Roundwood Park School

Computer Science

Skills Descriptors

Computers in Society

Computational Thinking

Data

Computer Science – Computers in Society

YEAR 7 – Computer Systems (Computer Systems, Peripherals, OS, Memory, Secondary Storage, Utility Software, Application Software)	YEAR 8 – Ethics (Digital Divide, Careers, Environment)	YEAR 9 – Cyber Security (Cyber Security, Malicious Code, Social Engineering, Detection and Prevention, Biometrics)
		MASTERING + Working above a mastering level
	MASTERING + Working above a mastering level	**MASTERING** The student can explain what cyber security is and the importance of studying it. They can define social engineering and give examples. They can define and describe malicious code with examples and justify the need for protection. The student has an awareness of the different methods that can detect and prevent cyber security threats.
MASTERING + Working above a mastering level	**MASTERING** The student can explain the causes of the digital divide, why there is a gap between those who have access to technology and those who don't and why it is important. They can suggest ideas on how to bridge the gap and minimise the effect it has. The student can explain how useful computer science is in a range of careers. They can describe the effect of e-waste on the environment and how to minimise this effect.	**SECURING** The student can explain what cyber security is and the importance of studying it. They can define social engineering and give examples. They can define and describe malicious code with examples and justify the need for protection. The student has an awareness of the different methods that can detect and prevent cyber security threats.
MASTERING The student can identify and explain the functions of internal and external components of a computer system. The student can explain what is meant by clock speed and how this affects the performance of the CPU. The student can define 3 types of secondary storage, and the pros and cons of each type. They can identify and explain the 5 main tasks of an operating system and identify the utility software tasks required to keep a computer system running smoothly. The student can identify the purpose of different types of application software and why there is the need for different variations.	**SECURING** The student can explain why there is a gap between those who have access to technology and those who don't and why it is important. They can suggest ideas on how to bridge the gap and minimise the effect it has. The student can explain how useful computer science is in a range of careers. They can describe the effect of e-waste on the environment and how to minimise this effect.	**DEVELOPING** The student can explain what cyber security is and the importance of studying it. They can define social engineering and identify an action that would classify as it. They can define and describe malicious code with examples and justify the need for protection. The student has an awareness of the different methods that can detect and prevent cyber security threats.
SECURING The student can identify the internal components of a computer system. They can identify all the peripheral components of a computer system and identify if they are input, output or storage. The student can define 3 types of secondary storage and can identify the 5 main tasks of an operating system. They can identify 3 types of utility software and explain their functions. They can explain the difference between proprietary and open source software.	**DEVELOPING** The student can explain why there is a gap between those who have access to technology and those who don't and why it is important. They can suggest ideas on how to bridge the gap and minimise the effect it has. The student can explain how useful computer science is in a range of careers. They can describe the effect of e-waste on the environment and how to minimise this effect.	**ACQUIRING** The student can explain what cyber security is. They can define social engineering and identify an action that would classify as it. They can define and describe malicious code with an example and justify the need for protection. The student has an awareness of the different methods that can detect and prevent cyber security threats.
DEVELOPING The student can explain what is meant by an embedded system and can identify all the peripheral components of a computer system. The student knows that ROM is non-volatile and RAM is volatile memory and explains the function of each. They can identify the 3 types of secondary storage and 2 types of operating system (and explain how they are different).	**ACQUIRING** The student can explain why there is a gap between those who have access to technology and those who don't. They can describe the effect of e-waste on the environment and suggest ways to minimise this effect. They can suggest ideas on how to bridge the gap. The student can explain how useful computer science is in their aspirational career.	
ACQUIRING The student knows that a computer system requires an input, a process and an output. The student can identify 4 essential peripherals and distinguish between ROM (non-volatile) and RAM (volatile).. The student can identify the purpose of the operating system (giving an example). They can explain what utility software and application software are and give an example of each.		

Computer Science – Computational Thinking

YEAR 7 – Python Turtle (Basic Commands, Colour, For Loops, Nested Loops, Procedures)	YEAR 8 – Python (Variables, Data Types and Operations, If Statements, For Loops, While Loops, Random)	YEAR 9 – DIT (Navigational Structure, Accessibility, Wireframes, Usability, WCAG)
	MASTERING + Working above a mastering level	**MASTERING +** Working above a mastering level
	MASTERING + Working above a mastering level	**MASTERING** The student can create a detailed, well-structured plan and effective navigation system, source and store a range of multimedia elements. They can successfully create a workable product that makes effective use of: alternative pathways, hyperlinks, user interaction and multimedia effects. The student can identify the aim and audience of a multimedia product giving detailed explanations of features, their effectiveness and suggest improvements where possible of an existing product and their own using user testing and evaluations.
MASTERING + Working above a mastering level	**MASTERING** The student has a practical experience of a high-level textual language, including using standard libraries when programming. They can declare and assign variables knowing their data type. The student understands and appropriately uses loops, if, if else and elif statements. They can use flow-diagrams to express solutions and apply logical reasoning to predict outputs and detect/correct simple semantic errors .	**SECURING** The student can create a detailed, well-structured plan and effective navigation system, source and store some multimedia elements. They can successfully create a product that makes effective use of: alternative pathways, hyperlinks, user interaction and multimedia effects where most elements work as intended. The student can identify the aim and audience of a multimedia product giving detailed explanations of features, their effectiveness and suggest improvements where possible of an existing product and their own using user testing and evaluations.
MASTERING The student understands that computers need precise instructions and can demonstrate care and precision to avoid errors. They can write basic commands to draw images and change the colour of the pen. The student understands how iteration and nested loops can be used to repeat instructions thus creating a plethora of complex shapes. The student can develop their own procedures, apply logical reasoning to predict outcomes and detect/correct errors.	**SECURING** The student has a practical experience of a high-level textual language, including using standard libraries when programming. They can declare and assign variables knowing their data type. The student understands and appropriately uses loops, if, if else statements. They can use flow-diagrams to express solutions and predict outputs and detect and attempt to correct simple semantic errors.	**DEVELOPING** The student can create a structured plan and navigation system, source and store some multimedia elements. They can create a product that makes effective use of: alternative pathways, hyperlinks, user interaction and multimedia effects where some elements work as intended. The student can identify the aim and audience of a multimedia product giving some explanations of features, their effectiveness and suggest improvements of an existing product and their using evaluations.
SECURING The student understands that computers need precise instructions and can demonstrate care and precision to avoid errors. They can write basic commands to draw images and change the colour of the pen. The student understands how iteration and nested loops can be used to repeat instructions to make basic shapes and can apply logical reasoning to predict outcomes and detect/correct errors.	**DEVELOPING** The student has a practical experience of a high-level textual language. They can declare and assign variables knowing their data type. The student can use if and if, else statements. They can use flow-diagrams to express solutions and can use logical reasoning to predict the behaviour of programs	**ACQUIRING** The student can create a plan and navigation system, source and store some multimedia elements. They can create a product that makes effective use of: alternative pathways, hyperlinks, user interaction and multimedia effects where few elements work as intended. The student can identify the aim and audience of a multimedia product giving some explanations of features, their effectiveness and suggest improvements of an existing product and their using evaluations.
DEVELOPING The student understands that computers need precise instructions and can demonstrate care and precision to avoid errors. They can write basic commands to draw images. The student understands how iteration can be used to repeat instructions to create basic shapes and can apply logical reasoning to predict outcomes and detect/correct errors.	**ACQUIRING** The student can use arithmetic operators, if statements, and loops, within programs. They can declare and assign variables knowing their data type. They can use flow-diagrams to express solutions. They can use logical reasoning to predict the behaviour of programs .	
ACQUIRING The student understands that computers need precise instructions and can demonstrate care and precision to avoid errors. They can write basic commands to draw images and change the colour of the pen. The student understands how iteration can be used to repeat instructions thus creating basic shapes and can apply logical reasoning to predict outcomes and detect errors.		

Computer Science – Data

YEAR 7 - Data Representation	YEAR 8 - Data Representation	YEAR 9 - SQL
		MASTERING + Working above a mastering level
	MASTERING + Working above a mastering level	**MASTERING** The student can explain the concept of a database and a relational database. They can define and explain the key concepts within a database (ie Table, Record, Field…). They are able to use SQL commands to retrieve data from a database (SELECT, FROM, WHERE, ORDER BY).
MASTERING + Working above a mastering level	**MASTERING** The student can convert from English to ASCII. The student can work out how many bits are needed to represent an image of 16 colours, and The student can calculate the file size. The student can shift a byte with a decimal point 2 places left or right and convert into denary and vice versa. The student can add together 3 bytes and know what is meant by the term Overflow Error. They know that computers transfer data in binary and understand the relationship between binary and file size (uncompressed). The student performs more complex searches and sorts for information.	**SECURING** The student can explain the concept of a database and a relational database. They understand the key concepts within a database (ie Table, Record, Field…). They can use SQL commands to retrieve data from a database.
MASTERING The student can convert between more complex sizes of data storage (b, B, KB, MB, GB, TB). They can convert a byte into denary and vice versa without using the grid. The student can add together 2 bytes and know what is meant by the term Overflow Error. The student can convert a byte with a decimal point into denary and vice versa. They can convert a byte into 2 hexadecimal digits and vice versa. They can perform simple operations using bit patterns such as binary addition.	**SECURING** The student can explain why extended ASCII and Unicode were developed. The student can work out how many bits are needed to represent an image of 8 colours, and the student can calculate the file size. The student can shift a byte 2 places to the left or right and convert the number into denary, including decimal places. The student can add together 3 bytes. The student can perform more complex searches and sorts for information.	**DEVELOPING** The student can explain the concept of a database. They can define the key concepts within a database (ie Table, Record, Field…). They can use SQL commands to retrieve data from a database.
SECURING The student can convert between different sizes of data storage (B, KB, MB, GB, TB). They can convert a byte into denary and vice versa using the grid The student can add together 2 bytes. They can shift a byte to the left or right and understand that there can be decimal points in binary numbers. The student can convert a binary nibble into hexadecimal and vice versa. They can perform simple operations using bit patterns such as binary addition.	**DEVELOPING** The student can write their name in binary. The student can work out how many bits are needed to represent an image of 4 colours, and they can calculate the file size. The student can shift a simple byte 2 places to the left and right and can convert them to denary. The student can add together 2 bytes. Student performs more complex searches and sorts for information.	**ACQUIRING** The student can explain the concept of a database. They can define the key concepts within a database (ie Table, Record, Field…). They can use basic SQL commands to retrieve data from a database.
DEVELOPING The student can name the different sizes of data storage and know the size order from smallest to largest. They can convert a nibble to denary and vice versa. The student can add together 2 4-bit numbers. The student can shift a simple byte to the left and right and can convert them to denary. They can convert the numbers 1-15 into hexadecimal and vice versa.	**ACQUIRING** The student knows that each letter of the alphabet is given a binary value. The student knows that colours are represented by combinations of bits. The student knows that 2 shifts left multiplies a binary number by 22 and 2 shifts right divides it by 22. They can add together 2 4-bit numbers. The student can perform more complex searches and sorts for information.	
ACQUIRING The student knows that digital computers use binary to represent all data. The student knows that a bit is a single binary digit and that a byte is made up of 8 bits. They know that binary is base 2 and uses the digits 1 and 0. The student can add together 2 2-bit numbers. They know that 1 shift left multiplies a binary number by 2 and a shift right divides it by 2. The student knows that hexadecimal is base 16 and uses the digits 0-9, A-F.		