

Progression to Computer Science at Prior Pursglove College

I Data Structures: Check the textbook: OCR A Level Computer Science, PM Heathcote and RSU Heathcote, Chapter 7. This is the recommended textbook for the course and Chapter 7 gives an idea of one of the more difficult areas.

I Algorithm Complexity (covers a large section of the theory content):

https://www.youtube.com/watch?v=RJw_2UiZkkc&list=PLCiOXwirraUDEHKiMKzXUIGLxKPRicn7I&index=2

I Visual Basic practice (this is the main language we use):

<https://www.tutorialspoint.com/vb.net/index.htm>

[VB.Net Programming Tutorial – Tutorialspoint](#)

VB.Net is a simple, modern, object-oriented computer programming language developed by Microsoft to combine the power of .NET Framework and the common language runtime with the productivity benefits that are the hallmark of Visual Basic.

www.tutorialspoint.com

I Expanded specification for the first year Computing Science

Area	Spec. Ref	Content
Structure and Function of the Processor	1.1.1	ALU, Control Unit, Registers, MAR, MDR, CIR, PC, SR, ACC. FEC. CPU performance factors. Caching, clock speed, number of cores, pipelining. Pipelining (advantages, disadvantages). Buses (control, address, data). Von Neumann, Harvard architecture (comparison). Von Neumann bottleneck.
Types of Processor	1.1.2	RISC and CISC (comparison). GPUS, use of GPUs for non-graphic processing. Multicore and Parallel processors (types of parallel organisation)
Input, Output, Storage	1.1.3	Use of I/O devices in different situations. Magnetic, flash and optical storage.

		RAM and ROM. BIOS. Bootstrapping. Magnetic drives and solid state drives. Virtual storage. Advantages and disadvantages of virtual memory. Speed of access to different memory types.
Systems Software	1.2.1	Uses, functions of OS. Memory management (paging, segmentation, virtual memory.) Interrupts. Scheduling algorithms. Device drivers. Virtual machines. Types of OS (real-time, multi-user, multi-tasking, distributed). Utility software (examples).
Applications Development	1.2.2	Uses of applications for different purposes. Utilities. Open source vs closed source. Translators -- compilers, interpreters and assemblers. Java Byte Code, Java Virtual Machine (advantages). Comparison of compilers, interpreters and Java Byte code. Stages of compilation (lexical analysis, syntax checking, optimisation). Linkers, libraries.
Software Development	1.2.3	Waterfall cycle, Agile, Extreme Programming, Spiral, RAD (comparison of different methods). Stakeholders, Requirements specification, success criteria and measurement of criteria. Validation. Algorithms. Testing (testing strategies and test logs, test strategies compared (advantages and disadvantages). Data Dictionaries. Entity-Relationship Diagrams.
Types of Programming Language	1.2.4	Procedural languages. 1GL, 2GL and 3GL. Low-level and high-level languages. Advantages/ disadvantages of high-level and low-level languages. Assembler (Little Man Simulator). Object oriented languages. Encapsulation, polymorphism, inheritance. Advantages and disadvantages of OO.
Compression, Encryption, Hashing.	1.3.1	Advantages of compression. Lossy and Lossless Compression. When to use lossy and when to use lossless. Compression ratio. Asymmetric and Symmetric Encryption compared. Digital Signatures. Run Length and Dictionary Encoding (when to use each method). Hashing, hashing vs indexing.
Databases	1.3.2	Relational Databases, flat files (disadvantages), primary key, secondary key, foreign key. Normalisation (1NF – atomicity of data). Referential integrity. Entity Relationship Diagrams, splitting M-to-M relationships.
Networks	1.3.3	Characteristics of networks (advantages, disadvantages), protocols (why they're needed). TCP/IP, DNS, Protocol layering (advantages). LANs

		and WANs, packet switching, circuit switching. Network security. Error-checking (different methods), proxy servers. Network hardware (routers, modems, switches, hubs – comparison of different devices). Client-server, peer-to-peer (comparison).
Web Technologies	1.3.4	HTML (know various tags), CSS (know various tags), JavaScript (variables, loops, IFs, functions). Uses of CSS and JavaScript. Advantages/ disadvantages of CSS. Server and client side processing (comparison). Page Rank Algorithm. Search Page Indexing.
Data Types	1.4.1	Primitive (simple) data types. Positive numbers in binary. Sign and Magnitude and Twos complement. Positive decimal conversions to binary and hex. Fixed point fractions. Floating point fractions, loss of precision with floating point. Normalisation of fixed point fractions (mantissa and exponent). Disadvantages and disadvantages of Fixed Point and Floating Point. ASCII and Unicode
Data Structures	1.4.2	Primitive (simple) data types. Composite data types. Dynamic and Static data structures (comparison). Arrays (1D to 3D), Records. Lists (Python), Tuples, Stacks, Queues, Trees. Add/remove elements from stacks/queues. Add elements to binary trees, (stored in sequence when searched in-order)
Boolean Algebra	1.4.3	Define problems, propositions, subject, predicate. Truth values, truth tables. Tautologies, contradictions. Logical connectors, logical equivalents. Logic gates. Deriving Boolean expressions from logic gates. Half-adder, full adder, uses of adders. Karnaugh maps for 2,3 and 4 variables. Uses of K maps. K maps for simplifying an expression. De Morgan's Law. Association, distribution, commutation, double negation for Boolean expressions.
Computer Legislation	1.5.1	Data Protection Act (terms). Computer Misuse Act (three offences). Copyright, Design and Patents Act. Regulation of Investigatory Power Act (arguments for and against)
Moral and Ethical Issues	1.5.2	Computers in the workplace – loss of jobs, making work easier. Computers making automated decisions – Knowledge-Based Systems. Artificial intelligence – good and bad aspects of robots. Effects of computers on the environment. Internet dangers (trolling, cyberbullying, internet fraud, terrorism, paedophilia). Checking on behaviour using cyber techniques. Storing personal information. Software

		piracy. Design of websites (appropriate use of colour, interfaces for disabled access, character sets for foreign languages.)
Thinking Abstractly	2.1.1	Need for abstraction (advantages). Abstract models versus real (concrete) situations. Use of abstraction in different scenarios. Structure diagrams used in abstraction. Control and data abstraction.
Thinking Ahead	2.1.2	Inputs and outputs for a given situation. Devising solutions to a give problem. Caching. Reusability.
Thinking Procedurally	2.1.3	Breaking down problems into manageable units (structure diagrams). Use of procedures (modular approach - - advantages).
Thinking Logically	2.1.4	Use of flow charts. Use of decision points (IF conditions).
Thinking Concurrently	2.1.5	Concurrent vs parallel processing, identifying parts of a program which can be done concurrently, benefits and disadvantages of concurrent processing. Deadlock, starvation, race conditions.
Programming Techniques	2.2.1	Program constructs -- sequence, iteration, branching (selection, IF and Case statements). Global, local variables. Procedures and functions. Parameters (passing by value and by reference). IDEs. Object-oriented techniques (advantages, disadvantages of object orientation, encapsulation, inheritance, polymorphism). Readability of programs.
Computational Methods	2.2.2	Solving problems using computational methods, things which the computer is not good at. Problem recognition, intractable problems, problem decomposition (structure charts to break down a problem). Divide and conquer algorithms. Abstraction, Backtracking, data mining, heuristics, performance modelling, pipelining, visualisation. Object oriented techniques.
Algorithms	2.3.1	Design algorithms for given situation. Algorithms for adding, deleting from Stacks, Queues, Trees. In-order tree traversal. Insert item into binary tree (in-order sequence). Linear and Binary search (compare). Bubble sort and insertion sort (efficiency). Trace tables for insertion sort and bubble sort. 2D array algorithms, array of records processing algorithms. File-processing algorithms.

If you have any further questions please contact Computer Science Teacher Peter Li-Ping via p.liping@pursglove.ac.uk