Roundwood Park School

Computer Science

Knowledge Outline for KS3 Computer Science

At KS3 we aim to:

- Equip pupils to be able to use computational thinking and creativity to understand the digital world, and to use information technology to create programs, systems and a range of content.
- Teach the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming and developing an understanding of the role of software development.
- Incorporate individual and collaborative activities that empower students to build on this knowledge and understanding.
- Ensure that pupils become digitally literate and responsible; able to use, and express themselves and develop their ideas through information and communication technology, at a level suitable for the future workplace and as active participants in a digital world.

Our curriculum allows students to question, explore, apply and gain knowledge of the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

Throughout the curriculum students will gain an understanding of how to analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems.

Students will be able to evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems and they will become responsible, competent, confident and creative users of information and communication technology.

Rationale

The plan is based on:

- 3 recurring themes which are Computers in Society, Data Representation and Programming across KS3.
- development of skills in preparation for the AQA GCSE at KS4
- using computer science to promote good practice in a digital world.

<table>
<tr><th>Year / part</th><th>Unit of work</th><th>Core Knowledge Year 7</th><th>Intent and Skills of Year 7 Computer Science</th></tr>
<tr><td>YEAR 7 part 1</td><td>Introduction to RPS IT</td><td>Be able to set up folders in their area (N:\ )
Be able to access RMunify at school and at home
Be able to access showmyhomework
Be able to access the google drive
Be able to access the google classroom
Be able to confidently utilise the G-Suite</td><td rowspan="5">Students will gain an understanding of what constitutes a computer and the roles of the different components that make up modern computers including the variety of hardware and the different types of software.

They will be able to explain the basics on computational thinking including abstraction, decomposition, patterns recognition and algorithms.

They will be able to explain what bits and bytes are as well as convert between the different sizes including Kilobyte and Megabyte.

They will be able to convert between denary, binary and hexadecimal.

Students will have opportunities to plan, monitor and improve own learning and work, be independent learners, listen to others and ask questions to extend understanding.</td></tr>
<tr><td colspan="3">ASPIRE: Responsibility, Insight, Being Resourceful
Reading: RPS responsible user agreement</td></tr>
<tr><td>YEAR 7 part 2</td><td>Programming Basics</td><td>Be able to predict the outcome of a block-based algorithm / program
Be able to amend block-based algorithms / programs
Be able to create block-based algorithms / programs
Be able to describe abstraction, decomposition and pattern recognition
Be able to identify different types of algorithms
Be able to create different types of algorithms
Be able to describe and apply sequence and iteration
Be able to apply Boolean and arithmetic operators
Be able to debug basic errors including syntax and logical</td></tr>
<tr><td colspan="3">ASPIRE: All of learning to understand, Planning ahead, Reflecting, Risk Taking, Collaboration, Communication, Perseverance
Challenge: Debugging, variable declaration, iteration
Reading: Kubica, J. (2012). Computational fairy tales.
Super Curricular:
<u>Read:</u> Scratch programming books
<u>Watch:</u> Youtube videos on scratch app making
<u>Do:</u> Create own programs using Scratch</td></tr>
<tr><td>YEAR 7 part 3</td><td>Computer Hardware</td><td>Be able to explain the difference between hardware and software
Be able to identify and explain peripherals
Be able to describe the role of internal components including the CPU (including the FDE cycle), Storage and Main Memory
Be able to compare and contrast components / machines by specification
Be able to use basic logic gates and complete truth tables</td></tr>
<tr><td colspan="3">ASPIRE: Learning Respectfully, Insight, Evaluating, Questioning
Challenge: CPU including MAR, MDR etc
Reading: Clark-Scott, J. (2009). But how do it know?. 1st ed.
Super Curricular:
<u>Read</u> Clark-Scott, J. (2009). But how do it know?.; Kubica, J. (2012). Computational fairy tales.</td></tr>
</table>

<table>
<tr><td colspan="3"><u>Watch:</u> YouTube videos from Craig and Dave
<u>Research:</u> What is the von Neumann bottle neck and why is it important to know about it now.
What is quantum computing and why is it needed?
What are the implications of quantum computing?</td><td rowspan="2"></td></tr>
<tr><td>YEAR 7
part 4</td><td>Data representation</td><td>Be able to explain the role of binary
Be able to describe bits and bytes
Be able to describe how numbers are represented in bits and bytes
Be able to convert between bits, nibbles, bytes, kilobytes and megabytes
Be able to convert between denary, binary and hexadecimal with accuracy
Be able to explain why binary and hexadecimal numbers are needed</td></tr>
<tr><td colspan="4">ASPIRE: Insight, Making Links, Risk taking, Organising, Logical Thinking, Valuing Others
Challenge: Mathematical applications
Reading: Kubica, J. (2012). Computational fairy tales.
Super Curricular:
<u>Read:</u> Kubica, J. (2012). Computational fairy tales.
<u>Do:</u> Create your own crossword using binary and hexadecimal numbers</td></tr>
</table>

<table>
<tr><th>Year / park</th><th>Unit of work</th><th>Core Knowledge Year 8</th><th>Intent and Skills of Year 8 Computer Science</th></tr>
<tr><td>YEAR 8 part 1</td><td>Data Representation</td><td>Be able to recap and explain the role of binary
Be able to recap and explain how numbers are represented in bits and bytes
Be able to recap and convert between bits, nibbles, bytes, kilobytes and megabytes
Be able to recap and convert between denary, binary and hexadecimal with accuracy
Be able to recap logic gates and truth tables
Be able to explain how letters are represented in ASCII
Be able to describe the need for Extended ASCII and Unicode
Be able to explain how bitmap images are made up
Be able to describe the relationship between the number of colours available and the number of binary digits needed
Be able to calculate image file sizes in bits, bytes, kilobytes
Be able to identify factors affecting file size</td><td rowspan="4">Students will build on their understanding programming basics including computational thinking such as abstraction, decomposition, patterns recognition and algorithms.

They will be able to create algorithms to solve specific problems and then implement these within programming languages.

They will be able to recall what bits and bytes are as well as convert between the different sizes including Kilobyte and extend their knowledge up to Gigabyte

They will be able to recall how to convert between denary, binary and hexadecimal.

They will the able to explain how images are represented using binary digits and the relationship between the number of bits and the size of the file.

Students will be able to explain moral, legal and ethical issues relating to computer science. They will seek to expand their understanding of the world around them be able to explain and in some cases argue specific positions in relation to these issues.

Students will have opportunities to plan, monitor and improve own learning and work, be independent learners, listen to others and ask questions to extend understanding.</td></tr>
<tr><td colspan="3">ASPIRE: Insight, Making Links, Risk taking, Organising, Logical Thinking, Valuing Others
Challenge: Mathematical applications
Reading: Kubica, J. (2012). Computational fairy tales.
Super Curricular:
<u>Read:</u> Kubica, J. (2012). Computational fairy tales.
<u>Do:</u> Create your own crossword using binary and hexadecimal numbers</td></tr>
<tr><td>YEAR 8 part 2</td><td>Algorithms and Programming</td><td>Be able to predict the outcome of a text-based algorithm / program
Be able to amend text-based algorithms / programs
Be able to create text-based algorithms / programs to solve specific problems
Be able to describe and apply sequence, selection and iteration
Be able to apply Boolean and arithmetic operators
Be able to debug basic errors including syntax and logical
Be able to identify functions within python</td></tr>
<tr><td colspan="3">ASPIRE: All of learning to understand, Planning ahead, Reflecting, Risk Taking, Collaboration, Communication, Perseverance
Challenge: Be able to define the term subroutine and implement it within a program
Reading: Kubica, J. (2012). Computational fairy tales.
Super Curricular:
<u>Read:</u> Kubica, J. (2012). Computational fairy tales.
<u>Do:</u> Create own programs using python turtle.</td></tr>
<tr><td>YEAR 8 part 3</td><td>Rights and Responsibilities</td><td>Be able to describe the need digital divide
Be able to explain the impact that the lack of access to technology has in different scenarios
Be able to identify and explain the impact of online communications
Be able to explain the Computer Misuse Act and how it relates to online communications
Be able to describe the need for these laws and how they protect individuals</td></tr>
</table>

		Be able to explain how computer science is changing industries and the links with careers Be able to identify and describe the environmental impacts of computers Be able to identify and describe ways or methods of minimising environmental impacts	
ASPIRE: Learning Respectfully, Being Resourceful, Organising, Evaluating, Responsibility, Questioning, Making Links **Challenge:** **Reading:** Current news articles, Academic Articles **Super Curricular:** **Read:** Kubica, J. (2012). *Computational fairy tales*. **Do:** Write an article for Roundup on one of the following: Implications of AI The digital divide and how we should address the issue E-Waste and what should be done about it How to stay safe online communications platforms such as social media Write a short play on the dangers of cyber bullying **Watch**: Big Hero 6,			

<table>
<tr><th>Year / part</th><th>Unit of work</th><th>Core Knowledge Year 9</th><th rowspan="1">Intent and Skills of Year 9 Computer Science</th></tr>
<tr><td>YEAR 9 part 1</td><td>Cyber Security</td><td>Be able to define cyber security
Be able to explain the need for cyber security
Be able to identify the threats to computer systems
Be able to explain the different threats to computer systems
Be able to describe ways of minimising the threats including biometrics
Be able to draw links between the Rights and Responsibilities unit in Year 8</td><td rowspan="5">Students will build on their understanding programming basics including computational thinking such as abstraction, decomposition, patterns recognition and algorithms.

They will be able to create algorithms to solve specific problems and then implement these within programming languages.

They will be able to recall what bits and bytes are as well as convert between the different sizes including Kilobyte and extend their knowledge up to Gigabyte

They will be able to recall how to convert between denary, binary and hexadecimal.

They will the able to explain how images and sound are represented using binary digits and the relationship between the number of bits used and the size of the file.

Students will be able to explain the issues relating to cyber security and suggest methods of keeping information safe. This will link with their moral, legal and ethical unit in year 8. They will seek to expand their understanding of the world around them be able to explain and in some cases argue specific positions in relation to these issues.

Students will have opportunities to plan, monitor and improve own learning and work, be independent learners, listen to others and ask questions to extend understanding.</td></tr>
<tr><td colspan="3">ASPIRE: Learning Respectfully,Insight, Evaluating, Questioning, Making Links
Challenge:
Reading: Current news articles, Academic Articles
Super Curricular:
<u>Read:</u> Brown, D (1998) Digital Fortress
<u>Do:</u> <u>http://targetedattacks.trendmicro.com/</u>
<u>Watch</u>: Hackers</td></tr>
<tr><td>YEAR 9 part 2</td><td>Algorithms and Programming</td><td>Be able to recall and apply sequence, selection and iteration within problems
Be able to describe and apply arrays and lists within algorithms and programs
Be able to describe algorithms used to search and sort data
Be able to apply algorithms to search and sort data
Be able to convert algorithms into code to search and sort data</td></tr>
<tr><td colspan="3">ASPIRE: All of learning to understand, Planning ahead, Reflecting, Risk Taking, Collaboration, Communication, Perseverance
Challenge: Levitin, A & Levitin, M (2011) Algorithmic Puzzles
Reading: Kubica, J. (2012). Computational fairy tales.
Super Curricular:
<u>Read:</u> Kubica, J. (2012). Computational fairy tales.
<u>Do:</u>
<u>Watch</u>:</td></tr>
<tr><td>YEAR 9 part 3</td><td>Data Representation</td><td>Be able to explain the role of binary
Be able to explain how numbers are represented in bits and bytes
Be able to convert between bits, nibbles, bytes, kilobytes and megabytes
Be able to convert between denary, binary and hexadecimal with accuracy and confidence
Be able to recall how letters are represented in ASCII
Be able to describe the need for Extended ASCII and Unicode
Be able to explain how bitmap images are made up, calculate the file size of an image and explain the impact of colour depth on file size.
Be able to describe how sound is represented using binary.
Be able to calculate the file size of a sound file
Be able to describe the implications of sample resolution on file size</td></tr>
</table>

ASPIRE: Insight, Making Links, Risk taking, Organising, Logical Thinking, Valuing Others **Challenge:** Mathematical Applications **Reading:** Kubica, J. (2012). *Computational fairy tales*. **Super Curricular:** **Read:** Kubica, J. (2012). *Computational fairy tales*. **Do:** **Watch**:	

Computing – Algorithms

YEAR 7	YEAR 8	YEAR 9
		MASTERING + Working above a mastering level
	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.**		

Computing – Programming & Development

YEAR 7	YEAR 8	YEAR 9
		MASTERING + Working above a mastering level
	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.**		

Computing – Data & Data Representation

YEAR 7	YEAR 8	YEAR 9
		MASTERING + Working above a mastering level
	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.**		

Computing – Information Technology

YEAR 7s	YEAR 8	YEAR 9
		MASTERING + Working above a mastering level
	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.**
MASTERING + Working above a mastering level	**MASTERING** **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.**	**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.**	**SECURING** **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.**	**DEVELOPING** **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.**		