Key Stage 3 Curriculum Overview Computing

Curriculum Intent

To ‘Live Life in All Its Fullness’ in the world that Holywell students will encounter as adults will require the embracing of technology as a discerning, responsible and competent Digital Citizen. In this light, our Computing Curriculum is designed to empower students with the foundational knowledge, skills, and understanding necessary to thrive in a rapidly evolving digital world. Our curriculum aims to inspire a passion for technology and problem-solving, equipping students with the confidence and competence to navigate, create, and innovate in the digital landscape. By fostering a deep understanding of computing principles, we strive to develop critical thinkers who are not only adept at using technology but also capable of understanding and shaping its future.

Through a blend of theory, practical application, and project-based learning, the KS3 Computing Curriculum seeks to nurture well-rounded individuals who are not only proficient in technology but also capable of using it to enhance their lives and the lives of others. We are dedicated to creating an inclusive and supportive environment that encourages every student to explore, discover, and thrive.

Term	Year 7	Year 8
Autumn term 1	**Title of unit:** Programming Essential in Scratch – Part I **Main focus:** Further develop programming skills in relation to: sequencing, variables, selection, and count-controlled iteration. **By the end of the unit, students will have learned to:** Use a programming language to solve computational problems Undertake creative projects with challenging goals Understand simple Boolean logic **By the end of the unit students will be able:** To compare how humans and computers understand instructions (understand and carry out) To recognise that computers follow the control flow of input/process/output To define a sequence as instructions performed in order, with each executed in turn To define a variable as a name that refers to data being held by the computer To design a sequence that includes variables (write an algorithm) To create expressions that use arithmetic operators (+ - / *) To define a condition as an expression that will be evaluated as either true or false	**Title of unit:** Computational Thinking & Logic **Main focus:** To introduces students to the concepts of computational thinking and logic. **By the end of the unit, students will have learned to:** Evaluate computational abstractions Model state of physical systems Model the behaviour of real-world problems Understand several key algorithms that reflect computational thinking Use at least one additional programming language (that must be textual) to solve computational problems Understand uses of Boolean logic in programming (e.g. AND, OR, and NOT) **By the end of the unit students will be able:** To know how to ask logical questions to solve problems To understand the common Boolean operators: AND, OR NOT To know different logic gates including: AND gates, OR gates, NOT gates To understand what an algorithm is To create a sequence of instructions to achieve a goal Understand how Boolean operators can be represented in written expressions and Venn diagrams Understand how logic is used in different situations To complete truth tables for logic gates and circuits with up to three inputs

	To create conditions that use comparison operators (>,<,=) To create conditions that use logic operators (and/or/not) Identify that selection uses conditions to control the flow of a sequence Define iteration as a group of instructions that are repeatedly executed Identify where count-controlled iteration can be used in a program Implement iteration in a program	Understand how loops can be used to reduce the amount of code required for a solution To refine algorithms to reduce the number of instructions required To understand the difference between lossy and lossless compression To use an algorithm to communicate data: understand how the algorithm can be improved, use a binary tree to further improve the algorithm Understand why compression is needed for video transmission and photo storage Understand how abstractions are used in everyday life To create abstractions for different purposes Understand how networks are used to make an abstraction of a maze Understand how decomposition can be used to break down problems into more manageable components To break down a large Computing problem into its parts and understand: how images are converted to binary using pixels, how text is converted to binary using ASCII Understand how nested loops can be used to improve solutions further To use an algorithm to communicate data: Use a binary tree to further improve the algorithm Understand network (graph) theory terms including: Nodes, Edges To break down a large Computing problem into its parts and understand: how data is broken up into packets, how data is sent through a network
Evidence of learning	Summative assessment: Lesson 6 Scratch program will be evaluated against a Rubric – self assessment. By the end of this unit of work, students should have learned the component knowledge above. The unit is designed to enable students to move advance from Year 6 to Year 7 Computer Science performance descriptors.	Summative assessment: End of unit Test. This unit links directly to the Year 8 performance descriptors related to algorithms, programming and Boolean logic.
Links to prior learning	Develops concept learnt in year 5 & 6 programming modules. Namely sequence, selection and iteration.	This unit will use Scratch skill developed in Year 6 & 7 programming modules. It will also develop earlier conceptual learning related to Algorithms.
Links to future learning	Year 7 Module Programming essentials in scratch II will build on this unit as will programming in year 8.	Algorithm will be further developed in Year 8 Mobile App development and Intro to Python units. This unit will be of great value at GCSE level.
Careers links	Systems engineer Software developer	Systems developer Software engineer Data Scientist
Autumn term 2	**Title of unit:** Understanding Computers **Main focus:** An introduction to the basic principles of computer architecture and binary. **By the end of the unit, students will have learned to:** Understand how numbers can be represented in binary	**Title of unit:** Developing for the Web **Main focus:** Students will explore the technologies that make up the internet and World Wide. **By the end of the unit, students will have learned to:** Understand the hardware and the software components that make up computer systems

	Understand the hardware components that make up computer systems Understand how text can be represented digitally in the form of binary digits Understand how pictures can be represented digitally in the form of binary digits **By the end of the unit students will be able to:** Distinguish between hardware and software Identify Input, Output and Storage devices Understand how processor speed affects computation time Define a Bit, Byte, KB Name three types of optical storage device Review the history and development of communication Understand how modern communication and computing devices combine multiple technologies Discuss the different ways and applications in which modern technology is used Name at least five pieces of software Suggest appropriate input and output devices for a given scenario Draw a block diagram of the main components of a computer including input, processor, output and storage Distinguish between main memory and permanent storage devices Name the three stages in the Fetch Execute Cycle Understand the meaning of Hz, kHz, MHz and GHz Understand why all computer data is represented in binary Define a MB and GB Convert integers to binary numbers Add two binary numbers State the typical capacities, strengths and weaknesses of different storage devices Describe how data is stored on an optical disc Describe how 0s and 1s are represented by pits and lands on an optical disc Discuss future uses of technology and the pace of change (Moore’s Law) Some students will be able to: Understand properties of binary numbers	Combine multiple applications to achieve challenging goals (NB. This needs to involve selecting, using and combining multiple applications) Revise digital artefacts for a given audience Meet the needs of known users Attend to trustworthiness of digital artefacts **By the end of the unit students will be able to:** Explain what HyperText Markup Language (HTML) is used for Use HTML to define the appearance of text on a web page Use HTML to display an image on a webpage Apply HTML tags to structure a web page Explain that websites are made up of multiple web pages Create hyperlinks to allow users to navigate between web pages Explain what Cascading Style Sheets (CSS) are used for Explain that web pages in a website are arranged hierarchically Describe what a search engine is Use CSS to style web pages Implement navigation to complete a functioning website Outline how search engines crawl the World Wide Web to create an index Use parameters to improve searches Explain how search engines select and rank results Explain how individual search engines have their own ranking algorithms

	Identify binary numbers as odd or even Convert binary numbers to integers Understand the effect of adding or removing a zero to a binary number	
Evidence of learning	Summative assessment: End of unit test. This unit is designed as an introduction to system architecture and binary usage in computing.	Summative assessment: End of unit multiple choice test. This unit further develops Web development knowledge and skills in line with Year 8 Performance descriptors. In particular those related to digital artefacts and meetings the needs of known users.
Links to prior learning	Year 5 Computer Systems module will give a basic understanding of I/O devices and may touch on binary.	This unit further develops ideas encountered in Year 6 Web page creation.
Links to future learning	Links to Computational thinking in Year 8 and Representational of data in GCSE curriculum.	Knowledge acquired here will provide a foundation for learning in KS4 creative computing.
Careers links	Systems architect Systems programmer	Web Developer Graphic Designer Content Creator
Spring term 1	**Title of unit:** Networks: from semaphores to the internet **Main focus:** Define networks in terms of hardware, software and protocols and discusses how they are integral to life in a modern society. **By the end of the unit, students will have learned to:** Understand the hardware components that make up computer systems (in particular, networked systems) **By the end of the unit students will be able to:** Define what a computer network is List network hardware Define ‘protocol’ and provide examples of non-networking protocols Explain how data is transmitted between computers across networks Define ‘bandwidth’, using the appropriate units for measuring the rate at which data is transmitted Define what the internet is Compare wired to wireless connections Describe key words such as ‘protocols’, ‘packets’, and ‘addressing’ Explain how data travels between computers across the internet Explain the difference between the internet, its services, and the World Wide Web Explain the term ‘connectivity’ Describe how services are provided over the internet	**Title of unit:** Understanding Computers **Main focus:** An introduction to the basic principles of computer architecture and binary. **By the end of the unit, students will have learned to:** Understand how numbers can be represented in binary Understand the hardware components that make up computer systems Understand how text can be represented digitally in the form of binary digits Understand how pictures can be represented digitally in the form of binary digits **By the end of the unit students will be able to:** Distinguish between hardware and software Identify Input, Output and Storage devices Understand how processor speed affects computation time Define a Bit, Byte, KB Name three types of optical storage device Review the history and development of communication Understand how modern communication and computing devices combine multiple technologies Discuss the different ways and applications in which modern technology is used Most students will be able to: Name at least five pieces of software Suggest appropriate input and output devices for a given scenario

	Explain how the IoT can collect and share information about one with or without one's knowledge Describe how internet-connected devices can affect one Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together	Be able to draw a block diagram of the main components of a computer including input, processor, output and storage Distinguish between main memory and permanent storage devices Name the three stages in the Fetch Execute Cycle Understand the meaning of Hz, kHz, MHz and GHz Understand why all computer data is represented in binary Define a MB and GB Convert integers to binary numbers Add two binary numbers State the typical capacities, strengths and weaknesses of different storage devices Describe how data is stored on an optical disc Describe how 0s and 1s are represented by pits and lands on an optical disc Discuss future uses of technology and the pace of change (Moore's Law) Some students will be able to: Understand properties of binary numbers Identify binary numbers as odd or even Convert binary numbers to integers Understand the effect of adding or removing a zero to a binary number
Evidence of learning	Summative assessment: End of unit multiple choice test. This unit is designed as an introduction to the structure and usage of networks including the internet.	Summative assessment: End of unit test. This unit is designed as an introduction to system architecture and binary usage in computing.
Links to prior learning	This unit assumes no prior learning.	Year 5 Computer Systems module will give a basic understanding of I/O devices and may touch on binary.
Links to future learning	At KS4 there will be a unit on computer systems that will further develop the learning from this module.	Links to Computational thinking in Year 8 and Representational of data in GCSE curriculum.
Careers links	Computer programming Network Engineering IT Support Technician	Systems Architect Systems Programmer
Spring term 2	**Title of unit:** Physical Computing – Using the Micro:bit **Main focus:** To apply previously learnt concepts such as sequence, selection, iteration to a physical computing environment. **By the end of the unit, students will have learned to:** Model the state of real-world problems Use a programming language to solve computational problems	**Title of unit:** Mobile App Development **Main focus:** To introduce mobile app development and take students from the design phase through to product development stage of the design process. **By the end of the unit, students will have learned to:** Use at least one additional programming language (that must be textual) to solve computational problems

	Undertake creative projects with challenging goals Reuse digital artefacts for a given audience **By the end of the unit students will be able to:** Understand how variables and inputs can be used on the micro:bit to create a sports counter Create an algorithm for a sport counter, and code, run and evaluate the use of the micro:bit to count activities Create a countdown timer on the micro:bit using variables Evaluate the effectiveness of the LED display on the micro:bit when used as a timer Modify a program using true and false statements and an if…else command Create an activity completion using a micro:bit counter and a micro:bit timer Compare different inputs on the micro:bit Define iteration Modify a program with count-controlled iteration Create a countdown program using count-controlled iteration Define a function Modify a program to gather data and visualise the data	Make use of appropriate data structures Design modular programs that use procedures or functions **By the end of the unit students will be able to:** Use a block-based programming language to create a sequence Recognise that events can control the flow of a program Implement and customise GUI elements to meet the needs of the user Use variables in an event-driven programming environment Pass the value of a variable into an object Use user input in an event-driven programming environment Identify and fix common coding errors in a block-based environment Use a block-based programming language to include selection Apply decomposition to break down a larger problem into more manageable steps Establish user needs when completing a creative project
Evidence of learning	Summative assessment: Peer assessment of programs created. Year 7 programming progress descriptors will be addressed by this unit.	Summative Assessment: End of unit test. Also, peer-assessment of final project using a rubric. This unit is designed to introduce mobile app development and will is related to Year 8 computer science progress descriptors.
Links to prior learning	This unit will incorporate year 5 and 6 learning of the three major concepts of programming: sequence, iteration and selection.	This module will incorporate all programming knowledge gained in Year 5, 6 and programming units.
Links to future learning	This module will support further learning in Year 7 Programming Essentials II and Year 8 Mobile App development and Introduction to Python and GCSE programming units.	Knowledge, skills and understanding acquired here will be further developed GCSE computing modules.
Careers links	Systems Developer Software Engineer	Software developer Project Manager Designer
Summer term 1	**Title of unit:** Programming Essential in Scratch – Part II **Main focus:** Learners will build on their understanding of the control structures' sequence, selection, and iteration (the big three), and develop their problem-solving skills. Learners will learn how to create their own	**Title of unit:** Media – Vector Graphics **Main focus:** This unit offers learners the opportunity to design graphics using vector graphic editing software. By the end of the unit learners will have produced an illustration, a logo, or some icons using vector graphics.

	subroutines, develop their understanding of decomposition, learn how to create and use lists, and build upon their problem-solving skills by working through a larger project at the end of the unit. **By the end of the unit, students will have learned to:** Model the state of real-world problems Use a programming language to solve computational problems Understand simple Boolean logic Undertake creative projects with challenging goals Reuse digital artefacts for a given audience **By the end of the unit students will be able to:** Compare how humans and computers understand instructions (understand and carry out) Recognise that computers follow the control flow of input/process/output Define a sequence as instructions performed in order, with each executed in turn Predict the outcome of a simple sequence Define a variable as a name that refers to data being held by the computer Trace variables within a sequence Make a sequence that includes a variable Create expressions that use arithmetic operators (+ - / *) Define a condition as an expression that will be evaluated as either true or false Create conditions that use lo comparison operators (>,<,=) Create conditions that use logic operators (AND/OR/NOT) Identify that selection uses conditions to control the flow of a sequence Identify where selection statements can be used in a program Define iteration as a group of instructions that are repeatedly executed Describe the need for iteration Implement iteration in a program (count-controlled and condition-controlled) Evaluate which type of iteration is required in a program Define a subroutine as a group of instructions that will run when called by the main program or other subroutines	**By the end of the unit, students will have learned to:** Meet the needs of known users Combine multiple applications to achieve challenging goals (NB. This needs to involve selecting, using and combining multiple applications) Attend to trustworthiness of digital artefacts **By the end of the unit students will be able to:** Explain that vector graphics are created using paths Create objects and change their properties Recognise an image can be created by combining shapes Explain that z-order describes the layer an object is in Outline the different ways of working with multiple objects Manipulate groups of objects Explain union, difference, and intersection when combining objects Explain the use of nodes in a path Change the form of an object by moving nodes Use tools and techniques to create a vector graphic for a given purpose Outline differences between vectors and bitmaps Explain why vectors can be scaled without impacting quality Describe the data used in a vector file format Review and improve a vector graphic Identify situations where using vector graphics would be appropriate

	Define decomposition as breaking a problem down into smaller, more manageable subproblems Identify how subroutines can be used for decomposition Define a list as a collection of related elements that are referred to by a single variable name Describe the need for lists Identify when lists can be used in a program Decompose a larger problem into smaller subproblems	
Evidence of learning	Self-assessment of a pair programming project. A rubric will be used. Links to Year 7 Programming Progress Descriptors.	Using a rubric student will self-assess their vector graphics project. This unit is designed to meet IT and Digital literacy progress descriptors for year 8.
Links to prior learning	This unit build on concepts learnt in Year 7 Programming Essentials I.	In year 5, students are introduced to the basics of vector drawings.
Links to future learning	Foundational knowledge of lists and subroutines will be explored further in Year 8 Mobile App Development and GCSE programming units.	The knowledge acquired in this unit could support GCSE learning in Creative Computing and computer Illustration modules in art lessons.
Careers links	Software Engineer Games developer	Illustrator Graphic Designer Motion Graphics Artist
Summer term 2	**Title of unit:** Using Computers Safely, Effectively and Responsibly **Main focus:** This is a theoretical unit covering the necessary basic knowledge to use computers safely, effectively and responsibly. Pupils begin by looking at file management and security. The unit then moves on to e-safety and online profiles to give pupils a better understanding and awareness of using social media and the Internet safely. The functionality and operation of email and search engines and how to use them effectively are covered. **By the end of the unit, students will have learned to:** Understand a range of ways to use technology respectfully Recognise inappropriate content Recognise inappropriate contact Recognise inappropriate conduct Know how to report concerns Attend to usability of digital artefacts Understand a range of ways to use technology safely **By the end of the unit students will be able to:** Understand your school's Acceptable Use Policy (AUP)	**Title of unit:** Introduction to Python Programming **Main focus:** This unit introduces learners to text-based programming with Python. The lessons form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. Emphasis is placed on tackling common misconceptions and elucidating the mechanics of program execution. **By the end of the unit, students will have learned to:** Evaluate computational abstractions Model state of physical systems Model the behaviour of real-world problems Understand several key algorithms that reflect computational thinking Use at least one additional programming language (that must be textual) to solve computational problems Make use of appropriate data structures Design modular programs that use procedures or functions Understand uses of Boolean logic in programming (e.g. AND, OR, and NOT) **By the end of the unit students will be able to:** Describe what algorithms are.

	Organise your files in File Explorer Be able to use appropriate file names Know what social media is and the risks of using it including: Strangers Cyberbullying Inappropriate content Be able to report concerns Be able to lock your computer Be able to send email including: Attachments Understand what a search engine is Understand the importance of backup Use common shortcut keys Understand the importance of protecting your online identity including the use of privacy settings Be able to make and use strong passwords including memorable passwords Understand alternative security methods to passwords Know what identity theft is and how to react to phishing and spam Understand how email works and the advantages and disadvantages of using it Send email including: Appropriate language Search email Give some examples of different search engines Add and search bookmarks and web history Understand that information on the world wide web may be inaccurate or unreliable	Describe what programs are. Describe differences between algorithms and programs. Recall how machines need translators for executing programs. Use an IDE to write and execute a Python program. Locate and correct common syntax errors. Arrange program statements in a sequence. Call functions and use the results that they return. Walk through a sequence and sketch the state and output. Walk through branches and sketch the state and output. Walk through loops and sketch the state and output. Use binary selection to control the flow of program execution. Use selection to control the flow of program execution. Use iteration to control the flow of program execution. Use variables as counters in iterative programs. Use Boolean variables as flags. Combine iteration and selection
Evidence of learning	Summative assessment: End of unit test. This unit fulfils all of the Year 7 Digital Literacy Progress descriptors.	Summative assessment: End of topic test. This unit relates to the Year 8 programming orientated computer science progress descriptors.
Links to prior learning	Year 5 and year 6 E-safety units will provide foundational knowledge.	Year 7 units Programming Essentials I & II are pre-requisite for this unit.

<table>
<tr><td>Links to future learning</td><td>Links to GCSE Cybersecurity learning.</td><td>GCSE Python programming units will build on knowledge, skill and understanding gained here.</td></tr>
<tr><td>Careers links</td><td>Security Architect Security Engineer Digital Forensics Security consultant Cybersecurity Lawyer</td><td>Software developer Software engineer Games developer Data Scientist.</td></tr>
<tr><td colspan="3"><b>Reading in the curriculum (Literacy & Vocabulary)</b>
New vocabulary is introduced to students through key terms in each lesson. Throughout the curriculum we use a range of different reading resources to add depth and knowledge to students understand.</td></tr>
<tr><td colspan="3"><b>Safeguarding including safety in the curriculum</b>
Every year group has an e-safety unit each year to explicitly explore safety and safeguarding in relation to computing/technology</td></tr>
<tr><td colspan="3"><b>Values across the curriculum</b>
Our curriculum supports the understanding of the school’s core values throughout all of our units of work.</td></tr>
<tr><td colspan="3"><b>Spirituality in the curriculum</b>
Our curriculum supports the spiritual development of students by creating an environment of curiosity, exploring interconnectedness, and fostering open-mindedness. By developing these key attributes, we hope to develop a sense of connection to something bigger than ourselves, to help students ‘Live life in all its fulness’, living our values; being the best we can be, in community.</td></tr>
<tr><td colspan="3"><b>How we track your progress</b>
Linking to the progress descriptors all students’ progress is tracked through the work they produce and contribute to in class, homework, end of unit assessments and in class assessments/quizzes.</td></tr>
<tr><td colspan="3"><b>Parents/Carers can support their child by:</b>
These are the programs we use in school for coding: Ozaria, Scratch, Flowol, Microbits. Scratch and Ozaria are available to download free.</td></tr>
</table>