

KS3 'Stages of Excellence': Computer Science

	Year 7	Year 8	Year 9
4. Extending	<p>Computational Thinking</p> <ul style="list-style-type: none"> • Decomposition: <ul style="list-style-type: none"> ○ Can consistently break down complex problems into smaller, manageable components and articulate a clear solution strategy. ○ Applies decomposition skills in a range of scenarios, from programming tasks to analysing real-world problems. • Pattern Recognition: <ul style="list-style-type: none"> ○ Identifies and uses patterns effectively to reduce complexity in problem-solving, recognizing commonalities between tasks and applying solutions across different contexts. • Abstraction: <ul style="list-style-type: none"> ○ Shows understanding of abstraction by removing unnecessary details and focusing on the critical parts of a problem. ○ Can explain the role of abstraction in computer science (e.g., in the design of computer systems and software). <p>Understanding Algorithms:</p> <ul style="list-style-type: none"> • Can explain the importance of algorithms in solving problems and use them effectively. • Able to design, trace, and evaluate algorithms, both independently and collaboratively. • Can create and modify algorithms using flowcharts, pseudocode, or similar tools, showing good logical thinking. <p>Understanding Components:</p> <ul style="list-style-type: none"> • Can describe the function and purpose of different computer hardware components (e.g., CPU, memory, hard drive, input/output devices) and how they interact in a system. 	<p>Networking Knowledge:</p> <ul style="list-style-type: none"> • Has a secure understanding of how networks are structured, including concepts like LANs (Local Area Networks), WANs (Wide Area Networks), and the internet. • Can explain how data is transmitted over a network using packets, including basic concepts such as IP addresses, routers, and protocols (e.g., HTTP, FTP). • <p>Internet Functionality:</p> <ul style="list-style-type: none"> • Understands the difference between the internet and the World Wide Web, and can explain how web pages are accessed using URLs, domain names, and DNS. <p>Problem-Solving & Creativity</p> <ul style="list-style-type: none"> • Problem-Solving: <ul style="list-style-type: none"> ○ Confidently uses computational thinking to approach new problems, designing creative and effective solutions. ○ Able to evaluate and refine solutions, learning from mistakes and improving their strategies. • Collaboration and Independence: <ul style="list-style-type: none"> ○ Works effectively both independently and as part of a team, contributing ideas and solving technical challenges collaboratively. ○ Demonstrates persistence in tackling challenging tasks, refining and improving their solutions based on feedback or testing results. <p>Binary and Number Systems:</p> <ul style="list-style-type: none"> • Can convert between binary, decimal, and hexadecimal number systems with confidence. • Understands how computers use binary to represent different types of data, such as numbers, characters (ASCII), and images (pixel representation). <p>Software:</p> <ul style="list-style-type: none"> • Can explain the features of system software (e.g., operating systems like Windows or macOS) and application software (e.g., word processors or web browsers). • Can clearly explain the purpose and usage of key types of software 	<p>Programming Proficiency:</p> <ul style="list-style-type: none"> • Can write more complex programs, using text-based languages like Python, confidently applying basic programming concepts such as loops, conditionals, and functions. • Understands and applies debugging techniques to identify and fix errors in programs independently. • Can write programs that use variables, arithmetic operations, and input/output to solve problems. • Understands the purpose and structure of functions and can use them to make code more efficient (e.g., defining and calling custom functions). <p>Impact of Technology</p> <ul style="list-style-type: none"> • Digital Safety: <ul style="list-style-type: none"> ○ Has a secure understanding of the importance of online safety, including managing privacy settings, avoiding phishing, and recognizing threats like malware and cyberbullying. ○ Understands the ethical issues related to data collection, social media, and the digital footprint, and can discuss how to protect personal information online. • Ethical and Social Impact: <ul style="list-style-type: none"> ○ Can discuss the broader social and ethical implications of computer science, such as automation, AI, and the digital divide. ○ Understands the importance of adhering to laws regarding intellectual property, data protection (e.g., GDPR), and other legal considerations related to technology use. <p>Evaluation:</p> <ul style="list-style-type: none"> • Can test solutions to ensure they work as intended and identify areas for improvement. <p>Evaluates the efficiency and correctness of solutions (e.g., comparing different algorithms or program designs for solving a problem). Understands how to refine and optimize solutions to improve performance or simplify code.</p>
3. Secure	<p>Computational Thinking</p> <ul style="list-style-type: none"> • Decomposition: Can break down a problem into smaller, manageable tasks. Understands that large problems can be simplified by tackling one part at a time. • Pattern Recognition: Can identify patterns or similarities in data or problems to simplify solutions. • Abstraction: Can ignore unnecessary details to focus on the important parts of a problem. 	<p>Basic Networking Concepts:</p> <ul style="list-style-type: none"> • Can explain what a network is and describe its benefits (e.g., sharing files, internet access). • Understands the difference between the internet and the World Wide Web. <p>Data Transmission:</p> <ul style="list-style-type: none"> • Understands that data is transmitted over networks in the form of packets. • Can explain the role of common network devices like routers and switches. <p>Software Basics:</p>	<p>Programming Proficiency:</p> <ul style="list-style-type: none"> • Can write basic programs, using text-based languages like Python, applying basic programming concepts such as loops and conditional. • Understands and applies debugging techniques to identify and fix errors in programs with occasional guidance. • Can make use of programs that use variables, arithmetic operations, and input/output to solve problems. <p>Impact of Technology</p>

	<p>Programming Fundamentals:</p> <ul style="list-style-type: none"> • Can write, test, and debug simple programs (typically in block-based or simple text-based languages like Python). • Understands basic control structures such as loops (e.g., for or while loops) and conditionals (e.g., if-else statements). • Can use variables to store and manipulate data. 	<ul style="list-style-type: none"> • Can differentiate between system software (e.g., operating systems like Windows or macOS) and application software (e.g., word processors or web browsers). • Can describe the purpose of key types of software. <p>Understanding Binary:</p> <ul style="list-style-type: none"> • Can fully explain the concept of binary as the basis of computer systems (how computers use 1s and 0s to represent data). • Can convert binary numbers to decimal and vice versa. 	<ul style="list-style-type: none"> • Digital Literacy: <ul style="list-style-type: none"> ○ Understands the importance of online safety, including issues related to privacy, data protection, and cyberbullying. • Ethical Use of Technology: <ul style="list-style-type: none"> ○ Can discuss the benefits and risks of technology in society, such as the impact of social media, the rise of artificial intelligence, or issues of digital divide. <p>Data Structures:</p> <ul style="list-style-type: none"> • Has a strong understanding of arrays (or lists) and can use them to store and manipulate collections of data within programs <p>Data Types and Manipulation:</p> <ul style="list-style-type: none"> • Can work with different data types (integers, floats, strings, booleans) in a program, ensuring appropriate usage in various contexts.
2.Developing	<p>Understanding Algorithms:</p> <ul style="list-style-type: none"> • Can explain what an algorithm is and understand how algorithms can be used for problem-solving. • Able to follow simple algorithms like flowcharts or pseudocode. • Can create basic sequences of instructions (e.g., in block-based programming environments like Scratch). <p>Basic Programming:</p> <ul style="list-style-type: none"> • Can write simple programs with guidance, using visual or block-based programming languages (e.g., Scratch or Blockly). • Understands the very basics of control structures such as sequential instructions (one after the other). • Can use simple instructions like "move forward" or "turn right" in basic programming tasks. <p>computer Components:</p> <ul style="list-style-type: none"> • Can identify basic components of a computer system, such as the CPU, memory (RAM), storage (hard drives/SSD), and input/output devices. • Understands the role of each component in processing and storing data. <p>Computational Thinking</p> <ul style="list-style-type: none"> • Decomposition: <ul style="list-style-type: none"> ○ Needs support to break down a problem into smaller parts. ○ Understands the idea that problems can be simplified but requires help in applying it. • Pattern Recognition: <ul style="list-style-type: none"> ○ Can identify basic patterns in simple data or problems when prompted. ○ Recognizes when something repeats in a process but struggles to use this observation to simplify tasks independently. • Abstraction: 	<p>Networks</p> <ul style="list-style-type: none"> • Can describe the role and importance of key network security measures, such as firewalls and encryption. • Has an understanding that computers can be connected in networks to share information and can explain main features of how this take place. • Can explain the term "internet" and that it allows access to websites and can explain how data is transferred over a network with some guidance. <p>Data Transmission:</p> <ul style="list-style-type: none"> • Can explain that data can be sent across a network with some technical details (e.g., packet switching or IP addresses). <p>collaboration and Independence</p> <ul style="list-style-type: none"> • Working with Others: <ul style="list-style-type: none"> ○ Participates in group work with guidance, sharing ideas and contributing to team activities with support. ○ Often relies on teacher or peer feedback to complete tasks and requires direction to work through challenge <p>Understanding Binary:</p> <ul style="list-style-type: none"> • Can explain the concept of binary as the basis of computer systems (how computers use 1s and 0s to represent data). • Can convert simple binary numbers to decimal and vice versa. <p>Basic Computer Components:</p> <ul style="list-style-type: none"> • Can name parts of a computer (e.g., monitor, keyboard, CPU) and has an understanding of their function. • Understands that software is the programs that run on a computer and can explain the difference between types of software (e.g., applications vs. operating systems) with some guidance. 	<p>Programming Proficiency:</p> <ul style="list-style-type: none"> • Can write basic programs, using text-based languages like Python, applying basic programming concepts such as loops and conditional. • Needs help debugging techniques to identify and fix errors in programs with guidance. • Can understand use of programs that use variables, arithmetic operations, and input/output to solve problems with some guidance <p>Data Structures:</p> <ul style="list-style-type: none"> • Has a basic understanding of arrays (or lists) and can use them to store and manipulate collections of data within programs <p>Data Types:</p> <ul style="list-style-type: none"> • Is aware that different types of data exist (e.g., numbers, text) and with examples can understand when and how to use them in programs. <p>Evaluation:</p> <ul style="list-style-type: none"> • Can test solutions to ensure they work with some guidance. • Can identify areas of improvement but may not know exactly how until guided

	<ul style="list-style-type: none"> ○ Understands that some details are unnecessary for solving a problem but finds it difficult to identify what to ignore on their own. 		
1.Novice	<p>Understanding Algorithms:</p> <ul style="list-style-type: none"> • Recognizes that an algorithm is a set of instructions to solve a problem. • Can follow a simple sequence of instructions provided to them (e.g., a step-by-step process for making a sandwich or navigating a maze). • Requires support to create simple algorithms (e.g., in block-based programming environments like Scratch). <p>Approaching Problems:</p> <ul style="list-style-type: none"> • With support, can follow a structured approach to solve simple problems, such as following a step-by-step plan or using trial and error. • Requires guidance to come up with solutions or alternative approaches to problems. <p>Creativity in Computing:</p> <ul style="list-style-type: none"> • Can complete guided activities where creativity is encouraged (e.g., creating a simple game in Scratch) but needs assistance to explore or create more independently. <p>Data Types:</p> <ul style="list-style-type: none"> • Understands basic data types like integers, floats (decimals), and strings (text). • Can explain the difference between these types and when to use them. 	<p>Understanding Binary:</p> <ul style="list-style-type: none"> • Knows that computers use 1s and 0s (binary) to represent data but may not fully understand how this works. • Needs help converting very simple binary numbers to decimal (e.g., 4-bit numbers). <p>Basic Computer Components:</p> <ul style="list-style-type: none"> • Can name some parts of a computer (e.g., monitor, keyboard, CPU) and has a basic understanding of their function. • Understands that software is the programs that run on a computer but may not yet be clear on the difference between types of software (e.g., applications vs. operating systems). <p>System Usage:</p> <ul style="list-style-type: none"> • Can use basic system software (e.g., opening and saving files, using simple applications) with support. • Requires guidance to understand the function of common hardware components like RAM or storage. <p>Understanding Networks:</p> <ul style="list-style-type: none"> • Has a basic understanding that computers can be connected in networks to share information. • Knows the term "internet" and that it allows access to websites but may not understand how data is transferred over a network. <p>Data Transmission:</p> <ul style="list-style-type: none"> • Can explain that data can be sent across a network but requires help understanding the details (e.g., packet switching or IP addresses). <p>collaboration and Independence</p> <ul style="list-style-type: none"> • Working with Others: <ul style="list-style-type: none"> ○ Participates in group work with guidance, sharing ideas and contributing to team activities with support. ○ Often relies on teacher or peer feedback to complete tasks and requires direction to work through challenge 	<p>Data Types:</p> <ul style="list-style-type: none"> • Has some awareness that different types of data exist (e.g., numbers, text) but requires guidance in understanding when and how to use them in programs. <p>online Safety Awareness:</p> <ul style="list-style-type: none"> • Knows basic online safety rules (e.g., don't share passwords or personal information) and can explain why they are important. • Needs guidance to understand more complex issues like cyberbullying or data privacy. <p>Ethical Use of Technology:</p> <ul style="list-style-type: none"> • Understands that there are rules for using technology responsibly (e.g., respecting copyright or using social media wisely) but may need support to discuss the wider implications.