

Computing KS3 Assessment - Bronze

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 - Bronze	• <u>Baseline assessment total /20</u> • 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	Basic understanding of binary and hexadecimal systems. Understanding binary operations and basic conversions. Introduction to image and sound	Perform simple binary addition. Convert between binary, decimal, and hexadecimal. Understand basic concepts of how images and sounds are	Binary, Hexadecimal, Decimal, Conversion, Image Representation, Sound Representation	Baseline quiz on binary and hexadecimal conversions. Simple binary addition exercises. End of unit assessment. /32

	representation.	represented in binary.		
Progress check - Bronze	What measurable criteria will you use to determine who is Above Expected, Expected, Below Expected, Cause for Concern? Using the average of assessed work for this term. However, the baseline test will not be taken into account at this stage, because it is a measure at the start of the course and will differ according to prior knowledge and primary school Computing lessons. • Above Expected: 70%+ • Expected: 40%-69% • Below expected: 25% - 39% • Cause for concern < 25%			
Aut2 - Data Representation 2	Introduction to logic gates and truth tables. Basics of combining logic gates and forming simple Boolean expressions.	Identify and use basic logic gates. Create and interpret simple truth tables. Combine basic logic gates to form simple Boolean expressions. Apply Boolean logic rules.	Logic Gates, Truth Tables, Boolean Expressions, Boolean Logic, AND, OR, NOT	Homework assignments on truth tables and logic gates. Practical tasks combining logic gates. End of unit assessment./60
Progress check - Bronze	What measurable criteria will you use to determine who is Above Expected, Expected, Below Expected, Cause for Concern? Using the average of assessed work for this term. However, the baseline test will not be taken into account at this stage, because it is a measure at the start of the course and will differ according to prior knowledge and primary school Computing lessons. • Above Expected: 70%+ • Expected: 40%-69% • Below expected: 25% - 39% • Cause for concern < 25%			

Spr1 - Spreadsheets	Basics of using Excel. Introduction to data summarization, basic formulas, and creating simple charts.	Use Excel to enter and format data. Summarize data using basic formulas. Create and format simple charts. Understand and use basic data validation techniques.	Excel, Data Summarization, Formulas, Charts, Data Validation	Practical exercises on data entry and summarization. Quiz on basic Excel functions and formulas. End of unit project creating a basic data summary and chart.
Spr2 - Creating Websites HTML 1	Introduction to HTML. Understanding basic tags, links, images, and design principles. Basics of CSS.	Create simple web pages using basic HTML tags. Insert links and images. Apply basic CSS for styling.	HTML, Tags, Links, Images, Design, CSS	Practical tasks creating simple web pages. Quiz on HTML tags and CSS basics. End of unit project creating a simple website./50
Progress check - Bronze	What measurable criteria will you use to determine who is Above Expected, Expected, Below Expected, Cause for Concern? Using the average of assessed work for this term. However, the baseline test will not be taken into account at this stage, because it is a measure at the start of the course and will differ according to prior knowledge and primary school Computing lessons. • Above Expected: 70%+ • Expected: 40%-69% • Below expected: 25% - 39% • Cause for concern < 25%			
Sum1 - HTML 2 / CSS	Advanced HTML and CSS techniques and tags. Design and layout principles. Creating forms.	Develop more complex web pages using advanced HTML and CSS. Implement design and layout principles. Create and style forms.	HTML, CSS, Design, Layout, Forms	Practical tasks on advanced HTML and CSS. Quiz on design and layout principles. End of unit project creating a fully styled web page with a form.

Sum2 - Python	Basics of programming with Python. Introduction to output, input, variables, simple calculations, and control structures.	Write simple Python programs. Use output and input functions. Perform basic calculations with variables. Understand and use selection and iteration. Create a simple calculator using functions.	Python, Output, Input, Variables, Selection, Iteration, Functions	Practical coding exercises. Quiz on Python basics. End of unit project creating a simple calculator program.
----------------------	---	--	---	--

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: $0 \text{ AND } 0 = 0$, $1 \text{ AND } 0 = 0$, $1 \text{ AND } 1 = 1$).
		Boolean Expressions: Expressions using AND, OR, NOT to combine logic (e.g., $A \text{ AND } B$). 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., <code>"=A1+B1"</code> 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., <code><h1>Hello World</h1></code>). Tags: Elements in HTML used to define structure (e.g., <code><p></code> for paragraphs).
		Links: Used to connect different web pages (e.g., <code>Click here</code>). Images: Embedding images in a webpage (e.g., <code></code>).
		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., <code>color: red;</code>).

Sum1 - HTML 2 / CSS	HTML, CSS, Design, Layout, Forms	HTML: The structure of a webpage (e.g., using <div> 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).