| | Topic | Road Map | Key Knowledge & Skills |
|---|---|---|---|
| **Aut1** | **Python Advanced** | L1 Welcome back to the School Network<br>L2 Sequencing & Variables<br>L3 Input/Output & Mathematical Operators<br>L4 Selection IF ELSE & Comparison Operators<br>L5 Selection ELIF<br>L6 Iteration WHILE<br>L7 While Loops Continued<br>L8 Iteration FOR<br>L9 Subroutines<br>L10 Data Structures<br>L11 Pseudocode<br>L12 End of Topic Assessment | **Strand 2:**<br>**Python Programming, Sequence, Selection & Iteration**<br><br>Building on prior knowledge of Python from year 7 & 8. Identifying the programming constructs. Using variables to store data. Basic data types (str, int, float, bool). Using input functions to get user data. Displaying output using print statements. Understanding conditional statements (IF, ELSE). Understanding how loops work, Iteration (FOR, WHILE). Using Lists and Arrays. Understanding Functions & Procedures. Writing using pseudocode. |
| **Aut2** | **Data Representation** | L1 Units of Data Storage<br>L2 Binary Conversion<br>L3 Boolean Logic<br>L4 Hexadecimal Conversion<br>L5 Binary Addition<br>L6 Character Sets<br>L7 How are Images Stored?<br>L8 Resolution & Colour Depth<br>L9 Compression & Metadata<br>L10 How is Sound Stored?<br>L11 End of Topic Assessment | **Strand 3:**<br>**Data Representation, Binary & Boolean Logic**<br><br>Understanding file sizes. Converting between binary, decimal and hexadecimal numbers. Using logic gates (AND, OR, NOT). Adding binary numbers. Differences between ASCII and Unicode. Understanding bitmaps and pixels. color depth (1-bit, 8-bit, 24-bit, 32-bit) means and how it affects images. How sound is converted into binary using sampling. |
| **Spr1** | **Web Design** | L1 Introduction to HTML<br>L2 HyperText Mark-up Language<br>L3 Cascading Style Sheets<br>L4 DIV Tags & Hyperlinks<br>L5 CSS Layouts<br>L6 JavaScript<br>L7 Developing for the Web<br>L8 End of Topic Assessment | **Strand 1:**<br>**Computational Thinking, Problem Solving & Algorithms**<br><br>Understanding the structure of an HTML document. Using structural tags. Introduction to CSS and its purpose in styling web pages. Basic introduction to JavaScript. Combining HTML, CSS, and JavaScript to create interactive web pages. |
| **Spr2** | **Computer Networks** | L1 Threats to a Network<br>L2 Network Security<br>L3 The Internet & Cloud Computing<br>L4 IP Addresses, Packets & Bandwidth<br>L5 Networking Protocols<br>L6 Networking Hardware<br>L7 Network Topologies<br>L8 End of Topic Assessment | **Strand 5:**<br>**Networking, Internet & Network Security**<br><br>Understanding different network threats. importance of cybersecurity in networks. How the internet and cloud computing works. Data transmission. Protocols (HTTPS, POP, IMAP, SMTP,TCP/IP, FTP). Understanding Topologies (Star, Mesh). Identifying key networking hardware (Switch, Router, Servers). |
| **Sum1** | **Impact of Technology** | L1 Environmental Impacts<br>L2 Cultural Impacts<br>L3 Legal Impacts<br>L4 Ethical Impacts<br>L5 Privacy<br>L6 Software Licences<br>L7 End of Topic Assessment | **Strand 6:**<br>**Information Technology, Graphic Design**<br><br>Understanding how technology affects the environment (mining, manufacturing, transportation, use, waste). The digital divide, Computing laws (data protection, computer misuse, copyright). Proprietary vs open source software. |
| **Sum2** | **Systems Architecture, Memory & Storage** | L1 Architecture of the CPU<br>L2 Factors that Affect CPU Performance<br>L3 Main Memory<br>L4 End of Year Assessment<br>L5 Embedded Systems<br>L6 Magnetic, Optical Solid State<br>L7 Selecting Secondary Storage | **Strand 4:**<br>**Computer Hardware, Software & OS**<br><br>Understanding the von Neumann architecture in modern computing. Fetch-Decode-Execute cycle. Understanding the different types of primary memory. How embedded systems differ from general-purpose computers. |

# Year 9 Computing Curriculum at The Herewoode Academy

Throughout the academic year, Year 9 students at The Herewoode Academy will further develop their digital skills, computational thinking, and problem-solving abilities. Building on their prior knowledge, they will explore more advanced computing topics. These topics will provide a strong foundation for students considering further study in computing and prepare them for the digital challenges of the future.

**1. Python Advanced (Strand 2: Python Programming, Sequence, Selection & Iteration)**
Students will build upon their previous Python knowledge, focusing on advanced programming concepts such as subroutines (functions and Procedures, data structures (lists and arrays), and error handling. They will develop more complex programs and apply computational thinking to solve real-world problems.

**2. Data Representation (Strand 3: Data Representation, Binary & Boolean Logic)**
This unit explores how data is represented and processed by computers. Students will learn about binary and hexadecimal number systems, character encoding (ASCII and Unicode), images, sound representation, and data compression techniques. These concepts are essential for understanding how computers store and manipulate information.

**3. Web Design (Strand 1: Computational Thinking, Problem Solving & Algorithms)**
Students will gain hands-on experience in web development by learning HTML, CSS, and introductory JavaScript. They will design and create their own functional websites, focusing on structure, style, and interactivity. This unit encourages creativity while reinforcing key ICT skills in digital design.

**4. Computer Networks (Strand 5: Networking, Internet & Network Security)**
Students will explore the fundamentals of computer networks, including different network types (LAN, WAN), protocols, and data transmission methods. They will also learn about key networking hardware such as routers and switches, along with cybersecurity measures to protect data.

**5. Impact of Technology (Strand 6: Information Technology & Design)**
This unit examines the ethical, legal, and social implications of technology in modern society. Students will discuss topics such as privacy, surveillance, artificial intelligence, digital divide, and the environmental impact of computing. They will critically assess the benefits and challenges of emerging technologies.

**6. Systems Architecture, Memory & Storage (Strand 4: Computer Hardware, Software & AI)**
Students will gain a deeper understanding of computer architecture, including the function of the CPU, fetch-execute cycle, and the role of RAM and ROM. They will also explore different storage devices (HDD, SSD, cloud storage) and compare their advantages and disadvantages in real-world applications.

**Overall Learning Outcomes:**
By the end of Year 9, students will have:
- ✅ Advanced programming skills in Python, including functions, data structures, and error handling.
- ✅ A strong understanding of how computers represent and manipulate data.
- ✅ Practical experience in web design using HTML, CSS, and JavaScript.
- ✅ Knowledge of networking concepts, protocols, and cybersecurity principles.
- ✅ Awareness of the ethical and societal impacts of technology.
- ✅ A deeper understanding of computer systems, including architecture, memory, and storage.

This curriculum is designed to align with the **OCR GCSE Computer Science specification**, ensuring students develop the fundamental knowledge and skills required for further study. It provides a comprehensive introduction to key computing concepts while preparing students for the ever-evolving world of technology.

# Year 9 Computing Curriculum at The Herewoode Academy

## *Assessment*

At the end of each topic in Computing, students will be assessed through a comprehensive evaluation process in the form a **summative assessment**, designed to test their understanding and practical application of the unit's content. This includes an **end-of-topic quiz** consisting of 40 questions that assess their prior learning and retention of key concepts. Additionally, students will complete a **practical challenge**, requiring them to apply their knowledge in a hands-on scenario, reinforcing problem-solving skills and real-world application. To further stretch and challenge their thinking, learners will tackle **three GCSE-style questions**, encouraging them to engage with higher-level questioning and develop exam-style responses. Finally, before leaving the classroom, students will participate in **diagnostic questioning**, allowing teachers to gauge their immediate understanding, address misconceptions, and provide instant feedback to support further learning.

At the end of the academic year, students will complete a **cumulative assessment** designed to evaluate both their theoretical understanding and practical skills in Computing. This assessment will revisit key topics from previous assessments, ensuring that students can recall and apply their knowledge effectively over time. By incorporating a mix of theory-based questions and practical challenges, the test will measure their ability to think critically, problem-solve, and demonstrate proficiency in essential computing concepts. The score from this assessment will serve as an indicator of each student's overall achievement in Computing throughout the academic year, providing valuable insight into their progress and areas for further development.

students will complete their **homework assessments** every three lessons through a **Microsoft Forms quiz**. These quizzes are designed to test their understanding of the previous lesson's content, reinforcing key concepts and ensuring knowledge retention over time. By regularly engaging with these assessments, students develop **independent study habits, self-discipline, and recall skills**, which are essential for mastering computing topics. The quizzes provide instant feedback, allowing students to identify areas for improvement and revisit challenging concepts. This structured approach to homework helps to consolidate learning, track progress, and prepare students for future assessments in a manageable and engaging way.

## *Independent Learning*

In Computing, students will develop their **independent learning skills** by working through a **topic-based booklet** for each unit. These booklets contain a variety of **problem-solving tasks and independent challenges**, encouraging students to take ownership of their learning and apply their knowledge in different contexts. Through this approach, students will enhance key skills such as **critical thinking, resilience, and self-motivation**, as they learn to tackle challenges without immediate teacher guidance. Additionally, independent learning will help improve **time management, research skills, and adaptability**, which are essential not only in Computing but also in real-world problem-solving and future studies. By developing these skills, students will become more confident and capable learners, better prepared for higher-level education and careers in technology.

## *Oracy*

In Computing, students will develop their **oracy skills** and understanding of **key vocabulary** through a range of activities designed to enhance their communication and technical language proficiency. Throughout each topic, students will engage in **class discussions, presentations, and collaborative problem-solving tasks**, allowing them to articulate their ideas clearly and confidently. Teachers will introduce and reinforce subject-specific terminology, ensuring that students can accurately use technical vocabulary when explaining concepts, debugging code, or discussing digital literacy topics. Additionally, students will apply key terms in written tasks, verbal explanations, and peer discussions, helping to strengthen their ability to communicate effectively in both academic and real-world computing contexts. By developing these skills, students will improve their ability to express complex ideas, justify their reasoning, and engage in meaningful discussions, preparing them for future learning and careers in technology.

| | Autumn 1<br>**8 weeks** | Autumn 2<br>**8 weeks** | Spring 1<br>**6 weeks** | Spring 2<br>**5 weeks** | Summer 1<br>**6 weeks** | Summer 2<br>**6 weeks** |
|---|---|---|---|---|---|---|
| Year 7 | **Introduction to The Herewoode Academy School Network**<br>- Office365, Teams<br>- Online Safety | **Computational Thinking**<br>- Flowcharts<br>- Algorithms<br>- Scratch<br>- Problem Solving | **Spreadsheets**<br>- Microsoft Excel<br>- Formulas<br>- Databases<br>- Charts & Graphs<br>- Selection IF, ELSE | **Introduction to Coding in Python**<br>- code.org<br>- Block-Based Programming<br>- Python | **Hardware & Software**<br>- Operating Systems<br>- Application & Utility<br>- Inside the Computer<br>- CPU, RAM & ROM | **Graphic Design**<br>- Raster Graphics<br>- Pixel Art (Binary)<br>- Storing Images<br>- Storyboards<br>- Creation |
| Year 8 | **Using Computers**<br>- Search Engines<br>- Copyright<br>- Cyber Abuse<br>- Online Dangers | **Python PRIMM**<br>- Predict, Run, Investigate, Modify, Make.<br>- Programming | **Digital Images**<br>- Bitmap Images<br>- Vector Graphics<br>- Image Manipulation<br>- Metadata | **Emerging Technologies**<br>- Artificial Intelligence<br>- Cryptography<br>- Encryption | **Cyber Security**<br>- Social Engineering<br>- Malicious Software<br>- Hacking<br>- Preventing Threats | **Flowol**<br>- Flowcharts<br>- Algorithms<br>- Problem Solving<br>- Sub-Routines |
| Year 9 | **Python Advanced**<br>- Input/Output<br>- Programming Constructs<br>- Lists/Arrays<br>- Subroutines | **Data Representation**<br>- Binary Conversion<br>- Character Sets<br>- Hexadecimal<br>- Storing Data | **Web Design**<br>- HTML<br>- CSS<br>- JavaScript<br>- Flowcharts<br>- Web Development | **Computer Networks**<br>- Network Dangers<br>- Network Security<br>- IP's & Protocols<br>- Network Hardware<br>- Types of Network | **Impact of Technology**<br>- Legal, Ethical, Cultural & Environmental Impacts of Technology | **Systems Architecture, Memory & Storage**<br>- CPU, RAM, ROM<br>- Secondary Storage |

**Strand 1:** Computational Thinking, Problem Solving & Algorithms

**Strand 2:** Python programming, sequence, selection & Iteration

**Strand 3:** Data Representation Binary & Boolean Logic

**Strand 4:** Computer Hardware, Software & AI

**Strand 5:** Networking, Internet & Network Security

**Strand 6:** Information Technology Graphic Design

# KS3 Computing Roadmap


**Strand 1:**
Computational Thinking, Problem Solving & Algorithms

**Strand 2:**
Python Programming, Sequence, Selection & Iteration

**Strand 3:**
Data Representation Binary & Boolean Logic

**Strand 4:**
Computer Hardware, Software & AI

**Strand 5:**
Networking, Internet & Network Security

**Strand 6:**
Information Technology Graphic Design

The Herewoode Academy

**Computing**

Computer Science | Information Technology | Creative iMedia