| Computer Science | Working towards expected outcomes<br>Your child is not yet making the expected progress within this course. | Working at expected outcomes<br>Your child is achieving the expected progress for this point within the course. | Working beyond expected outcomes<br>Your child is working beyond the expected progress for this point within the course. |
|---|---|---|---|
| **Year 9** | Students working **towards** expected outcomes in Year 9 can: | Students working **at** expected in Year 9 can: | Students working **beyond** expected in Year 9 can: |
| **Autumn 1**<br>Python | Can write simple lines of Python code with guidance but struggles with syntax accuracy.<br><br>Understands what variables and basic input/output do, but needs support to use them correctly.<br><br>Can follow simple instructions but struggles to break a problem into smaller steps.<br><br>Limited confidence using text-based programming after previously using block-based tools. | Writes correct Python programs using variables, input/output, and sequencing.<br><br>Can break a problem down into steps and translate this into working code.<br><br>Can use simple selection (if statements) with support.<br><br>Debugs simple errors using line-by-line checking or teacher prompts. | Writes well-structured Python programs using sequencing, selection, and occasionally iteration.<br><br>Breaks down more complex problems independently using decomposition.<br><br>Confidently debugs programs, explaining why an error occurs and how to fix it.<br><br>Shows growing independence in exploring new code constructs and experimenting beyond lesson examples. |
| **Autumn 2**<br>Bebras and Web Design | Attempts Bebras problem-solving questions but finds it difficult to select appropriate strategies such as pattern recognition, decomposition, or logical reasoning.<br><br>Needs guidance to write or understand basic HTML tags and often struggles with correct syntax.<br><br>Can add simple content (text or images) to a webpage when guided but cannot yet structure a page independently.<br><br>Understands that the web uses webpages and links, but has limited understanding of how websites are organised or accessed. | Uses computational thinking (decomposition, sequencing, logical reasoning) to solve Bebras challenges with some independence.<br><br>Writes basic HTML pages including headings, paragraphs, images, lists, and links with correct syntax.<br><br>Structures webpages clearly using appropriate HTML tags and can create simple multi-page websites.<br><br>Understands how the web works at a basic level, including webpages, links, browsers, and the purpose of HTML. | Approaches Bebras problems using advanced strategies such as abstraction and optimisation, demonstrating resilience and creativity.<br><br>Writes well-structured, semantically meaningful HTML code and begins to show awareness of good coding practices (indentation, comments, clean structure).<br><br>Creates multi-page websites with well-designed navigation, effective layout choices, and consistent visual styling.<br><br>Demonstrates deeper understanding of how the web works, such as client-server interaction or how browsers interpret HTML. |

| | | | |
|---|---|---|---|
| | Needs significant support to create multi-page websites or connect pages using links. | Can design a simple website with a logical hierarchy, consistent layout, and clear navigation. | Experiments beyond the core curriculum (e.g., adding CSS for styling, embedding media, or expanding website complexity). |
| **Spring 1**<br>HTML and JavaScript | Can add basic content to an HTML page but struggles to apply CSS rules without step-by-step guidance.<br><br>Finds it difficult to understand how CSS selectors, properties, and values work together to style a webpage.<br><br>Attempts to use layout features but often applies them incorrectly or inconsistently.<br><br>Needs support to understand the purpose of a navigation bar or to link pages together effectively.<br><br>Is beginning to recognise what JavaScript does but struggles to write or apply basic scripts independently.<br><br>Can follow instructions to add simple interactivity but does not yet understand how JavaScript affects webpage behaviour.<br><br>Needs guidance to combine HTML, CSS, and JavaScript in a structured way to build a multi-page site. | Can style webpages using CSS, applying common selectors and properties to control colour, fonts, spacing, and layout.<br><br>Uses HTML and CSS together to build pages with clear structure and consistent design.<br><br>Can create a working navigation bar and link multiple pages together to form a simple website.<br><br>Understands the basics of JavaScript and can create simple interactive features.<br><br>Uses JavaScript to modify webpage content or behaviour in simple but meaningful ways.<br><br>Can design and build a small multi-page website with functional navigation and consistent styling.<br><br>Demonstrates understanding of how HTML, CSS, and JavaScript work together to create interactive web experiences. | Applies CSS confidently to create well-structured, visually appealing layouts.<br><br>Designs websites with strong visual coherence, using spacing, alignment, and typography deliberately to improve user experience.<br><br>Builds more sophisticated navigation systems, including hover effects or responsive-style behaviour.<br><br>Writes JavaScript that adds meaningful interactivity, such as dynamic content updates, user input handling, or simple animations.<br><br>Combines HTML, CSS, and JavaScript in creative ways to produce polished, interactive websites that go beyond basic requirements.<br><br>Demonstrates strong independence in debugging and improving code, explaining how changes affect structure, style, or behaviour.<br><br>Experiments with more advanced techniques to enhance the quality and professionalism of their website. |
| **Spring 2**<br>Building a Computer and Spreadsheets | Can identify some internal and external computer components but struggles to explain their purpose or how they work together. | Can describe the function of key computer components (e.g., CPU, RAM, GPU, storage, peripherals) and explain how they contribute to system performance. | Gives detailed explanations of computer components and how they interact, including advanced considerations such as cooling, compatibility, or bottlenecks.<br><br>Designs a highly appropriate, well-justified “dream PC” that clearly matches the scenario and includes |

| | | | |
|---|---|---|---|
| | Has difficulty understanding how hardware choices (e.g., CPU, RAM, storage) affect the performance or role of a computer.<br><br>Needs support to select appropriate components when designing a "dream PC" and may choose parts without justification.<br><br>Finds it challenging to present ideas clearly or explain how the proposed system meets the scenario's requirements.<br><br>Can enter data into a spreadsheet but struggles to organise it or use basic formulas without guidance.<br><br>Needs help applying simple modelling tools such as sorting, filtering, or basic calculations.<br><br>Has limited understanding of how spreadsheets support decision-making or analyse data. | Selects suitable components for a "dream PC" scenario and provides logical explanations for their choices.<br><br>Presents system designs clearly with consideration for performance, purpose, and user needs.<br><br>Organises data effectively in a spreadsheet using formatting, basic formulas (SUM, AVERAGE), and simple modelling tools.<br><br>Can analyse data using sorting, filtering, or charts to support conclusions.<br><br>Understands how hardware and software interact and how spreadsheets help solve real-world problems. | thoughtful performance comparisons or cost considerations.<br><br>Creates a polished, professional presentation explaining design choices and system performance in depth.<br><br>Uses advanced spreadsheet features (e.g., multi-step formulas, conditional formatting, multiple charts) to model and analyse data effectively.<br><br>Draws clear, well-reasoned conclusions based on spreadsheet evidence and can explain how the model could be improved.<br><br>Demonstrates strong understanding of how hardware, software, and data systems work together to support real-world decision-making. |
| **Summer 1**<br>Cyber Security | Can identify some examples of online risks (scams, phishing, unsafe communication) but struggles to explain how to respond safely.<br><br>Understands that digital security is important but finds it difficult to apply basic protective measures such as strong passwords or verifying trustworthy sources.<br><br>Has a limited understanding of what artificial intelligence is and may confuse AI with general computer automation. | Recognises a wide range of online risks and can explain how to stay safe by applying appropriate digital security strategies (e.g., using secure passwords, checking URLs, questioning unexpected messages).<br><br>Understands how scams and phishing attacks operate and can describe how users can protect themselves.<br><br>Can explain how AI systems are trained using data and how this affects the decisions they make. | Provides clear, detailed explanations of online risks and can advise others on safe digital behaviour, including advanced security considerations (e.g., two-factor authentication, data privacy settings).<br><br>Analyses real scam examples and explains how social engineering techniques manipulate users.<br><br>Gives confident, accurate explanations of how AI models learn from data, including limitations such as bias, training quality, and transparency. |

| | | | |
|---|---|---|---|
| | Needs support to understand how AI systems learn from data or make decisions.<br><br>Can identify some examples of AI in everyday life but struggles to explain their purpose or impact.<br><br>Finds it challenging to express opinions about ethical or social issues related to AI without prompt or support. | Identifies common AI applications in real life (recommendation systems, chatbots, facial recognition) and understands their benefits.<br><br>Can discuss ethical, legal, and social implications of AI, including issues such as bias, misinformation, privacy, and job impact.<br><br>Shows the ability to think critically about how AI affects individuals and society. | Evaluates a wide range of AI applications, comparing benefits and risks with thoughtful reasoning.<br><br>Engages in mature discussion of the ethical, legal, and social issues surrounding AI, showing awareness of complex topics such as algorithmic fairness, surveillance, and digital rights.<br><br>Demonstrates strong critical thinking, offering well-supported arguments about the future impact of AI on society and individuals. |
| **<u>Summer 2</u>**<br>Spreadsheets and Databases | Can create basic word-processed documents but struggles to format them professionally or organise information clearly.<br><br>Needs support to design a database and may find it difficult to choose appropriate fields, data types, or layouts.<br><br>Can enter data into a database but struggles with sorting, filtering, or running simple queries.<br><br>Uses a spreadsheet but finds it difficult to apply formulas or model costs accurately without step-by-step guidance.<br><br>Has limited understanding of how the different digital tools (word processing, databases, spreadsheets, programming) connect within the project.<br><br>Can write simple Python code (e.g., print statements) but struggles to create a working | Creates well-structured, professional word-processed documents using appropriate formatting, images, and layout features.<br><br>Designs a suitable database with the correct fields, data types, and structure for storing student information.<br><br>Uses basic queries, sorting, and filtering to retrieve useful information from the database.<br><br>Builds a spreadsheet model that calculates trip costs using formulas such as SUM, multiplication, and simple functions.<br><br>Analyses spreadsheet results and can explain how changes in cost affect the final total.<br><br>Writes a functional Python FAQ program using variables and selection, providing relevant answers to user questions.<br><br>Understands how each tool contributes to the wider project and can combine them effectively to solve the problem. | Produces highly polished documents with advanced word-processing features such as tables, consistent styles, and professional layout choices.<br><br>Designs a well-normalised, efficient database and uses advanced queries or validation to improve accuracy and reliability.<br><br>Uses advanced spreadsheet features (e.g., multi-step formulas, conditional formatting, graphs, scenario modelling) to explore trip costs in depth.<br><br>Interprets spreadsheet data thoughtfully, offering well-reasoned recommendations based on cost modelling.<br><br>Creates a more sophisticated Python FAQ program using loops, menus, functions, or user-friendly features to enhance usability.<br><br>Shows strong ability to integrate multiple computing tools to complete a complex, realistic task, demonstrating independence and excellent organisation. |

| | | | |
|---|---|---|---|
| | FAQ program using variables, selection, or loops.<br><br>Needs significant support to manage a multi-stage project and keep work organised across different applications. | | Reflects on the quality and effectiveness of the final project, suggesting meaningful improvements. |