

Create hypothesise formulate design imagine compose develop improve	<ul style="list-style-type: none"> <li>• Create pieces of work to meet a variety of objectives using school software and online services.</li> <li>• Design and create a presentation on leading a safe digital life that has a suitable house style, purpose and audience.</li> <li>• Design and plan an effective computer game for a chosen audience.</li> </ul>
Evaluate recommend persuade debate justify assess conclude determine	<ul style="list-style-type: none"> <li>• Evaluate best practice for communicating and working using school services and software.</li> <li>• Evaluate the success of their computer game and identify further improvements.</li> </ul>
Analyse infer research investigate question appraise examine prioritise organise	<ul style="list-style-type: none"> <li>• Know the advantages and disadvantages of certain software, communication tools and features of school services for any given purpose.</li> <li>• Know the school network and cloud services to choose/ use the appropriate methods for saving, storing and sharing work.</li> <li>• Analyse programming commands in a computer game to 'debug' and solve problems.</li> <li>• Analyse and present some of the moral or legal decisions necessary in order to lead a safe digital life.</li> <li>• Choose appropriate software or websites for a task based on the given purpose.</li> </ul>
Apply demonstrate manipulate calculate practise identify use	<ul style="list-style-type: none"> <li>• Use key features of the operating system and school services effectively.</li> <li>• Demonstrate the functions a computer, a network and the internet by finding and mapping their use within the school.</li> <li>• Apply key programming commands to create and operate a computer game with reasonable success.</li> <li>• Demonstrate the leading of a safe digital life by explaining key findings to the school community through a presentation.</li> <li>• Independently and effectively work on the school computers, operating system, network and online services.</li> </ul>
Understand explain interpret give examples estimate illustrate	<ul style="list-style-type: none"> <li>• Explain the purpose and usage of key features of the operating system and school services that they use.</li> <li>• Explain parts of a computer, a network and key functions of the internet.</li> <li>• Know the function and purpose of key programming commands.</li> <li>• Give examples of safe practice online.</li> <li>• Understand key software well enough to be able to format work more effectively.</li> </ul>
Remember list recognise define recall label	<ul style="list-style-type: none"> <li>• Recognise key features of the operating system and school services that they use.</li> <li>• Recognise and label parts of a computer, a network and key functions of the internet.</li> <li>• Recognise the use of key programming terminology.</li> <li>• Recall key do's and don'ts of staying safe online.</li> <li>• Be able to open and use common software applications and websites so that they can be used across all subjects.</li> </ul>

<p>           Create            hypothesise            formulate            design            imagine            compose            develop            improve         </p>	<ul style="list-style-type: none"> <li>• Create a well-connected mind map showcasing a wide range of issues around 'my digital life'.</li> <li>• Create a professional Sway presentation about future technologies and the potential impact on their lives.</li> <li>• Plan, design and create a movie poster using specialist graphic editing software.</li> <li>• Create a range of documents and aids to present an idea for new mobile phone app as a group.</li> <li>• Improve documents and media continually to make work high quality and professional.</li> </ul>
<p>           Evaluate            recommend            persuade            debate            justify            assess            conclude            determine         </p>	<ul style="list-style-type: none"> <li>• Draw conclusions on actions to be taken to help improve 'my digital life'.</li> <li>• Evaluate the impact of future digital technologies and debate the advantages and disadvantages through Sway presentation.</li> <li>• Evaluate the binary data representation tasks and whether computer science would be a good options choice going forward.</li> <li>• Evaluate the movie poster tasks and whether ICT/ media would be a good options choice going forward.</li> <li>• Evaluate the group presentation of the phone app project and consider what could be learnt for future presentations or business-enterprise projects.</li> </ul>
<p>           Analyse            infer            research            investigate            question            appraise            examine            prioritise            organise         </p>	<ul style="list-style-type: none"> <li>• Add moral questions and problems to 'my digital life' mind map.</li> <li>• Analyse the research on future technologies and examine the impact that these may have in the future – include thoughts in Sway write-up.</li> <li>• Analyse existing movie posters to inform a design of their own.</li> <li>• Analyse research, ideas of others and feedback to help implement a mobile phone app design.</li> </ul>
<p>           Apply            demonstrate            manipulate            calculate            practise            identify            use         </p>	<ul style="list-style-type: none"> <li>• Apply understanding of a range of issues around 'big data', digital footprint, computer law and malware through a well-designed and connected mind map.</li> <li>• Apply research on future technologies to an Office 365 Sway presentation.</li> <li>• Demonstrate binary calculations through a series of worksheets leading to more advanced tasks.</li> <li>• Apply knowledge of movie posters and graphic editing techniques to their own idea.</li> <li>• Apply a range of research, business and planning tools in small groups to plan and design a new mobile app.</li> </ul>
<p>           Understand            explain            interpret            give examples            estimate            illustrate         </p>	<ul style="list-style-type: none"> <li>• Understand a range of issues around 'big data', digital footprint, computer law and malware through a connected mind map.</li> <li>• Explain some future technologies through research.</li> <li>• Understand basic binary calculations.</li> <li>• Know some key features and editing techniques to produce a movie poster.</li> <li>• Understand the necessary elements of a design for a new mobile app.</li> </ul>
<p>           Remember            list            recognise            define            recall            label         </p>	<ul style="list-style-type: none"> <li>• Recall and list a range of issues around 'big data', digital footprint, computer law and malware through a basic mind map.</li> <li>• Recognise and define some future technologies through research.</li> <li>• Perform some basic binary calculations.</li> <li>• Know some key features and editing techniques to produce a movie poster.</li> <li>• Work to plan key elements of a design for a new mobile app.</li> </ul>

Create hypothesise formulate design imagine compose develop improve	<ul style="list-style-type: none"> <li>• Create a range of electronic, paper, visual or written documents and media to demonstrate key concepts.</li> <li>• Design, develop and improve a range of basic computer programs to meet given briefs.</li> <li>• Use self, peer and teacher feedback to improve work and answers to exam questions.</li> </ul>
Evaluate recommend persuade debate justify assess conclude determine	<ul style="list-style-type: none"> <li>• Draw conclusions on ELCE impacts and discuss these through extended exam questions.</li> <li>• Assess their own and others' exam answers.</li> </ul>
Analyse infer research investigate question appraise examine prioritise organise	<ul style="list-style-type: none"> <li>• Research key terms and uses of computer equipment for a range of purposes.</li> <li>• Investigate the ELCE impacts of computing in a deeper way.</li> <li>• Analyse existing algorithms in order to decompose and abstract from them, and to debug, complete, or build them as a computer program.</li> <li>• Organise program code in logical and structured ways to use common constructs effectively.</li> </ul>
Apply demonstrate manipulate calculate practise identify use	<ul style="list-style-type: none"> <li>• Apply understanding of key words, terms and functionality through exam questions, quizzes, exercise book work, worksheets and activities.</li> <li>• Demonstrate key programming concepts, constructs and functions through the continual development of Python programs.</li> <li>• Apply skills in representation of data to a range of programming and theory tasks.</li> </ul>
Understand explain interpret give examples estimate illustrate	<ul style="list-style-type: none"> <li>• Understand and explain a range of keywords, terms and functionality of systems architecture, memory and ELCE impacts.</li> <li>• Explain the parts of a computer system and how they work together.</li> <li>• Give examples of a range of key programming constructs and terms through introductory textual programming using Python.</li> <li>• Understand and explain a range of keywords and terms around computational thinking, algorithms, flowcharting and pseudocode including the giving of examples of common algorithms in computing and everyday life.</li> <li>• Understand basic binary and data representation from year 8 and extend this to more advanced forms.</li> </ul>
Remember list recognise define recall label	<ul style="list-style-type: none"> <li>• Define a range of keywords and terms about systems architecture, memory and ELCE (ethical, legal, cultural, environmental) impacts.</li> <li>• Recognise parts of a computer system.</li> <li>• Define a range of key programming constructs and terms.</li> <li>• Define a range of keywords and terms around computational thinking, algorithms, flowcharting and pseudocode.</li> <li>• Recall basic binary and data representation from year 8 and repeat/ extend this.</li> </ul>

<b>Create</b> hypothesise formulate design imagine compose develop improve	<ul style="list-style-type: none"> <li>• Create a range of electronic, paper, visual or written documents and media to demonstrate key concepts.</li> <li>• Design, develop and improve a range of more advanced computer programs to meet given briefs.</li> <li>• Use self, peer and teacher feedback to improve work and answers to exam questions.</li> </ul>
<b>Evaluate</b> recommend persuade debate justify assess conclude determine	<ul style="list-style-type: none"> <li>• Draw conclusions on ELCE impacts and discuss these through extended exam questions.</li> <li>• Assess their own and others' exam answers.</li> <li>• Evaluate their programs for successes and improvements.</li> </ul>
<b>Analyse</b> infer research investigate question appraise examine prioritise organise	<ul style="list-style-type: none"> <li>• Research key terms, advantages, disadvantages and comparisons in uses of storage, network setups, security policies.</li> <li>• Investigate the ELCE impacts of computing in a deeper way.</li> <li>• Examine programs to ensure robustness and testing is demonstrated.</li> <li>• Organise program code in logical and structured ways to use constructs effectively.</li> </ul>
<b>Apply</b> demonstrate manipulate calculate practise identify use	<ul style="list-style-type: none"> <li>• Apply understanding of key words, terms and functionality through exam questions, quizzes, exercise book work, worksheets and activities.</li> <li>• Demonstrate more advanced programming concepts, constructs and functions through the continual development of Python programs.</li> <li>• Apply skills in representation of data to a range of programming and theory tasks.</li> <li>• Show programming techniques that allow for robustness, testing and input validation/ sanitisation.</li> </ul>
<b>Understand</b> explain interpret give examples estimate illustrate	<ul style="list-style-type: none"> <li>• Understand and explain a range of keywords and terms about computer storage, networks, security, systems software and ELCE (ethical, legal, cultural, environmental) impacts.</li> <li>• Give examples of a broad range of key programming functions and their purposes through Python and assorted algorithms.</li> <li>• Understand and explain a range of keywords and terms around the production of robust programs and facilities of programming languages.</li> <li>• Understand data representation (base 2, base 10 and base 16) and build on these to be able to answer exam questions and perform addition and shifts.</li> </ul>
<b>Remember</b> list recognise define recall label	<ul style="list-style-type: none"> <li>• Define a range of keywords and terms about computer storage, networks, security, systems software and ELCE (ethical, legal, cultural, environmental) impacts.</li> <li>• Define a broad range of key programming functions and their purposes.</li> <li>• Define a range of keywords and terms around the production of robust programs and facilities of programming languages.</li> <li>• Recall forms of data representation from year 9 and build on these to a more advanced level.</li> </ul>

<b>Create</b> hypothesise formulate design imagine compose develop improve	<ul style="list-style-type: none"> <li>• Create a range of electronic, paper, visual or written documents and media to demonstrate key concepts.</li> <li>• Use self, peer and teacher feedback to improve work and answers to exam questions.</li> </ul>
<b>Evaluate</b> recommend persuade debate justify assess conclude determine	<ul style="list-style-type: none"> <li>• Evaluate program planning and actions through the programming project write-up.</li> <li>• Evaluate answers in exam questions to improve them, especially for longer questions.</li> </ul>
<b>Analyse</b> infer research investigate question appraise examine prioritise organise	<ul style="list-style-type: none"> <li>• Analyse program planning and actions through the programming project write-up.</li> <li>• Analyse command words in exam questions to improve answers, especially for longer questions.</li> </ul>
<b>Apply</b> demonstrate manipulate calculate practise identify use	<ul style="list-style-type: none"> <li>• Apply understanding of key words, terms and functionality through exam questions, quizzes, exercise book work, worksheets and activities.</li> <li>• Demonstrate more advanced programming concepts, constructs and functions through the programming project and write-up.</li> <li>• Show programming techniques that allow for robustness, testing and input validation/ sanitisation, explaining the uses of these through the programming project.</li> <li>• Practise exam questions regularly.</li> </ul>
<b>Understand</b> explain interpret give examples estimate illustrate	<ul style="list-style-type: none"> <li>• Understand and explain a range of keywords and terms regarding any remaining aspect of the specification that has not otherwise been covered.</li> <li>• Give examples and explain a broad range of key programming functions and their purposes through the programming project and write-up.</li> <li>• Understand and be able to explain key words and terms through revision in preparation for the exams.</li> </ul>
<b>Remember</b> list recognise define recall label	<ul style="list-style-type: none"> <li>• Define a range of keywords and terms regarding any remaining aspect of the specification that has not otherwise been covered.</li> <li>• List and define a broad range of key programming functions and their purposes through the programming project and write-up.</li> <li>• Recall and define key words and terms through revision in preparation for the exams.</li> </ul>