Year 13 Computer Science
December Mock Revision List

1.1 The characteristics of contemporary processors, input, output and storage devices	
Components of a computer and their uses	
1.1.1 Structure and function of the processor	(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: 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.
1.1.2 Types of processor	(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.
1.1.3 Input, output and storage	(a) How different input, output and storage devices can be applied to the solution of different problems. (b) The uses of magnetic, flash and optical storage devices. (c) RAM and ROM. (d) Virtual storage.

Year 13 Computer Science
December Mock Revision List

1.2 Software and software development	
Types of software and the different methodologies used to develop software	
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 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 code or running an operating system within another.
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 analysis, code generation and optimisation). (f) Linkers and loaders and use of libraries.
1.2.3 Software Development	(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.
1.2.4 Types of Programming Language	(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). See appendix 5d. (d) Modes of addressing memory (immediate, direct, indirect and indexed). (e) Object-oriented languages (see appendix 5d for pseudocode style) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism.

Year 13 Computer Science
December Mock Revision List

1.3 Exchanging data	
How data is exchanged between different systems	
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. (d) Different uses of hashing.
1.3.2 Databases	(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.
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.
1.3.4 Web Technologies	(a) HTML, CSS and JavaScript. See appendix 5d. (b) Search engine indexing. (c) PageRank algorithm. (d) Server and client side processing.

Year 13 Computer Science
December Mock Revision List

1.4 Data types, data structures and algorithms	
How data is represented and stored within different structures. Different algorithms that can be applied to these structures	
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.
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).*
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.

Year 13 Computer Science
December Mock Revision List

1.5 Legal, moral, cultural and ethical issues	
The individual moral, social, ethical and cultural opportunities and risks of digital technology. Legislation surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers	
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.
1.5.2 Moral and ethical Issues	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.