

Time period			Spring 1	Spring 2	Summer 1	Summer 2
<b>Content</b> <i>Declarative Knowledge</i> – <i>‘Know What’</i>	<u>1.3.4 – Web technologies</u> a) HTML, CSS and JavaScript continued d) Server and client-side processing	<u>1.3.3 - Networks</u> a) The TCP/IP Stack b) Protocol Layering c) LANS and WANS d) Packet and circuit switching e) Protocols f) Client-Server and Peer to Peer	<u>1.4.1 – Data types</u> (f) Floating point arithmetic (g) Bitwise manipulation and masks (h) character sets (ASCII and UNICODE) <u>1.4.2 – Data structures</u> (b) 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.	<u>1.4.2 – Data structures - continued</u> b) linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table <u>1.2.4 -Types of Programming Language</u> b) Assembly language <u>1.3.4 – Web technologies</u> a) HTML, CSS and JavaScript	<u>1.1.3 - Input, output and storage devices</u> (a) input output and storage devices (b) magnetic, flash and optical storage devices (c) RAM and ROM (d) Virtual storage <u>1.1.1 Structure and function of the processor</u> (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) (b) The Fetch-Decode- Execute Cycle	<u>1.2.1 - Systems Software</u> (a) Operating systems (e) Distributed, Embedded, Multi-tasking, Multi-user and Real-Time operating systems (f) BIOS (g) Device drivers <u>1.2.2 - Applications Generation</u> (a) The nature of applications (b) Utilities (c) Open source vs Closed source <u>1.4.1 – Data types</u> (a) Positive integers in binary (d) hexadecimal (b) Negative numbers in binary (c) Addition and subtraction of binary integers (e) Floating-point numbers in binary (f) Floating point arithmetic, positive and negative numbers, addition and subtraction
<b>Skills</b> <i>Procedural Knowledge</i> – <i>‘Know How’</i>	<u>1.3.4 – Web technologies</u> a) HTML, CSS and JavaScript d) the benefits of client-side processing (JavaScript) server-side	<u>1.3.3 - Networks</u> a) The TCP/IP Stack, (b) Protocol Layering (e) Protocols - how the use of sockets and TCP allows guaranteed connections.	<u>1.4.1 – Data types</u> (f) To use addition and subtraction arithmetic on Floating point positive and negative numbers, (g) Use Bitwise manipulation and masks:	<u>1.4.2 – Data structures - continued</u> b) The structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table	<u>1.1.3 - Input, output and storage devices</u> a) How different input output and storage devices can be applied to the solution of different problems	<u>1.2.1 - Systems Software</u> (a) The function and purpose of operating systems (e) Distributed, Embedded, Multi-tasking, Multi-user and Real-Time operating systems.

	<p>processing across a network. Practical exercises using PHP, MySQL and web technologies. Students need to understand the purpose of HTML, CSS and JavaScript. They should have experience of writing webpages using HTML, CSS and JavaScript. (d) understand the difference between server and client-side processing, and should be aware of examples (for example JavaScript code vs PHP code) of processing on both sides. They should be aware of the benefits and drawbacks of both types of processing.</p>	<p>(c) LANS and WANs allow data to be interrogated and transmitted. (d) Packet and circuit switching (f) Client-Server and Peer to Peer - How networks are configured and connected to the internet -How IP addresses are mapped in and out of networks, use of proxies, -Network hardware, encryption, compression, hashing, sockets, protocols -the threats posed by malicious software and how computer systems can be attacked</p>	<p>shifts, combining with AND, OR, and XOR (h) How character sets (ASCII and UNICODE) are used to represent text <u>1.4.2 – Data structures</u> (b) Students will know the data structures used to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table. They will know when each of these should be used and how to write program code implementing these data structures. (c) Students will understand the behaviour of each data type and know how to</p>	<p>c) How to create, traverse, add data to and remove data from the data structures mentioned above <u>1.2.4 -Types of Programming Language</u> (b) Assembly language Using and writing simple programs with the Little Man Computer instruction set to understand simple assembly code language. Understand how high- level language routines are translated into low level processor instructions, and what those low-level instructions look like. <u>1.3.4 – Web technologies</u> a) Students need to understand the purpose of HTML, CSS and JavaScript. They will know how to write webpages using HTML, CSS and JavaScript. They will be able to read, write, amend and interpret code using HTML, CSS and JavaScript.</p>	<p>b) The uses of magnetic, flash and optical storage devices c) The differences between RAM and ROM d) How Virtual storage works <u>1.1.1 Structure and function of the processor</u> a) How this relates to assembly language programs b) How the FDE cycle decodes each instruction/data and its effects on the registers.</p>	<p>(f) The BIOS is used in the bootup process (g) The purpose of Device drivers and how they communicate with the Operating system. <u>1.2.2 - Applications Generation</u> (a) Justify suitable applications and (b) utility software for a specific purpose (c) Understand the difference and uses of Open source vs Closed source <u>1.4.1 – Data types</u> (a) Represent positive integers in binary (b) Use of Sign and Magnitude and Two’s Complement to represent negative numbers in binary (c) To use Addition and subtraction of binary integers (d) Represent positive integers in hexadecimal (e) Representation and normalisation of floating- point numbers in binary (f) Addition and subtraction of Floating-point arithmetic, positive and negative numbers,</p>
<b>Key Questions</b>	<p>What is the purpose of HTML, CSS and JavaScript? What is the difference between server and client-side processing? What are the benefits of server-side</p>	<p>What is the definition and purpose of a network? What is the importance of using protocols? What are network standards? What is the purpose and benefits of layering</p>	<p>Why do we need to normalise floating-point numbers? What is the effect of right and left binary shifts on binary numbers? What is the</p>	<p>How does assembly language make use of registers? How are data and addresses transferred between registers? What is addressing? Which should be integrated with</p>	<p>What are the range of input and output storage devices used? What are the different types of storage devices? What are the characteristics of each type? Magnetic? Optical? And Flash? What is</p>	<p>Why do devices need an operating system? What different tasks does the OS perform? How do operating systems manage memory? What is paging and segmentation? How do</p>

	processing? What are the benefits of client-side processing?	protocols, particularly within the TCP/IP stack. What are the different layers within the TCP/IP stack? What is the purpose of each layer? How data is transmitted on the Internet? How are IP addresses and packets used in the transfer of data? What is a LAN and WAN? How is the Domain Name System used to find the IP of a URL? What is the purpose, function, benefits and drawbacks of both packet and circuit switching? What are network security issues and threats? How do we minimise or prevent these threats? What is the hardware required to connect to and/or build a network? What is the difference between a client-server and peer-to-peer network.	purpose of using masks with bitwise operations? What is a data structure? What is the difference between a dynamic and static data structure? What is LIFO structure? What is a FIFO structure? What is the relationship between data and memory? How is data stored in each of these structures, how is memory allocated? What is a circular queue? What is a priority queue? What is a linked list? What is Pre? Post and in order traversal? Where are each of the data structures suitable for a given data set or application?	assembly language? What is immediate, direct, indirect and indexed addressing in the writing, reading and tracing of programs written in assembly language? What is the purpose of HTML, CSS and JavaScript? What is the difference between server and client-side processing?	the purpose of ROM and RAM within a computer system? Why is there a need for virtual storage? How does virtual storage work? What are the benefits and drawbacks of virtual storage? What is the purpose and function of the core components of the processor. What is the purpose of the registers? The Data, addresses and control buses? What is the purpose of the FDE cycle? What are the stages of the FDE cycle? How are the registers used within the FDE cycle? What factors will affect the performance of the CPU? What is the Von Neumann architecture? What is the Harvard architecture? How does these architectures differ in storing instructions and data in memory? What are the benefits of each architecture?	segments allow access to memory What is virtual memory? What are the different classifications of operating systems? What are the key features of each? What is the role of the BIOS in a computer system? What is meant by device drivers? Why are they needed for communication between hardware and the operating system? What is meant by a virtual machine? How are they used to execute intermediate code? What is the purpose of applications? Utility? General purpose? What are the differences between open and closed source software? How are real numbers represented in a binary floating-point representation?
<b>Assessment</b>	End of unit tests, Exam style HBL questions Past exam questions to consolidate learning Practical activities using HTML, CSS And Javascript. Trial exams					
<b>Literacy/Numeracy/SMSC/Character</b>	Programming language literacy Computational literacy Exemplar modelling of answers Understanding of key word definitions.	Programming language literacy Computational literacy Exemplar modelling of answers Understanding of key word definitions.	Computational literacy Exemplar modelling of answers Understanding of key word definitions. Scaffolded answers to LAQ, guided through AO1,	Programming language literacy Computational literacy Exemplar modelling of answers Understanding of key word definitions.	Computational literacy Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills	Computational literacy Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills Mathematical computation Data handling Linear Algebra
	Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills	Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills	AO2 and AO3 evaluative skills Mathematical computation Data handling Linear Algebra Discrete mathematics Graph theory	Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills Mathematical computation Data handling Linear Algebra Discrete mathematics Graph theory		Discrete mathematics