS5 Long Term Plan

Subject: Computer Science **Exam Board OCR H446**

2020/2022

Curriculum Statement of Intent

Computer science KS5 builds on the skills and knowledge of KS4 computer science. The exam board is OCR. By the end of two years we intend all students to have strong content knowledge and be equipped with the necessary skills needed to succeed in their exams. The department aims to encourage all students to develop an interest in computing and to work in a confident and independent manner. We strive to equip students with the practical and theoretical skills necessary to flourish in the world of work. The department achieves this through providing a supportive learning environment and challenging all students to be the best that they can be.

Curriculum Statement of Implementation

We will achieve the above mentioned intend by:

- Having engaging and challenging lessons, creating a love for the subject.
- Ensuring pupil progress in every lesson, no student should be left behind.
- Covering curriculum content and making sure students have sound, in-depth knowledge.
- Mapping every lesson to curriculum content.
- Promoting independent activities / tasks to help students become independent learners.
- Students completing and reflecting on their PLCs every half term.
- Working online and teacher marking their progress using showbie.
- Regular formative tests to check student knowledge.
- Summative assessments every half term to recap / test learning.

	Autumn 1		Autumn 2		Spring 1	
Yr. 12	**Topics**	**Skills**	**Topics**	**Skills**	**Topics**	**Skills**
	2.2.1 Programming techniques **2.1.1 Thinking abstractly**	a) Programming constructs: sequence, iteration, branching. (b) Recursion, how it can be used and compares to an iterative approach. (c) Global and local variables. (d) Modularity, functions and procedures, parameter passing by value and by reference. (e) Use of an IDE to develop/debug a program. (f) Use of object oriented techniques (a) The nature of abstraction. (b) The need for abstraction. (c) The differences between an abstraction and reality. (d) Devise an abstract model for a variety of situations.	**1.2.3 Software Development** **1.2.4 Types of Programming Language**	(a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development. (b)The relative merits and drawbacks of different methodologies and when they might be used. (c)Writing and following algorithms (a) Need for and characteristics of a variety of programming paradigms. (b) Procedural languages. (c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set (d) Modes of addressing memory (immediate, direct, indirect and indexed). (e) Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism	**1.3.2 Databases** **1.3.4 Web Technologies**	(a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing. See appendix 5f. (b) Methods of capturing, selecting, managing and exchanging data. (c) Normalisation to 3NF. (d) SQL – Interpret and modify. See appendix 5d. (e) Referential integrity. (f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy. (a) HTML, CSS and JavaScript. See appendix 5d. (b) Search engine indexing. (c) PageRank algorithm. (d) Server and client side processing
Assessments	**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)** **Assessment**		**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)** **Assessment**		**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)**	

	Spring 2		**Summer 1**		**Summer 2**	
Yr. 12	**Topics**	**Skills**	**Topics**	**Skills**	**Topics**	**Skills**
	2c. NEA **1.4.2 Data Structures**	(a) Arrays (of up to 3 dimensions), records, lists, tuples. (b) The following structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table. (c) How to create, traverse, add data to and remove data from the data structures mentioned above. (NB this can be either using arrays and procedural programming or an object-oriented approach)	**2c. NEA** **1.4.1 Data Types**	(a) Primitive data types, integer, real/floating point, Character, string and Boolean. (b) Represent positive integers in binary. (c) Use of sign and magnitude and two's complement to represent negative numbers in binary. (d) Addition and subtraction of binary integers. (e) Represent positive integers in hexadecimal. (f) Convert positive integers between binary hexadecimal and denary. (g) Representation and normalisation of floating point numbers in binary. (h) Floating point arithmetic, positive and negative numbers, addition and subtraction. (i) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR. (j) How character sets (ASCII and UNICODE) are used to represent text.	**2c. NEA** **2.3.1 Algorithms**	(a) Analysis and design of algorithms for a given situation. (b) The suitability of different algorithms for a given task and data set, in terms of execution time and space. (c) Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity). (d) Comparison of the complexity of algorithms. (e) Algorithms for the main data structures, (stacks, queues, trees, linked lists, depth-first (post-order) and breadth-first traversal of trees). (f) Standard algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search)
	1.3.3 Networks	(a) Characteristics of networks and the importance of protocols and standards. (b) The internet structure: • The TCP/IP Stack. • DNS • Protocol layering. • LANs and WANs. • Packet and circuit switching. (c) Network security and threats, use of firewalls, proxies and encryption. (d) Network hardware. (e) Client-server and peer to peer network.	**1.3.1 Compression, Encryption and Hashing**	a) Lossy vs Lossless compression. (b) Run length encoding and dictionary coding for lossless compression. (c) Symmetric and asymmetric encryption.	**2.1 Elements of computational thinking**	2.1.2 Thinking ahead 2.1.3 Thinking procedurally 2.1.4 Thinking logically 2.1.5 Thinking concurrently

				(d) Different uses of hashing		
Assessments	**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Assessment**		**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)**		**Self-assessment of tasks (AfL)** **Teacher assessment of exam style questions (AoL)** **EOY Assessment**	

	Autumn 1		Autumn 2		Spring 1	
Yr. 13	**Topics**	**Skills**	**Topics**	**Skills**	**Topics**	**Skills**
	2c. NEA		**2c. NEA**		**2c. NEA**	
	2.2.2 Computational methods	(a) Features that make a problem solvable by computational methods. (b) Problem recognition. (c) Problem decomposition. (d) Use of divide and conquer. (e) Use of abstraction. (f) Learners should apply their knowledge of: • backtracking • data mining • heuristics • performance modelling • pipelining • visualisation to solve problems	**1.4.3 Boolean Algebra**	(a) Define problems using Boolean logic. See appendix 5d. (b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions. (c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation. (d) Using logic gate diagrams and truth tables. See appendix 5d. (e) The logic associated with D type flip flops, half and full adders	**1.5.1 Computing related legislation**	(a) The Data Protection Act 1998. (b) The Computer Misuse Act 1990. (c) The Copyright Design and Patents Act 1988. (d) The Regulation of Investigatory Powers Act 2000.
	1.2.1 Systems Software	(a) The need for, function and purpose of operating systems. (b) Memory Management (paging, segmentation and virtual memory). (c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the	**1.2.2 Applications Generation**	(a) The nature of applications, justifying suitable applications for a specific purpose. (b) Utilities. (c) Open source vs closed source. (d) Translators:Interpreters, compilers and assemblers. (e) Stages of compilation (lexical analysis, syntax	**1.1.1 Structure and function of the processor** **1.1.2 Types of processor** **1.1.3 Input, output and storage**	(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). Buses: data, address and control: how this relates to assembly language programs. (b) The Fetch-Decode-Execute Cycle; including its effects on registers. (c) The factors affecting the performance of the CPU:

		Fetch-Decode-Execute Cycle. (d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time. (e) Distributed, embedded, multi-tasking, multi-user and Real Time operating systems. (f) BIOS. (g) Device drivers. (h) Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate cod		analysis, code generation and optimisation). (f) Linkers and loaders and use of libraries		clock speed, number of cores, cache. (d) The use of pipelining in a processor to improve efficiency. (e) Von Neumann, Harvard and contemporary processor architecture
Assessments	**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)** **Assessment**		**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)** **Mocks**		**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)**	
	Spring 2		**Summer 1**		**Summer 2**	
Yr. 13	**Topics** **1.5.2 Moral and ethical Issues** **Revision**	**Skills** The individual moral, social, ethical and cultural opportunities and risks of digital technology: • Computers in the workforce. • Automated decision making. • Artificial intelligence. • Environmental effects. • Censorship and the Internet. • Monitor behaviour. • Analyse personal information. • Piracy and offensive communications. • Layout, colour paradigms and character sets	**Topics** **Revision**	**Skills**	**Exam Season**	

Assessments	**Self-assessment of tasks (AfL)** **Peer assessment of exam style questions (AfL)** **Teacher assessment of exam style questions (AoL)** **PPE**		