Computing Curriculum Map

Intent

The curriculum has been designed to empower children with virtues that enable them to excel academically and spiritually inspiring them to serve humanity selflessly (Nishkam), with an abundance of love, compassion and forgiveness. The curriculum aims to support students to learn about peace, forgiveness, love and faith in the Divine through their academic subjects, faith practice and personal development.

Our curriculum is constructed around our vision to ensure we remain:

Faith-inspired: learning from the wisdom of religion
Our students explore the divine context of humanity and wonder of all creation. They not only learn about, but also learn from, the wisdom of religions and in so doing explore the infinite human potential to do good unconditionally. We support students to develop aspects of their own religious, spiritual or human identities. They learn about serenity through prayer and humility in service and in so doing, they deepen their own respective faith, and respect the common purpose of all religious traditions, as well as respecting the beliefs of those with no faith tradition. They explore the unique divinity of the individual, and our common humanity.

Virtues-led: nurturing compassionate, responsible human beings
We believe that the fostering of human virtues forms the foundation of all goodness. Our curricula are carefully enriched to allow experiences where our students, teachers and parents alike learn to grow through a conscious focus on virtues. Our virtues-led education approach helps to provide guidance to enable students to understand their choices in order to help lead better lives. Our students become self-reflective and flourish; they are able to build strong, meaningful relationships and understand their responsibilities to the global family and all creation, founded in faith. Students learn to experience faith through lived out through righteous living in thought, action and deed.

Aspiring for Excellence: in all that we do.
Our students and staff alike aim to become the best human beings they can possibly be, in all aspects of spiritual, social, intellectual and physical life. We foster a school culture which inspires optimism and confidence, hope and determination for all to achieve their best possible. This is accomplished through a rich and challenging curriculum, along with excellent teaching to nurture awe and wonder. Students gain a breadth and depth of knowledge and a love of learning to achieve their full potential.
The curriculum at Nishkam School West London has been carefully crafted to be broad, balanced and stimulating, giving every Nishkam student the opportunity to be knowledgeable, multi-skilled, highly literate, highly numerate, creative, expressive, compassionate and

confident people. Knowledge-rich, skills based and Faith-inspired, the Curriculum at Nishkam School West London is delivered through three **Golden Threads** that are unique to our ethos and virtues:

1	Love and forgiveness vs. Enmity and Hate
2	Peace and Collaboration vs. Conflict and War
3	Trust in God

Every composite of our curriculum is constructed of components that have each of these threads at their core. These elements can be clearly identified in our subject-based curriculum maps and Schemes of Learning documents.

A high-quality computing education equips students to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which students are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, students are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that students become digitally literate, preparing them for the future workplace and enabling them to be active participants in a digital world.

The curriculum is necessarily aspirational, focused on excellence and on securing in all learners a love of learning through the acquisition of knowledge, the study and practice of faith, and an understanding of the world around them. One aspect of the curriculum is the school ethos of the golden threads. Students will learn via collaboration, peace, forgiveness, and love through each unit of work.

Implementation:

The Primary Phase is guided by the iCompute Scheme of Work for Computing. This model effectively supports teachers to deliver a rich, broad and balanced computing curriculum fully mapped to the National Curriculum. It offers students a computing education designed for mastery using research-led computing pedagogies and covers all three strands of the computing curriculum:

- Computer Science
- Information Technology
- Digital Literacy (including eSafety)

Students are taught through whole-class interactive teaching with students working together on the same lesson content at the same time. Lessons are sequenced so that concepts are developed in logical steps with particular attention given to fundamental concepts. This ensures that all children can master concepts before moving to the next stage. Curriculum equity is offered with all students being given the time and opportunity to fully understand, explore and apply skills and ideas in different ways, in different situations and in different subjects. This enables students to fully grasp a concept and understand the relevance of their learning.

In the Secondary Phase, our intent is to broaden and deepen our students understanding of IT and Computer Science, allowing students to be more independent with their approaches to computational thinking and their solutions to complex problems relating to real-world scenarios. Students can link knowledge from across different units to support their overall understanding of the course.

Curriculum Overview

Year	Half Term 1	Half Term 2	Half Term 3	Half Term 4	Half Term 5	Half Term 6
EYFS	**iMake Music** **iMake Media** **iMake Video** **iCan Play** Exploring and creating with materials	**iCan Move** **iCan Direct** **iFind Patterns** **iAm Logical** Moving and handling: Directions	**iOrganise Data** **iSearch Online** **iCan Sequence** **iCan Program** Number and Technology	**iStay Safe** **iMake Art** **iCan Control** **iCan Sort/Turn** **iCan Animate** The world and keeping safe online	**iTell Stories** **iSend Email** **iMake Pictograms** **iCan Surf** Being imaginative and exploring technology	**iCan Report** **iCatch Aliens** **iMake Algroithms** **iGuess Beasts** **iMake Pixel Art** Creating and representing
1	**iAlgorithm** Read, produce, and understand instructions	**iData & iModel** Navigate, explore, and create using IT	**iModel & iSafe** Using websites and being digitally safe	**Finish iSafe & iWrite** Use digital tools to communicate meaning	**iProgram 1** Read, produce & understand instructions	**iProgram- Unit 2** Producing instructions using IT software
2	**iPub** Using IT tools for a purpose	**iProgram** Produce sequenced instructions	**iSearch** Problem solving by creating instructions	**iSafe** Communicate and identify risks of IT	**iDo Mail & iBlog** Make choices about applications & tools to use for a particular purpose	**iBlog** Identify suitable information to present
3	**iData** Design and develop basic computer programs	**iConnect** Communication and collaboration online	**iNetwork & iSafe** Understand computer networks	**iSafe** Use technology safely, respectfully and responsibly	**iProgram** Design and develop basic computer programs	**iSimulate** Using software, IT & programs to create and apply
4	**iAnimate** Exploring the uses of information online	**iProgram** Test, debug and refine algorithms and programs	**iProgram & iSafe** Demonstrate, check and understand risks of ICT	**iSafe** Demonstrate, check and understand risks of ICT	**iMail** Send and reply to emails	**iData** Explore and use more advanced features of applications
5	**iWeb** Understand, develop, and refine digital content	**iCrypto** Digital communication with an audience	**iSafe** Write and amend computer programs	**iSafe** Communicate effectively and safely online	**iProgram - Unit 1** Program algorithms that achieve a range of specified outcomes	**iProgram - Unit 2** Create programs. Test, debug and refine computer programs.
6	**iProgram** Write & amend more complex	**iNetwork**	**iSafe** Find information online and check it	**iSafe & iApp- Unit 1** Discuss opportunities for	**iApp - Unit 2** Communicate and collaborate using	**iData** Develop and refine digital content

	computer programs to create a variety of outcomes	Create digital content e.g. webpage	for accuracy and reliability	communication and collaboration online	technology and online services	
7	**Digital Literacy And E-Safety** Students revise the practical skills of using desktop Windows PCs. Students are encouraged to become independent, resilient problem solvers. Students will also demonstrate how to use internet connected devices safely – recognising online threats and taking appropriate safety measures. <u>Topics Covered:</u> • Logins and Passwords • File Explorer and OneDrive • MS Office and Teams. • Keyboard shortcuts • PowerPoint and Word	**Spreadsheet Modelling** Students deepen their understanding of the MS Office applications by creating Excel spreadsheets, applying formatting, using formulas, functions, and creating graphs. Students also use the MS Forms application to create a survey and MS Office to send and receive emails. <u>Topics Covered</u>: • Online and standalone applications. • Using cell references. • Using arithmetical formulas. • Formatting with fill colour.	**Programming in Scratch** Students use block-based game development environment Scratch to develop their skills using programming constructs. Students develop a game project over the half term – developing a Pac Man game. This will require use of programming constructs - input, output, variables, selection, sequence, iteration. <u>Topics Covered:</u> • Tutorials, File Management and sharing work.	**Hardware** Students explore the physical hardware devices that make up a computer. Students will learn about different types of computers and the purpose of memory, CPU, BIOS and the motherboard. <u>Topics Covered:</u> • Embedded, personal and supercompu-ters. • Input and output devices. • Primary and secondary storage.	**Data Representation: Binary** Students begin to understand how computers can represent anything as they are electrical systems which can only represent a binary state 0 or 1. Students learn to convert denary to binary and vice versa. Students will learn units of data bit, nibble, byte, megabyte, kilobyte, gigabyte. Students will learn how computers use binary to represent ASCII characters and images. <u>Topics Covered:</u> • How do computers count? The binary system	**Game Development in Scratch** Students continue to develop their programming skills and knowledge in the Scratch environment. Students will complete their knowledge of programming constructs – subroutines, condition-controlled loops, count controlled loops, and lists. <u>Topics Covered:</u> • Develop a dance battle game using subroutines. • Develop a cat flying game using condition-controlled loops.

	• Cyberbullying and Online Grooming. • Browser and Search • Plagiarism	• Formatting with borders. • Creating a Forms survey. • Emailing links. • Using functions for average, max, min, sum. • Displaying pie, bar and line graphs. • Conditional formatting.	• Creating sprite graphics and animation. • Developing an algorithm for movement. • Creating a backdrop graphic. • Developing an algorithm for collision detection. • Developing an algorithm for score keeping using a variable. • Creating enemy sprites. • Developing an enemy sprite movement algorithm. • Testing and debugging	• The CPU and BIOS. • Motherboar-ds, sound cards and graphics cards.	• How do humans count? The denary system. • Converting denary to binary • Units of data. • Converting binary to denary. • ASCII characters.	• Comparing types of loops. • Develop a treasure hunt game using lists. • Develop a quiz game using all skills. • Testing and debugging
8	**Online Safety and Cyber Security** Students review existing online safety knowledge and expand their knowledge of online threats to include malware and malicious hackers – and	**Computational Thinking** Students define the four cornerstones of computational thinking – decomposition, abstraction, pattern recognition and algorithm design	**Programming in Python** Students build on their previous programming experience using the block-based Scratch and progress to the textual programming language Python.	**Computer Systems and Artificial Intelligence** Students build on their knowledge of physical hardware to explore the properties of computer systems, their limitations	**Python Turtle Graphics** Students build on their experience of Python programming and use the Turtle Graphics library to demonstrate programming constructs. This will also build on	**Data Representation: Images** Students build on their existing knowledge of data representation (binary) to understand ways that computers

	how to defend against them. <u>Topics Covered:</u> • Personal data security and privacy • Social engineering attacks blagging, shouldering and phishing. • Automated attacks, DDoS attacks. • Security Protection – antimalware, firewall. • Online cyber games.	as well as the essential skill of debugging to tackle computational thinking puzzles <u>Topics Covered:</u> • What are algorithms? • Decomposition – breaking larger problems into smaller ones. • Abstraction – ignoring unnecessary details • Pattern Recognition – making use of repeated patterns or trends to solve similar problems • Algorithm Design – representing algorithms with flowcharts.	Students will learn to output, input, store data in variables, and use selection. <u>Topics Covered:</u> • Using the Replit online IDE • Output using the print function. • Input and storing in variables. • Using comments to improve code. • Using if statements for conditional branches. • Using elif statements for multiple branches.	and what the future may hold. <u>Topics Covered:</u> • What is a general-purpose computer? • Processor, memory and storage. • Operating systems as the conductor. • Logic gates and circuits. • Artificial intelligence and machine learning. • Open-source vs proprietary software.	their experience of computational thinking. <u>Topics Covered:</u> • Defining an algorithm to draw a square. • Using sequence to draw rectangles. • Using pattern recognition to solve the problem of drawing polygons with different number of sides. • Using repetition with for loops to draw patterns with shapes. • Using the define function to reuse code and call subroutines.	can represent images. Students will learn how key terms resolution and colour depth affect file size. Students will learn to edit images using vector graphics and raster graphics. <u>Topics covered:</u> • Using Inkscape vector graphics editing software. • Shape properties, fill and outline. • Using Gimp 2.0 bitmap editing software. • Image files, resolution and colour depth. • How compression affects filesize.

9	**Cyber Security and Malware** Students build on existing cyber security knowledge to deepen their knowledge on cyber threats and security defences, learning how the law defines criminal behaviour online. <u>Topics Covered:</u> • Data Protection Act – the law and personal information. • Giving advice on suspicious emails to prevent phishing attacks. • Computer Misuse Act – the law and computer crimes and sentencing. • Online cyber security	**Algorithms, Flowcharts and Pseudocode** Students build on their knowledge of computational thinking to show how algorithms can be represented using both flowcharts and pseudocode. Students will use flowcharts to represent sequence, selection, and iteration. Flowcharts can then be used to control simulated systems using Flowol. <u>Topics Covered:</u> • Algorithm definition and giving examples. • Flowchart standard shapes. • Pseudocode model examples. • Representing sequence in flowcharts.	**Python Programming Constructs** Students build on the knowledge of Python programming and their experience using Python turtle graphics as well as computational thinking and algorithms to use programming constructs from the GCSE curriculum. <u>Topics Covered:</u> • Working collaboratively using an online IDE. • Using lists to represent sequences of data. • Using strings to represent characters. • Manipulating strings with built in Python functions. • Using condition-controlled	**Data Representation Images and Sounds** Students build on their knowledge of data representation (binary, bitmap and vector graphics) to understand ways that computers can represent sounds. Students will learn how key terms sample rate, sample size and channels affect file size. Students will write Python programs to calculate file sizes for different data representations. <u>Topics covered:</u> • Data representation of characters • Data representation of numbers • Database schemas.	**Networks and the Internet** Students prepare for the GCSE curriculum by introducing the underlying platforms that enable the Internet – TCP-IP, HTTPS, and HTML. Students will explore how web pages are represented in HTML, CSS and JavaScript. <u>Topics covered.</u> • What are networks? • What is the Internet? • What are protocols? • Building a web page. • Using CSS to give a web page style.	**Developing mobile applications** Students put together all their knowledge and skills from the KS3 curriculum to develop a mobile phone application. <u>Topics Covered:</u> • Decomposing problems. • Using the event driven programming paradigm to get user responses. • Capturing user input. • Using variables to store data. • Updating display. • Using success criteria to evaluate.

	simulations and puzzles.	• Representing selection in flowcharts • Representing iteration in flowcharts	loops with while • Using count-controlled loops with for.	• Data representation of sounds. • Data representation of images.		
10	**Hardware** Students begin by examining the internal hardware components and how they function in a Von Neumann architecture. The three essential components are CPU, memory and storage. Students will explore the fetch-decode-execute cycle and cover hardware related topics <u>Topics Covered:</u> <u>Hardware</u> • CPU • Purpose of CPU • Purpose of components in the Von	**Programming Constructs** Students begin a long journey through the skills and knowledge required to use a general-purpose programming language – Python. As they gain skills, they will take on projects of increasing scope and challenge as individual, pair work or team projects. <u>Topics Covered:</u> <u>Python programming skills</u> • Input • Output • Selection • Iteration	**Networks** **Cyber Security** Students build on their knowledge of the standalone computer to explore the modern world of interconnected devices. This will lead to a study of the protocols required to exchange information on the Internet. Students must then consider cyber security threats and defences. <u>Topics Covered:</u> <u>Networking and cybersecurity</u> • Characteristics of networks • Advantages and	**Data representation** Students now have the prerequisite knowledge of hardware, logic and programming to consider how computers, which are electronic systems can represent information. From the basic data types of character and integer to floating point numbers, images, sound and video. <u>Topics Covered:</u> <u>Representation of characters</u> • Digital storage of characters • Unicode & ASCII	**Impacts and law** Students will now take a very high-level view looking at what software runs the computer, what are the impacts of this technology and what laws define computer use. <u>Topics Covered:</u> <u>Algorithms</u> Validation Verification <u>Operating systems</u> • Purpose of OS • Utility software • Purpose and functionality of range of software Impacts of digital	**Programming projects** Students tackle GCSE level practical programming projects. <u>Topics Covered:</u> • Programming using GUI • Programming project working individually Programming Project working in groups <u>Assessment</u> • Programming project • Mock Exams Component 1 & component 2

	Neumann CPU architecture including: ➢ Control unit ➢ Arithmetic and logic Unit ➢ Main memory ➢ Cache ➢ Registers ▪PC▪CIR▪ACC▪MAR▪MDR • Input and output devices • Fetch, decode and execute cycle • CPU performance in relation to: ➢ Cache size and levels of cache ➢ Clock speed ➢ Number of cores <u>Primary storage use and characteristics</u> • RAM • ROM • Flash memory	• Functions • File handling • Validation • Authentication • String handling <u>Data structures</u> • list, tuples, dictionary <u>Logical operators</u> • Logical operators: AND, OR, NOT, XOR • Boolean logic • Assessment • Programming project-in pairs • Knowledge recall logical operators <u>Cybersecurity</u> • Characteristics of different threats to computer systems: ➢ Malware ➢ Phishing ➢ Social engineering ➢ Brute force attacks	disadvantages of network • Network topologies: ➢ Star ➢ Mesh ➢ Ring ➢ Bus • Advantages and disadvantages over ring and bus Hardware required to establish wired and wireless connectivity including: ➢ Routers ➢ Hubs ➢ Switches ➢ Bridges ➢ WAP ➢ NIC <u>Internet</u> • The purpose of DNS and how they work • Structure of URLS and the role and function of a web browser	<u>Representation of graphics</u> • Digital storage of graphics • Effect of different resolution and colour depths on the quality of graphics • How to calculate storage requirements for graphics using different dimensions and colour depths. <u>Representation of sound</u> • Digital storage of sound • Effect of different sampling rates on the quality of sound • How to calculate the storage requirements for sound using different sampling rates <u>Compression</u>	technology on wider society • Impacts ➢ Ethical ➢ Legal ➢ Environmental ➢ Privacy issues • Legislation ➢ GDPR & Data protection 2018 ➢ Computer misuse act 1990 ➢ Copyright designs & patent act 1988 ➢ Creative commons licensing ➢ Regulation of investigatory powers act 2000 ➢ Telecommunications regulations act 2000 ➢ Freedom of information act 2000 • Professional standards	

	• Cache memory • Virtual memory <u>Additional hardware components</u> • GPUs • Sound cards • Motherboards <u>Secondary storage</u> • Optical • Magnetic • Solid state • Cloud storage • Suitability and characteristics of contemporary storage media <u>Embedded systems</u> • Characteristics and purpose of embedded systems • Examples of embedded systems <u>Algorithms</u> • Sequence	➢ Denial of service attacks ➢ Data interception and theft <u>SQL injection</u> • Different ways of protecting against threats during system, design, creation, testing & use: ➢ Penetration testing ➢ Network forensics ➢ Anti-malware software ➢ Firewalls ➢ User access levels ➢ Passwords ➢ Double authentication ➢ Encryption <u>Data representation and storage</u> <u>Representation of numbers</u>	<u>Cybersecurity</u> • Characteristics of different threats to computer systems: ➢ Malware ➢ Phishing ➢ Social engineer-ing ➢ Brute force attacks ➢ Denial of service attacks ➢ Data intercepti-on and theft <u>SQL injection</u> • Different ways of protecting against threats during system, design, creation, testing & use: ➢ Penetration testing ➢ Network forensics ➢ Anti-malware software ➢ Firewalls	• Lossy • Lossless • Calculation of compression ratio <u>Data Structures</u> • Records • One dimensional arrays • Two dimensional arrays <u>Principles of programming</u> • Level of computer language <u>Software tools</u> • IDEs • Program construction • Translators <u>Algorithms</u> • Searching • Sorting	<u>Assessment</u> • End of topic tests • Programming - Working in pairs on programming project	

	Selection • Iteration • Sorting • Searching • Variables • Counts • String handling • Validation • Verification • Flow charts using flowgorithm <u>Data types Assessment</u> • CPU written test • Algorithms- writing flow charts and pseudo code	• Use and convert between denary, binary & hexadecimal counting systems • Representation of positive and negative integers in a fixed length store using both twos complement and sign & magnitude representation • Binary addition & subtraction • Concept of overflow & underflow • Arithmetic shift function & their effect <u>Assessment</u> • End of topic test for network, data representation • Programming project Working on Betty's Bakery	➢ User access levels ➢ Passwords ➢ Double authentication ➢ Encryption <u>Data representation and storage Representation of numbers</u> • Use and convert between denary, binary & hexadecimal counting systems • Representation of positive and negative integers in a fixed length store using both twos complement and sign & magnitude representation • Binary addition & subtraction • Concept of overflow & underflow			

		and Magic numbers	• Arithmetic shift function & their effect <u>Assessment</u> • End of topic test for network, data representation • Programming project Working on Betty's Bakery and Magic numbers			
11	<u>Topics Covered:</u> Throughout Year 11 students will use the knowledge and understanding of the topics gained in Year 10 to apply that knowledge. Students will analyse problems in computational terms, make reasoned judgements, design, program, evaluate and refine solutions. (AO3)	<u>Topics Covered:</u> <u>Python programming skills</u> • Input • Output • Selection • Iteration • Functions • File handling • Validation • Authentication • String handling <u>Data structures</u> • list, tuples, dictionary	<u>Topics Covered:</u> <u>Networks</u> • Characteristics of networks • Advantages and disadvantages of network • Network topologies: ➢ Star ➢ Mesh ➢ Ring ➢ Bus • Advantages and disadvantages over ring and bus Hardware required to	<u>Topics Covered:</u> <u>Representation of characters</u> • Digital storage of characters • Unicode & ASCII • Representation of graphics • Digital storage of graphics • Effect of different resolution and colour depths on the quality of graphics • How to calculate storage	Revision Mocks	Exams

	Topics Covered: Hardware CPU • Purpose of CPU • Purpose of components in the Von Neumann CPU architecture including: ➢ Control unit ➢ Arithmetic and logic Unit ➢ Main memory ➢ Cache ➢ Registers ▪PC▪CIR▪ACC▪MAR▪MDR • Input and output devices • Fetch, decode and execute cycle • CPU performance in relation to:	Logical operators • Logical operators: AND, OR, NOT, XOR • Boolean logic Assessment • Programming project-in pairs • Knowledge recall logical operators • KAT: End of term test Cybersecurity • Characteristics of different threats to computer systems: ➢ Malware ➢ Phishing ➢ Social engineering ➢ Brute force attacks ➢ Denial of service attacks ➢ Data interception and theft	establish wired and wireless connectivity including: ➢ Routers ➢ Hubs ➢ Switches ➢ Bridges ➢ WAP ➢ NIC Internet • The purpose of DNS and how they work • Structure of URLS and the role and function of a web browser Cybersecurity • Characteristics of different threats to computer systems: ➢ Malware ➢ Phishing ➢ Social engineering ➢ Brute force attacks ➢ Denial of service attacks	requirements for graphics using different dimensions and colour depths. Representation of sound • Digital storage of sound • Effect of different sampling rates on the quality of sound • How to calculate the storage requirements for sound using different sampling rates Compression • Lossy • Lossless • Calculation of compression ratio • Data Structures • Records • One dimensional arrays • Two dimensional arrays		

	➢ Cache size and levels of cache ➢ Clock speed ➢ Number of cores <u>Primary storage</u> • use and characteristics • RAM • ROM • Flash memory • Cache memory • Virtual memory <u>Additional hardware components</u> • GPUs • Sound cards • Motherboards <u>Secondary storage</u> • Optical • Magnetic • Solid state • Cloud storage • Suitability and characteristics of contemporary	<u>SQL injection</u> • Different ways of protecting against threats during system, design, creation, testing & use: ➢ Penetration testing ➢ Network forensics ➢ Anti-malware software ➢ Firewalls ➢ User access levels ➢ Passwords ➢ Double authentication ➢ Encryption <u>Data representation and storage</u> <u>Representation of numbers</u> • Use and convert between denary, binary & hexadecimal counting systems	➢ Data interception and theft <u>SQL injection</u> • Different ways of protecting against threats during system, design, creation, testing & use: ➢ Penetration testing ➢ Network forensics ➢ Anti-malware software ➢ Firewalls ➢ User access levels ➢ Passwords ➢ Double authentication ➢ Encryptio <u>Data representation and storage</u> <u>Representation of numbers</u> • Use and convert between denary, binary	<u>Principles of programming</u> • Level of computer language • Software tools • IDEs <u>Program construction</u> • Translators <u>Algorithms</u> • Searching • Sorting		

	storage media ➢ Capacity ➢ Durability ➢ Portability ➢ Speed cost <u>Embedded systems</u> • Characteristics and purpose of embedded systems • Examples of embedded systems <u>Algorithms</u> • Sequence • Selection • Iteration • Sorting • Searching • Variables • Counts • String handling • Validation • Verification • Flow charts using flowgorithm	• Representation of positive and negative integers in a fixed length store using both twos complement and sign & magnitude representation • Binary addition & subtraction • Concept of overflow & underflow • Arithmetic shift function & their effect <u>Assessment</u> • End of topic test for network, data representation	& hexadecimal counting systems • Representation of positive and negative integers in a fixed length store using both twos complement and sign & magnitude representation • Binary addition & subtraction • Concept of overflow & underflow • Arithmetic shift function & their effect <u>Assessment</u> • End of topic test for network, data representation • Programming project Working on Betty's Bakery and Magic numbers			

Year 6 to 7 Transition:

Over the course of the academic year there is regular discussion with the Secondary Computing Curriculum Lead and Primary Phase Computing Lead. There are opportunities for cross-phase leaders to spend time in classrooms and observe cross-phase. This provides the opportunity for collaboration on content studied across the Key Stages and the skills that are developed. This ensures that at Key Stage 3 we can build on and develop the foundations laid at Key Stage 2.

Assessment for learning is used throughout lessons to ensure that the students have a sound understanding of the concepts taught. Effective modelling takes place at the outset of each task to ensure that students are able to progress regardless of starting points.

Year 7 pupils begin with a reminder of safe and responsible use of computer systems, the internet, and social media. Pupils will be given a half term course in digital literacy covering use of Windows PCs including MS Office and Teams. We teach these essential digital literacy skills at the start of the year to ensure that all pupils are starting with the relevant knowledge and skills to use the IT safely and effectively without any gaps as some of our pupils have joined us from different primary schools. Year 7 pupils are then introduced to data and programming concepts through Excel functions and formulas.

For students who have not selected computing at KS4:

All students should be taught to:

- develop their capability, creativity and knowledge in computer science, digital media and information technology

Non-computing subjects can apply IT to creative solutions. Students studying GCSE Drama use video editing software. GSCE Music students have access to a suite of PCs in the music room and use these to produce their compositions. Business students should be adept at using the Office suite. Design and Technology students will be able to use 2D and 3D design applications. Other subjects can use upper library facilities after school for computer-based homework.

- develop and apply their analytic, problem-solving, design, and computational thinking skills

Brebas challenge of computational thinking to be open to all students in second week of Autumn 2. Numerous subjects used various computer-based platforms to support their students in their learning. Quizlet, Seneca Learning, Sparx maths are all platforms used routinely with all students in the secondary phase.

- understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns

Digital Literacy is an integral part of our PSHCE curriculum, and this is studied in both tutor time sessions and hourly lesson in Year 7, 8, 9 and 12. Drop down days in Year 10 and 11 address changing issues such as sexting, influence of pornography, online grooming. Specialist facilitators discuss these sensitive issues with students, giving them a safe space to express themselves and ask questions. Key messaging on these topics is shared with students throughout the year in assemblies and tutor time.

Enrichment Opportunities:

The department runs a range of extracurricular clubs to develop problem solving and programming abilities. Within the Primary Phase students have the opportunity join extra-curricular clubs such as Touch Typing and Coding. Every student can access technology and resources at school by using computers and iPads as we are guided by the scheme iCompute within the Primary Phase. Every child is involved in both "plugged" and "unplugged" lessons whereby students have the correct balance between theoretical and practical lessons. Students can attend Code Club – run in conjunction with raspberrypi.org which offers three highly structured tracks of programming either in Scratch, HTML or Python, as well as a large number of project ideas.

Secondary Students can participate in an extra-curricular club to develop problem solving and programming abilities. Our current club is focused on robotics. Students have access to robotics kit from Lego Spike Prime. Students work together to develop, test and debug Lego robots to solve specific problems. There is an opportunity to compete against other teams through Lego's Competition Ready series of training. In the past students have used computing club to develop Scratch, Python and HTML application using raspberry pi's Code Club projects. Students have also been designing 3D models in TinkerCAD to print out using 3D printer. Finally, students can attend computer-based Chess club which uses lichess.org to enable blitz games, as well as tutorials, practice and game analysis.

Impact:

Formative assessment is an integral part of our approach to Teaching and Learning. Over the course of their study, we will use weekly cumulative formative diagnostic assessments (in class or for homework) to ensure that students are consistently retrieving their knowledge of different components. The purpose of this is to ensure all knowledge is retained (and any gaps are identified and addressed promptly) and to inform teachers' planning. Using this style of assessment, we will make use of the advantages of spaced practice as well as allowing students to be able to apply their knowledge to a wide variety of contexts.

In the Primary phase, teachers use assessment for learning within lessons to provide live feedback to allow pupils to deepen their understanding and identify gaps in knowledge and skills. Knowledge reviews are planned for spaced retrieval and allow for misconceptions to be addressed and further embed pupils understanding of key knowledge, skills and vocabulary. The progression of skills and knowledge allows teachers to assess the impact over the course of a unit, year and across phases. The scheme of learning is used to identify prior links

and future learning which informs teacher assessment and allows building blocks of learning to further develop schemas within topics and across subjects.

Summative assessments are used alongside knowledge organisers to assess the impact of learning at the end of a unit. This in turn informs future teaching adaptations, based on misconceptions and gaps in knowledge and skills. Enquiry questions are used to assess the impact of the teaching of knowledge, skills and vocabulary by allowing pupils to apply their understanding through reflections and critical thinking. In the Primary phase, students complete end of unit iCompute assessments. Assessment data allows teachers to see where students are in their learning and to identify any gaps in coverage, knowledge, understanding and skills which then informs the curriculum and future teaching.

In the Secondary phase, students will sit a summative assessment every full term. This assessment will be cumulative and will assess not only what the students have learned over the previous term, but also their understanding of all relevant material previously taught. Staff are supported to mark these accurately and post assessment moderation also takes place to ensure the validity of the data. All data is analysed centrally (not by teachers), and each Curriculum Leader is given a report outlining the areas of strength and weakness. Curriculum Leaders use this information to inform future planning, support with additional interventions and set changes.