Curriculum Mapping: Computer Science Year 12-13

St Martin's School
Be the best you can be

Year	Autumn 1.1	Autumn 1.2	Autumn 2.1	Autumn 2.2	Spring 1	Spring 2	Summer 1	Summer 2
Year 12	***Getting started***	***Problem solving***	***Data representation***	***Hardware and software***	***Computer organisation and architecture***	***Communications, networks and consequences***	***Databases and software development***	***OOP***
	Concepts/Tier 3 vocabulary algorithm, structured programming, data type, variables, constants, assignment, arithmetic operations, Boolean operators, sequence, selection, definite and indefinite iteration, top down design, modular programming, subroutine, procedure, function, parameter, argument, exception handling, global and local variables, field, record, binary file, text file, data structure	***Concepts/Tier 3 vocabulary*** computational thinking, algorithm, simulation, enumeration, divide and conquer, top down design, hierarchy chart, test plan, erroneous data, trace table, abstraction, information hiding, procedural abstraction, functional abstraction, data abstraction, decomposition, composition, automation, finite state machine, transition table	***Concepts/Tier 3 vocabulary*** *Natural, rational, irrational, hexadecimal, binary, signed and unsigned, kibi, mebi, gibi, ASCII, Unicode, parity, checksum, check digit, overflow, raster, bitmap, resolution, bit or colour depth, sample, MIDI, frequency, Hertz, lossy, lossless, compression, encryption, ciphertext, plaintext, cryptanalysis.*	***Concepts/Tier 3 vocabulary*** *Hardware, general-purpose/special-purpose software, operating system, utility programs, defragmenter, virus checker, library program, translator, virtual machine, processor scheduling, interrupt, embedded system, machine code, assembly language, assembler, compiler, interpreter, bytecode, logic gate, truth table, Boolean algebra*	***Concepts/Tier 3 vocabulary*** *Processor, main memory, address bus, data bus, control bus, I/O controller, von Neumann, Harvard, addressable memory, stored program concept, fetch, decode, execute, arithmetic logic unit, control unit, clock, register, buffer, instruction set, opcode, operand, immediate addressing, direct addressing, machine-code, branch, logical bitwise operator, logical shift, assembly language, cores, RFID, polarisation, pulse, flash, block, page, transistors, latency.*	***Concepts Tier 3 vocabulary*** *Serial transmission, parallel transmission, USB (Universal Serial Bus), synchronous transmission, asynchronous transmission, start bit, stop bit, baud rate, bit rate, bandwidth, latency, protocol, star topology, bus topology, network, client-server networking, peer-to-peer networking, Wi-Fi, wireless access point, WPA, WPA2, SSID, MAC, CSMA/CA, RTS/CTS, privacy.*	***Concepts Tier 3 vocabulary*** *entity, attribute, identifier, primary key, composite primary key, foreign key, relationship, entity relationship (E-R) diagram, normalisation, relation, relational database, First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), partial dependency, non-key dependence, data integrity, SQL, client-server database, record locking, serialisation, timestamp ordering, commitment ordering, agile modelling, prototyping*	***Concepts Tier 3 vocabulary*** *Object, class, attribute, method, encapsulation, information hiding, constructor, instantiation Inheritance, subclass, superclass, polymorphism, overriding, modifier, public, private, protected, class diagram, aggregation, composition, association, abstract method, virtual method, static method, interface.*

	Justification: This unit covers the principles of structured programming in Python, arrays, subroutines, parameter passing and text and binary files.	***Justification:*** This unit describes what is meant by "computational thinking" and is designed to develop this skill with the aid of many practical examples related to problem solving, abstraction and algorithm design.	***Justification:*** *This unit covers the representation of data, six topics in this unit cover data representation of numbers, text, images and sound, with the final topic explaining and giving examples of the uses of data compression and encryption.*	***Justification:*** *The unit begins with a lesson on hardware and software and the classification of software. The role of an operating system is then covered, followed by lessons on the classification of programming languages as low-level and high-level, and programming language translators. The last two topics deal with logic gates and Boolean algebra.*	***Justification:*** *The unit begins by describing the internal hardware components of a computer, different architectures and the stored program concept. The fetch-execute cycle is explained in a detailed and practical way including the role of the major components and dedicated registers used by the processor. Instruction sets and addressing are covered along with basic machine code and assembly language operations. The function and characteristics of various external hardware devices and storage methods are explained in the final two topics.*	***Justification*** *Students learn about: Communications methods, including baud rate, bit rate, bandwidth, latency and protocols; Network topologies, including physical star and logical bus; Client-server and peer-to-peer networking; Wireless networking, including CSMA and SSID; The unit concludes with two lessons on communications and privacy and the social, legal and cultural issues presented by the use of computers and communication methods in today's world*	***Justification*** *The first two lessons cover the production of a data model, entity definitions and entity relationship diagram, and normalisation to Third Normal Form. The next two lessons cover the use of SQL to retrieve, update, insert and delete data from multiple tables in a database, and the creation of new tables. Client server database and problems of concurrent databases are also covered. The final lesson covers aspects on software development.*	***Justification*** *These lessons cover the basics of object-oriented programming and object-oriented design principles, with practical examples in Python*
	Assessment: All units have an end-of-unit test, which draws together skills and knowledge from the previous lessons.							
	Wider reading/Cultural capital Throughout KS5 there are opportunities for students to access varied curriculum that is ambitious and goes beyond the specification. From the onset we focus on ensuring that our students are aware of the ethical, legal, environmental, and cultural issues related to computer science. Students are							

	regularly reminded how to be respectful digital citizens. A solid understanding of how technology works from the inside out and how it affects the real world is vital for students to be able to succeed. Students in year 12 and 13 attend Computer Science in Action lectures.					
	Autumn 1	**Autumn 1-2**	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**
Year 13	***Data Structures***	***Algorithms***	***Regular Languages***	***Non-exam assessment***	***The Internet***	***Functional programming and Big Data***
	Concepts/Tier 3 vocabulary *Elementary data type, abstract data type, encapsulation, information hiding, static data structure, dynamic, heap, overflow, underflow Queue, circular queue, priority queue, First In, First Out (FIFO), enqueue, dequeue, stack, Last In, First Out (LIFO), call stack, stack frame, Hashing, hash table, collision, mid-square method, folding method, dictionary Graph, edge, arc, vertex, adjacency matrix, adjacency list, Tree, root, child, parent, subtree, leaf node, binary, pre-order, in-order and post-order traversal vector, dot product,*	***Concepts/Tier 3 vocabulary*** Recursion, call stack, tree traversal, pre-order, in-order, post-order traversal, Big-O notation, linear, polynomial, exponential, logarithmic functions, permutation, time complexity, binary tree search, bubble sort, merge sort, depth-first traversal, breadth-first traversal, optimisation problem, Dijkstra's, travelling salesman problem (TSP), tractable and intractable problems, heuristic solution, computable and non-computable problems, Halting problem	***Concepts/Tier 3 vocabulary*** *Finite state machine, Mealy machine, transition, state transition table Set, member, element, set comprehension, compact representation, membership, union, intersection, difference, subset, proper subset, Cartesian product, infinite, finite, countably infinite, cardinality Regular Expressions, regular language, decompose, Turing machine, state transition diagram, tape, read-write head, halting, Universal Turing machine, computable, Backus-Naur Form, pipe, syntax diagram, parsing, parse tree, Infix, prefix, postfix, Reverse Polish Notation*	***Concepts/Tier 3 vocabulary*** *Analysis Documented design Technical solution Testing Evaluation*	***Concepts/Tier 3 vocabulary*** Internet, World Wide Web, URL, Internet registry, registrar, DNS, FDQN, Internet Protocol, packet, packet switching, router, gateway, hop, header, NIC, firewall, filter, proxy server, port, stateful inspection, encryption, symmetric, asymmetric, public, private key, digital signature, hash, digital certificate, worm, Trojan, malware, virus, TCP, stack, protocol, MAC address, FTP, SSH, POP, SMTP, IMAP, server, browser, subnet mask, DHCP, routable, non-routable, NAT, port forwarding, client server model, API, CRUD, JSON, XML, REST, thick client, thin client.	***Concepts/Tier 3 vocabulary*** first-class object, functional composition, partial function application, higher-order functions, map, filter, fold, graph

	Justification: *The unit gives practical and worked examples of each of the different abstract data structures including queues, stacks, lists, graphs, trees, hash tables and dictionaries. The function and practical application of each data type is discussed, with pseudocode and coded program solutions for relevant algorithms Python. Vectors and dot products and their application are covered in a final topic.*	***Justification:*** Searching and sorting algorithms are covered in an interactive and practical way, with reference to Big-O notation in terms of time and space complexity. It also covers the role of stack frames in subroutine calls, and recursive techniques, putting these into practice with tree traversals and a depth-first graph traversal. Optimisation algorithms, such as Dijkstra's shortest path algorithm are covered along with a complete topic on the limits of computation.	***Justification:*** After covering Mealy machines in the first lesson, sets and regular expressions are covered. The structure and use of Turing machines that perform simple computations are discussed and Backus-Naur form and syntax diagrams are explained. The last topic covered is Reverse Polish notation with students being given plenty of opportunity to practise skills and techniques throughout each lesson.	***Justification:*** The project allows students to develop their practical skills in the context of solving a realistic problem or carrying out an investigation. The project is intended to be as much a learning experience as a method of assessment; students have the opportunity to work independently on a problem of interest over an extended period, during which they can extend their programming skills and deepen their understanding of computer science.	***Justification:*** Internet functions including packet switching, DNS and the role of the router are covered in the first two topics of this unit. Symmetric and asymmetric encryption, and the use of digital signatures are covered in the following topic. Standard Application Layer protocols such as SSH are covered with reference to the TCP/IP protocol stack. Subnetting, DCHP and Network Address Translation are covered in the penultimate topic, rounded off with a final topic on web CRUD and RESTful applications in relation to the client server model.	**Justification:** Function programming is taught using Haskell, accompanied by theory to enable students to answer exam questions on this topic. The final lesson describes examples of Big Data, its application and benefits in areas such as healthcare and medicine, business, communication and many other fields.		
		Assessment: All units have an end-of-unit test, which draws together skills and knowledge from the previous lessons.						
		Wider reading/Cultural capital Throughout KS5 there are opportunities for students to access varied curriculum that is ambitious and goes beyond the specification. From the onset we focus on ensuring that our students are aware of the ethical, legal, environmental, and cultural issues related to computer science. Students are regularly reminded how to be respectful digital citizens. A solid understanding of how technology works from the inside out and how it affects the real world is vital for students to be able to succeed. Students in year 12 and 13 attend Computer Science in Action lectures.						