

Roundwood Park School

Computer Science



Skills descriptors

Algorithms

Programming & Development

Data & Data Representation

Information Technology

Computer Science – Algorithms

		YEAR 9	
		YEAR 8	MASTERING + Working above a mastering level
YEAR 7		MASTERING + Working above a mastering level	MASTERING Student can understand a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. They can recognise that some problems share the same characteristics and use the same algorithm to solve both. Understands the notion of performance for algorithms and appreciates that some algorithms have different performance characteristics for the same task.
MASTERING + Working above a mastering level	MASTERING Student can explain that iteration is the repetition of a process such as a loop. They recognise that different algorithms exist for the same problem. They can represent solutions using a structured notation. They can identify similarities and differences in situations, identify the most efficient and can use these to solve problems (pattern recognition).	SECURING Student can explain that iteration is the repetition of a process such as a loop. They recognise that different algorithms exist for the same problem. They can represent solutions using a structured notation. They can identify similarities and differences in situations, identify the most efficient and can use these to solve problems (pattern recognition).	
MASTERING Student shows an awareness of tasks best completed by humans or computers. They can recognise that different solutions exist for the same problem.	SECURING Student shows an awareness of tasks best completed by humans or computers. They can recognise that different solutions exist for the same problem.	DEVELOPING Student shows an awareness of tasks best completed by humans or computers. They can recognise that different solutions exist for the same problem.	
SECURING Student can design solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. They can use diagrams to express solutions and apply logical reasoning to predict outputs.	DEVELOPING Student can design solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. They can use diagrams to express solutions and apply logical reasoning to predict outputs.	ACQUIRING Student can design solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. They can use diagrams to express solutions and apply logical reasoning to predict outputs.	
DEVELOPING Student understands that algorithms are implemented on digital devices as programs. They can design simple algorithms using loops, and selection i.e. if statements. They can apply logical reasoning to predict outcomes and, detect and corrects errors i.e. debugging, in algorithms.	ACQUIRING Student understands that algorithms are implemented on digital devices as programs. They can design simple algorithms using loops, and selection i.e. if statements. They can apply logical reasoning to predict outcomes and, detect and corrects errors i.e. debugging, in algorithms.		
ACQUIRING Student understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. They understand that computers need precise instructions and can demonstrate care and precision to avoid errors.			

Computer Science – Programming & Development

		YEAR 9	
		YEAR 8	MASTERING + Working above a mastering level
YEAR 7		MASTERING + Working above a mastering level	MASTERING Student is able to use nested selection statements and can appreciate the need for, and writes, custom functions including use of parameters. They know the difference between, and uses appropriately, procedures and functions. They understand and use negation with operators. They can use and manipulates one dimensional data structures and within programs can detect and correct syntactical errors.
MASTERING + Working above a mastering level		MASTERING Student understands that programming bridges the gap between algorithmic solutions and computers. They have practical experience of a high-level textual language, including using standard libraries when programming. They are able to use a range of operators and expressions e.g. Boolean, and applies them in the context of program control. They can selects appropriate data types.	SECURING Student understands that programming bridges the gap between algorithmic solutions and computers. They have practical experience of a high-level textual language, including using standard libraries when programming. They are able to use a range of operators and expressions e.g. Boolean, and applies them in the context of program control. They can selects appropriate data types.
MASTERING Student can understand the difference between, and appropriately uses if and if, then and else statements. They are successfully able to use a variable and relational operator within a loop to govern termination. They can design, write and debug modular programs using procedures. They knows that a procedure can be used to hide the detail with sub-solution.		SECURING Student can understand the difference between, and appropriately uses if and if, then and else statements. They are successfully able to use a variable and relational operator within a loop to govern termination. They can design, write and debug modular programs using procedures. They knows that a procedure can be used to hide the detail with sub-solution.	DEVELOPING Student can understand the difference between, and appropriately uses if and if, then and else statements. They are successfully able to use a variable and relational operator within a loop to govern termination. They can design, write and debug modular programs using procedures. They knows that a procedure can be used to hide the detail with sub-solution.
SECURING Student can create programs that implement algorithms to achieve given goals. They can declare and assigns variables. They can use post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.		DEVELOPING Student can create programs that implement algorithms to achieve given goals. They can declare and assigns variables. They can use post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.	ACQUIRING Student can create programs that implement algorithms to achieve given goals. They can declare and assigns variables. They can use post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.
DEVELOPING Student can use arithmetic operators, if statements, and loops, within programs. They can use logical reasoning to predict the behaviour of programs and detect and correct simple semantic errors i.e. debugging, in programs.		ACQUIRING Student can use arithmetic operators, if statements, and loops, within programs. They can use logical reasoning to predict the behaviour of programs and detect and correct simple semantic errors i.e. debugging, in programs.	
ACQUIRING Student knows that users can develop their own programs, and can demonstrate this by creating a simple program. They understand that programs execute by following precise instructions. They can execute a program, checking for errors.			

Computer Science – Data & Data Representation

		YEAR 9	
		YEAR 8	MASTERING + Working above a mastering level
YEAR 7		MASTERING + Working above a mastering level	MASTERING Student understand how numbers, images, sounds and character sets use the same bit patterns. They can perform simple operations using bit patterns such as binary addition. They understand the relationship between resolution and colour depth, including the effect on file size. They can distinguish between data used in a simple program (a variable) and the storage structure for that data.
MASTERING + Working above a mastering level		MASTERING Student knows that digital computers use binary to represent all data. They understand how bit patterns represent numbers and images. They know that computers transfer data in binary and understand the relationship between binary and file size (uncompressed). They can define data types: real numbers and Boolean. They can interrogate data.	SECURING Student knows that digital computers use binary to represent all data. They understand how bit patterns represent numbers and images. They know that computers transfer data in binary and understand the relationship between binary and file size (uncompressed). They can define data types: real numbers and Boolean. They can interrogate data.
MASTERING Student performs more complex searches for information e.g. using Boolean and relational operators. They can analyse and evaluate data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions.		SECURING Student performs more complex searches for information e.g. using Boolean and relational operators. They can analyse and evaluate data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions.	DEVELOPING Student performs more complex searches for information e.g. using Boolean and relational operators. They can analyse and evaluate data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions.
SECURING Student understands the difference between data and information. They know why sorting data in a flat file can improve searching for information. They are able to use filters or can perform single criteria searches for information.		DEVELOPING Student understands the difference between data and information. They know why sorting data in a flat file can improve searching for information. They are able to use filters or can perform single criteria searches for information.	ACQUIRING Student understands the difference between data and information. They know why sorting data in a flat file can improve searching for information. They are able to use filters or can perform single criteria searches for information.
DEVELOPING Student can recognise different types of data: text, number. They appreciate that programs can work with different types of data. They recognises that data can be structured in tables to make it useful.		ACQUIRING Student can recognise different types of data: text, number. They appreciate that programs can work with different types of data. They recognises that data can be structured in tables to make it useful.	
ACQUIRING Student recognises that digital content can be represented in many forms. They can distinguish between some of these forms and can explain the different ways that they communicate information.			

Computer Science – Information Technology

		YEAR 9	
YEAR 8		MASTERING + Working above a mastering level	
		MASTERING Student justifies the choice of and independently combines and uses multiple digital devices, internet services and application software to achieve given goals. They can evaluate the trustworthiness of digital content and considers the usability of visual design features when designing and creating digital artefacts for a known audience. They can Identify and explain how the use of technology can impact on society. They are able to design criteria for users to evaluate the quality of solutions, uses the feedback from the users to identify improvements and can make appropriate refinements to the solution.	
YEAR 7s	MASTERING + Working above a mastering level		SECURING Student can evaluate the appropriateness of digital devices, internet services and application software to achieve given goals. Recognises ethical issues surrounding the application of information technology beyond school. Designs criteria to critically evaluate the quality of solutions, uses the criteria to identify improvements and can make appropriate refinements to the solution.
	MASTERING Student makes judgements about digital content when evaluating and repurposing it for a given audience. They recognise the audience when designing and creating digital content. They use criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions.		DEVELOPING Student makes judgements about digital content when evaluating and repurposing it for a given audience. They recognise the audience when designing and creating digital content. They use criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions.
	SECURING Student collect, organises and presents data and information in digital content. They can create digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. They make appropriate improvements to solutions based on feedback received, and can comment on the success of the solution.		ACQUIRING Student collect, organises and presents data and information in digital content. They can create digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. They make appropriate improvements to solutions based on feedback received, and can comment on the success of the solution.
	DEVELOPING Student can use technology with increasing independence to purposefully organise digital content. They show an awareness for the quality of digital content collected. Student can use a variety of software to manipulate and present digital content: data and information. Shares their experiences of technology in school and beyond the classroom. Talks about their work and makes improvements to solutions based on feedback received.		ACQUIRING Student can use technology with increasing independence to purposefully organise digital content. They show an awareness for the quality of digital content collected. Student can use a variety of software to manipulate and present digital content: data and information. Shares their experiences of technology in school and beyond the classroom. Talks about their work and makes improvements to solutions based on feedback received.
	ACQUIRING Student uses software under the control of the teacher to create, store and edit digital content using appropriate file and folder names. They understand that people interact with computers. They are able to share their use of technology in school. They know common uses of information technology beyond the classroom. They are able to talk about their work and makes changes to improve it.		