Computing KS3 Assessment - Gold
Year 8

Year 8 has one lesson of Computing each week. In Year 8 the aim is to enable the girls to develop further the basic skills and knowledge in Computing and digital technologies. They will be taught how to become digital citizens and to develop a range of skills. There is also a great emphasis on teaching them how to stay safe online and to protect themselves. The Computing curriculum has been designed to cover the three main areas of Digital Literacy, Computer Science and Information Technology.

Topics include: Data representation, spreadsheets, html and python.

Overview	Knowledge: What will they learn?	Skills: Understanding - What will they be able to do?	Literacy - Key Terminology	Assessment
Progress check - Gold	• Baseline assessment total /20 • Key vocabulary • Computational Thinking • Problem Solving • Binary 80 - 100% in baseline assessment = GOLD 40 - 79% in baseline assessment = SILVER 0 - 39% in baseline assessment = BRONZE			
Aut1 - Data Representation 1	Advanced knowledge of binary, hexadecimal, and other number systems. In-depth binary operations and conversions. Comprehensive image and	Perform complex binary arithmetic including subtraction and multiplication. Convert between multiple number systems (binary, decimal,	Binary, Hexadecimal, Decimal, Octal, Conversion, Image Compression, Sound Compression	Comprehensive quiz on multiple number system conversions. Complex binary arithmetic exercises. End of unit assessment./32

	sound representation techniques.	hexadecimal, octal). Analyze and explain detailed concepts of image and sound representation, including compression techniques.		
Progress check - Gold	What measurable criteria will you use to determine who is Above Expected, Expected, Below Expected, Cause for Concern? Using the average of assessed work this term. Above Expected: 90%+ Expected: 65%-89% Below expected: 50% - 64% Cause for concern < 50%			
Aut2 - Data Representation 2	Advanced knowledge of logic gates, truth tables, and Boolean algebra. Designing complex logic circuits.	Identify and use advanced logic gates. Create and interpret complex truth tables. Design complex logic circuits using multiple gates. Simplify and optimize Boolean expressions using Boolean algebra.	Logic Gates, Truth Tables, Boolean Expressions, Boolean Algebra, AND, OR, NOT, NAND, NOR, XOR, XNOR, Logic Circuits	Homework assignments on advanced truth tables and logic gates. Practical tasks designing complex logic circuits. End of unit assessment./60
Progress check - Gold	What measurable criteria will you use to determine who is Above Expected, Expected, Below Expected, Cause for Concern? Using the average of assessed work this term. Above Expected: 90%+ Expected: 65%-89% Below expected: 50% - 64% Cause for concern < 50%			

Spr1 - Spreadsheets	Advanced use of Excel. Complex data analysis, advanced formulas, and functions. Creating detailed and interactive charts.	Use Excel to perform complex data analysis. Implement advanced formulas and functions. Create detailed, interactive charts and dashboards. Use advanced data validation and conditional formatting.	Excel, Data Analysis, Advanced Formulas, Functions, Interactive Charts, Dashboards, Conditional Formatting	Practical exercises on advanced data analysis and interactive charts. Quiz on advanced Excel functions and formulas. End of unit project creating an interactive data dashboard.
Spr2 - Creating Websites HTML 1	Advanced HTML and CSS techniques. Comprehensive styling and responsive design. Creating dynamic content.	Develop advanced web pages using comprehensive HTML and CSS techniques. Implement responsive design principles. Create dynamic content using HTML5 and CSS3.	HTML, CSS, Responsive Design, HTML5, CSS3, Dynamic Content	Practical tasks creating advanced web pages. Quiz on advanced HTML5 and CSS3. End of unit project creating a responsive website with dynamic content./50
Progress check - Gold	What measurable criteria will you use to determine who is Above Expected, Expected, Below Expected, Cause for Concern? Using the average of assessed work this term. Above Expected: 90%+ Expected: 65%-89% Below expected: 50% - 64% Cause for concern < 50%			
Sum1 - HTML 2 / CSS	Expert HTML and CSS techniques. Advanced responsive design and interactive elements. Creating fully interactive and dynamic websites.	Develop expert-level web pages using advanced responsive design principles. Implement interactive elements using advanced CSS and JavaScript. Create fully interactive and dynamic	HTML, CSS, Responsive Design, Interactive Elements, JavaScript, Dynamic Websites	Practical tasks on advanced responsive design and interactivity. Quiz on expert HTML, CSS, and JavaScript. End of unit project creating a fully interactive and dynamic website.

		websites.		
Sum2 - Python	Advanced programming with Python. Complex problem-solving and algorithm design. Advanced use of data structures and libraries.	Write advanced Python programs. Solve complex problems using advanced algorithms. Use advanced data structures and libraries. Debug and optimize programs for efficiency.	Python, Data Structures, Algorithms, Libraries, Debugging, Optimization	Practical coding exercises. Quiz on advanced Python concepts. End of unit project creating a complex application using advanced algorithms and data structures.

Literacy - Key Terminology Overview		
Overview	Literacy - Key Terminology	Definitions and Examples
Aut1 - Data Representation 1	Binary, Hexadecimal, Decimal, Conversion, Image Representation, Sound Representation	Binary: A system of numbers with only 0 and 1 (e.g., the binary for 5 is 101). Hexadecimal: A base-16 number system (e.g., the hexadecimal for 10 is A).
		Decimal: The standard number system (e.g., 10 in decimal is 1010 in binary). Conversion: Changing numbers from one base to another (e.g., converting 1010 to decimal gives 10).
		Image Representation: How images are stored as binary data (e.g., pixels encoded as binary). Sound Representation: Storing sound as binary data (e.g., converting audio to WAV format).
Aut2 - Data Representation 2	Logic Gates, Truth Tables, Boolean Expressions, Boolean Logic, AND, OR, NOT	Logic Gates: Basic components of digital circuits that perform logical operations (e.g., AND gate outputs 1 only if both inputs are 1).
		Truth Tables: Tables that show all possible outcomes of logical operations (e.g., AND gate: 0 AND 0 = 0, 1 AND 0 = 0, 1 AND 1 = 1).
		Boolean Expressions: Expressions using AND, OR, NOT to combine logic (e.g., A AND B). Boolean Logic: Mathematical logic based on true/false values.
Spr1 - Spreadsheets	Excel, Data Summarization, Formulas, Charts, Data Validation	Excel: A spreadsheet program used for organizing data (e.g., using cells for numbers and text). Data Summarization: Using functions to summarize data (e.g., SUM, AVERAGE).

		Formulas: Mathematical expressions used in spreadsheets (e.g., "=A1+B1" adds values in cells A1 and B1). Charts: Graphical representations of data (e.g., pie charts).
		Data Validation: Ensuring the accuracy of data (e.g., restricting cell values to numbers only).
Spr2 - Creating Websites HTML 1	HTML, Tags, Links, Images, Design, CSS	HTML: Hypertext Markup Language, used to create web pages (e.g., <h1>Hello World</h1>). Tags: Elements in HTML used to define structure (e.g., <p> for paragraphs).
		Links: Used to connect different web pages (e.g., Click here). Images: Embedding images in a webpage (e.g., <img src="image.jpg">).
		Design: The layout and appearance of a website (e.g., deciding the position of text and images). CSS: Cascading Style Sheets used for styling HTML elements (e.g., color: red;).
Sum1 - HTML 2 / CSS	HTML, CSS, Design, Layout, Forms	HTML: The structure of a webpage (e.g., using to create a section). CSS: A language used to style HTML elements (e.g., changing text color with color: blue;).
		Design: Planning the visual elements of a webpage (e.g., creating a user-friendly layout). Layout: The arrangement of elements on a webpage (e.g., grid or flex layout).
		Forms: HTML elements for gathering user input (e.g., <form>, <input> for text fields).

Sum2 - Python	Python, Output, Input, Variables, Selection, Iteration, Functions	Python: A high-level programming language (e.g., writing print("Hello World") to output text). Output: Displaying data from the program (e.g., printing to the screen).
		Input: Accepting data from the user (e.g., using input() to get the user's name). Variables: Storing values for use later (e.g., x = 5).
		Selection: Making decisions in a program (e.g., using if statements). Iteration: Repeating actions (e.g., using loops like for and while).
		Functions: Reusable blocks of code (e.g., def greet(name): return "Hello, " + name).