



At St Peter's we believe that a broad and balanced curriculum with a strong academic core is a right for all pupils. We seek to encourage pupils to explore subjects of interest around their in-school learning and to enhance their curriculum experience through enrichment.

Autumn 1	Autumn 2 & Spring 1		Spring 2	Summer 1	Summer 2
Unit of Work/Big Question	Unit of Work/Big Question	Unit of Work/Big Question	Unit of Work/Big Question	Unit of Work/Big Question	Unit of Work/Big Question
<p>E-Safety & Ethics</p> <p>What issues do we currently face due to computing devices?</p>	<p>Photoshop for Industry</p> <p>How are images manipulated in the media industry?</p>	<p>Data Manipulation</p> <p>How can data be presented in spreadsheet software to help draw conclusions and identify insights?</p>	<p>Computational Logic and Sorting</p> <p>How are logic gates and sorting algorithms used to solve complex problems in computing?</p>	<p>User Interface Design</p> <p>How can we design a user-friendly, accessible user interface that meets the needs of its target audience?</p>	<p>Python Programming</p> <p>How can we solve complex problems through coding?</p>
Knowledge	Knowledge	Knowledge	Knowledge	Knowledge	Knowledge
<p>Definition and examples of fake news</p> <p>Detection techniques (fact-checking, source analysis)</p> <p>Impact of fake news (social, political, personal)</p> <p>Understand online gaming types and popularity</p> <p>Understand Gambling mechanics in games (loot boxes, microtransactions)</p> <p>Understand the risks of gambling addiction and financial consequences</p> <p>Strategies for healthy gaming habits</p> <p>Understand the concept of privacy in the digital age</p> <p>Understand types of surveillance (government, corporate, personal)</p> <p>Understand the risks to personal privacy and security</p> <p>Know strategies to protect personal information</p> <p>Know major computer-related laws (Computer Misuse Act, Data</p>	<ul style="list-style-type: none"> • Definition and characteristics of bitmap images • Differences between bitmap and vector images • Common bitmap file formats (e.g., JPEG, PNG, GIF) • Understanding image resolution and its impact on quality • Know when to use the Magnetic Lasso Tool for edge detection and complex shapes • Know the different types of Erasers (Standard, Background, Magic Eraser) • Know when to use the Standard Eraser for basic erasing tasks • Know when to use the Background Eraser to remove backgrounds while preserving foreground elements • Know when to use Magic Eraser for quick, colour-based erasing • Adjusting eraser settings (size, hardness, opacity) for different effects • Know how the Healing Brush Tool works to blend and repair image areas 	<ul style="list-style-type: none"> • Understanding how to import and format data from various file types (e.g., CSV, Excel) into spreadsheet software. • Understand the need to clean and pre-process data, ensuring accuracy and consistency. • Understand the need for sorting, filtering, and using pivot tables to transform and organise data. • Understand when to use spreadsheet formulae (e.g., SUM, AVERAGE, VLOOKUP) to summarise and analyse data. • Understanding the principles of effective dashboard design to create user-friendly and informative dashboards. • Know how to use various visualisation tools (e.g., bar charts, line graphs, pie charts) within spreadsheet software to represent data. • Know how to identify potential errors or anomalies in the data and understand their impact. 	<ul style="list-style-type: none"> • Know about the fundamental components of digital circuits. • Understanding various types of logic gates: AND, OR, NOT. • Know symbols and truth tables associated with each logic gate. • Understand how logic gates execute Boolean operations. • Recognize the importance of truth tables in logic. • Understand logical expressions using truth tables. • Know how to interpret logical equivalences through truth tables. • Know how to assemble logic gates to create complex circuits. • Know how to design logic circuits tailored to specific functions. • Understand the analytical process of logic circuits using truth tables. • Comprehend sorting algorithms and their significance. • Understand Bubble Sort: its 	<ul style="list-style-type: none"> • Know the purpose and functionality of a user interface in a museum setting. • Understand how to identify and analyse the needs of the target audience. • Know how to incorporate user accessibility requirements into the design. • Understand how to ensure the interface is easy to use and facilitates quick navigation. • Understand how to address accessibility needs in the design process. • Know how to determine and address audience requirements and user accessibility needs. • Understand how to compile and manage a comprehensive task list. • Understand how to create mood boards and mind maps to visualise design ideas. • Know how to develop initial design concepts for the user interface. 	<ul style="list-style-type: none"> • Know what Syntax is and why it is required. • Understand why variables are needed. • Understand how to initiate different data types in Python. • Understand what a Conditional Statement is and when to use it. • Understand what a FOR Loop is and when to use it. • Understand what a WHILE Loop is. • Know how a WHILE Loop could be used in a program. • Understand what a Function is. • Know what a Procedure is. • Know the difference between a function and a procedure. • Know the use of variables, constants, operators, inputs, outputs and

<p>Protection Act, Copyright Laws) Types of cybercrime and legal consequences Understanding electronic waste (e-waste) and its environmental impact Know the importance of recycling and reusing electronic components</p>	<ul style="list-style-type: none"> Know the difference between the Healing Brush Tool and the Spot Healing Brush Tool Know when to use the Healing Brush Tool to remove blemishes, scratches, and other imperfections Know different types of brushes and their effects (e.g., soft round, hard round, specialty brushes) Understanding brush blending modes and their impact on layers Understand practical applications of the Brush Tool in digital painting and photo editing 	<ul style="list-style-type: none"> Know how to interpret data trends and patterns to draw meaningful conclusions. Know how to use screenshots and visual evidence from dashboards to support analysis and conclusions. Know how to assess the effectiveness of different data presentation methods and their impact on data communication. 	<p>functionality and core algorithm.</p> <ul style="list-style-type: none"> Know the algorithm for Bubble Sort in a programming language. Understand Merge Sort: its principles and algorithms. Know the algorithm for Merge Sort in a programming language. Know Insertion Sort: its concepts and algorithms. Know the algorithm for Insertion Sort in a programming language. 	<ul style="list-style-type: none"> Understand how to detail user requirements, audience considerations, accessibility, and purpose in the design specification. Know how to specify input and output requirements for the system. Understand how to identify and select appropriate hardware and software for creating the interface. Know how to design and create storyboards for at least four different screens. Understand how to ensure the interface enhances user attention, confidence, and familiarity. Understand how to address and mitigate security issues related to the touch screen information system. Understand how to apply design principles to create user-friendly interfaces. 	<p>assignments</p> <ul style="list-style-type: none"> Know the use of the three basic programming constructs used to control the flow of a program: <ul style="list-style-type: none"> Sequence Selection Iteration (count- and condition-controlled loops) Know common arithmetic operators Know the common Boolean operators AND, OR and NOT
<p>Skills & Procedural Knowledge</p>	<p>Skills & Procedural Knowledge</p>	<p>Skills & Procedural Knowledge</p>	<p>Skills & Procedural Knowledge</p>	<p>Skills & Procedural Knowledge</p>	<p>Skills & Procedural Knowledge</p>
<p>Skills</p> <ul style="list-style-type: none"> Analytical thinking Creative thinking Understanding what risks are out there <p>Word Processing</p> <ul style="list-style-type: none"> Text font, size, layout Line spacing Bullet points and Numbers Spell check Word count Page numbers Date/Time Header & Footer Images Tables Special characters Horizontal line 	<ul style="list-style-type: none"> Assessing image resolution and quality Differentiating between bitmap and vector images Opening and saving bitmap image files Checking and adjusting image resolution Converting between different bitmap file formats Modifying selection shapes and edges Using keyboard shortcuts for efficiency Selecting areas with Rectangular and Elliptical Marquee Tools Feathering and refining selection edges Creating freehand selections with the Standard Lasso Tool 	<ul style="list-style-type: none"> Importing and formatting data from various file types (e.g., CSV, Excel) into spreadsheet software. Cleaning and preprocessing data to ensure accuracy and consistency. Sorting, filtering, and using pivot tables to transform and organise data effectively. Applying spreadsheet formulae (e.g., SUM, AVERAGE, VLOOKUP) to summarise and analyse data appropriately. Effectively design a dashboard to create a user-friendly and informative way to share information. Utilising various visualisation tools (e.g., bar charts, line 	<ul style="list-style-type: none"> Comprehend the fundamental components and principles of digital circuits. Identify and understand various types of logic gates (AND, OR, NOT, etc.). Recognize symbols and construct truth tables associated with each logic gate. Understand and execute Boolean operations using logic gates. Analyse logical expressions and circuits through truth tables. Assemble logic gates to create complex circuits for specific functions. Design logic circuits tailored to meet specific requirements and functions. 	<ul style="list-style-type: none"> Define the purpose and functionality of a user interface. Identify and analyse the needs of the target audience. Incorporate user accessibility requirements into the design. Ensure the interface is easy to use and facilitates quick navigation. Address accessibility needs in the design process. Determine and address audience requirements and user accessibility needs. Compile and manage a comprehensive task list. Create mood boards and mind maps to visualise design ideas. Develop initial design concepts 	<ul style="list-style-type: none"> Read existing Python code and understand its functionality. Write Python code using correct syntax. Declare and initialise variables. Use variables to store and manipulate data. Update variable values during program execution. Identify and use different data types: integers, floats, strings, and Booleans. Perform operations specific to each data type. Write if, elif, and else

<ul style="list-style-type: none"> • Margins, Orientation, page colour <p>Presentation software</p> <ul style="list-style-type: none"> • Master slides • Layout • Slide transitions • Slide animations • Images • Test font, size, layout • Interactivity • Hidden slides 	<ul style="list-style-type: none"> • Drawing straight-edged selections with the Polygonal Lasso Tool • Following edges with the Magnetic Lasso Tool • Selecting complex shapes with the Lasso Tool • Adjusting and refining Lasso selections • Combining Lasso selections with other selection tools • Erasing parts of an image with precision • Applying non-destructive erasing techniques • Erasing with the Standard, Background, and Magic Erasers • Blending image areas using the Healing Brush Tool • Applying the Healing Brush to remove imperfections • Using the Patch Tool for complex repairs • Customising brush settings for various effects • Changing brush size, hardness, opacity, and flow • 	<p>graphs, pie charts) within spreadsheet software to represent data effectively.</p> <ul style="list-style-type: none"> • Identifying potential errors or anomalies in data and understanding their impact on analysis. • Interpreting data trends and patterns to draw meaningful conclusions. • Using screenshots and visual evidence from dashboards to support analysis and conclusions effectively. • Assessing the effectiveness of different data presentation methods to improve data communication strategies. 	<ul style="list-style-type: none"> • Analyse logic circuits and logical expressions using truth tables. • Apply Boolean algebra principles to design efficient logic circuits. • Develop a comprehensive understanding of various sorting algorithms and their practical significance. • Analyse the time complexity and efficiency of sorting algorithms. • Implement sorting algorithms using a programming language. 	<p>for the user interface.</p> <ul style="list-style-type: none"> • Detail user requirements, audience considerations, accessibility, and purpose in the design specification. • Specify input and output requirements for the system. • Identify and select appropriate hardware and software for creating the interface. • Design and create storyboards for at least four different screens. • Ensure the interface enhances user attention, confidence, and familiarity. • Apply design principles to create user-friendly interfaces. 	<p>statements to control the flow of the program.</p> <ul style="list-style-type: none"> • Evaluate conditions using comparison operators (<code>==</code>, <code>!=</code>, <code><</code>, <code>></code>, <code><=</code>, <code>>=</code>). • Use logical operators (and, or, not) within conditions. • Write for loops to iterate over sequences (e.g., lists, strings). • Write while loops to perform repeated actions based on a condition. • Use loop control statements (break, continue) to manage loop execution. • Define functions with appropriate parameters. • Call functions with arguments. • Use return statements to return values from functions. • Define and use procedures (functions without return values). • Differentiate between functions and procedures in terms of usage and output. • Identify and fix syntax errors in the code. • Use debugging techniques to locate and resolve logic errors. • Implement basic error handling using try and except blocks. • Break down problems into smaller, manageable tasks. • Develop algorithms to solve specific problems. • Implement algorithms in Python to achieve desired outcomes.
---	--	--	--	--	---

Tier 3 Subject Vocabulary	Tier 3 Subject Vocabulary	Tier 3 Subject Vocabulary	Tier 3 Subject Vocabulary	Tier 3 Subject Vocabulary	Tier 3 Subject Vocabulary
Misinformation Disinformation Fact-checking Credibility Bias Algorithm Echo chamber Verification Satire Hoax Online Gaming and the Link to Gambling Loot box In-app purchase Skin betting Pay-to-win Esports Digital currency Metadata Encryption Biometric data GDPR (General Data Protection Regulation) Firewall Two-factor authentication (2FA) Surveillance capitalism Privacy policy Cybercrime Intellectual property Phishing Malware Ransomware Data breach Copyright infringement Hacking E-waste Sustainability Carbon footprint Energy efficiency Digital sustainability	Bitmap images Vector images Image resolution Magnetic Lasso Tool Standard Eraser Background Eraser Magic Eraser Eraser settings Healing Brush Tool Spot Healing Brush Tool Soft round Hard round Specialty brushes Brush blending modes Saturation Hue Contrast Exposure Cloning Histogram Curves adjustment Dodge and Burn Gradient Layer masking Opacity Sharpening Resampling Lens correction Non-destructive editing	Cell Range Conditional formatting Data validation Function Chart Trendline Data label Legend Pivot chart Freeze panes Data analysis Import Format Clean Pre-process Sort Filter Pivot table Formula Summarise Analyze Dashboard Visualisation Anomaly Interpret Assess	Truth Table Computational Not AND OR Logic Gate Expression Truth Table Combination Boolean Logic Circuit Complexity Algorithm Insertion Merge Bubble Sorting Computation	User Interface (UI) Accessibility Navigation Mood Board Mind Map Design Specification Storyboard Input Requirements Output Requirements Hardware Software Usability Prototyping Annotated Maps Milestones Gantt Chart SMART Objectives User-centred Design Task Analysis Information Architecture Wireframe User Experience (UX) Heuristic Evaluation Interaction Design Interface Testing	Algorithm Python IDE Array Boolean Concatenation Data Type Debugger Exception Float Function Immutable Indentation Integer Iterable Library Loop Method Module Parameter Recursion String Syntax Tuple Variable Input Output Selection Iteration Section Count Random
Key Assessment Task (KAT)	Key Assessment Task (KAT)	Key Assessment Task (KAT)	Key Assessment Task (KAT)	Key Assessment Task (KAT)	Key Assessment Task (KAT)
End of Unit Assessment	Production and analysis of data sets using spreadsheet software. Spreadsheet formulas assessment. Practical Assessment		End of Unit Assessment	Practical Assessment	End of Unit Assessment



St Peter's Collegiate Academy

At St Peter's we believe that a broad and balanced curriculum with a strong academic core is a right for all pupils. We seek to encourage pupils to explore subjects of interest around their in-school learning and to enhance their curriculum experience through enrichment.

Yr 9 Computing Progress Ladder

	E-Safety & Ethics	Photoshop for Industry	Spreadsheet Project	Computational Logic & Sorting	User Interface Design	Python Programming
--	-------------------	------------------------	---------------------	-------------------------------	-----------------------	--------------------

Mastering	<p>You confidently use the internet safely, protecting personal information and respecting others online. To stay at the top, keep up with the latest e-safety practices and help others learn.</p>	<p>You can create professional-quality designs with advanced techniques in Photoshop. To keep growing, focus on staying updated with the latest features and trends, and consider teaching others to solidify your knowledge.</p>	<p>You expertly use advanced spreadsheet features like pivot tables, macros, and complex formulas. To stay at the top, keep exploring new functions and techniques, and help others improve their spreadsheet skills by sharing your knowledge and tips.</p>	<p>You can efficiently use advanced sorting algorithms and understand complex computational logic. To stay at the top, keep exploring more intricate algorithms, optimise your code further, and start learning about algorithm analysis and computational complexity.</p>	<p>You create user interfaces that are both functional and visually appealing, ensuring a smooth and intuitive user experience. To stay at the top, keep exploring new design trends and usability testing methods to further refine your skills.</p>	<p>You write complex Python programs efficiently and understand advanced concepts like decorators and generators. To stay at the top, keep learning about new libraries and frameworks, and challenge yourself with bigger projects and contribute to open-source communities.</p>
Advancing	<p>You understand and practise good e-safety habits and show respect in online interactions. To reach the next level, stay updated on new online threats and consistently guide peers in responsible internet use.</p>	<p>You can confidently use most Photoshop tools and create complex projects. To move to mastering, practice refining your skills, explore advanced tutorials, and start experimenting with your own creative ideas.</p>	<p>You can use intermediate features like charts, conditional formatting, and complex formulas. To reach the next level, focus on mastering advanced tools like pivot tables and macros, and start automating repetitive tasks to enhance your efficiency.</p>	<p>You can implement intermediate sorting algorithms and apply computational logic to solve problems. To reach the next level, focus on mastering advanced sorting techniques and understanding how to analyse the efficiency of different algorithms.</p>	<p>You design interfaces that are easy to use and look good. To reach the next level, focus on advanced techniques like responsive design and user feedback analysis to make your interfaces even better.</p>	<p>You can create functional programs using intermediate Python features like classes and modules. To reach the next level, focus on mastering advanced concepts, optimising your code, and exploring popular libraries like NumPy and pandas.</p>
Securing	<p>You know how to keep your information private and behave responsibly online. To improve, make sure you regularly check privacy settings and learn more about preventing and responding to cyberbullying.</p>	<p>You can complete intermediate projects and understand the core functions of Photoshop. To advance, challenge yourself with more complex tasks, learn about advanced tools, and seek feedback to improve your work.</p>	<p>You understand how to use basic functions and create simple charts. To improve, practice using more advanced functions and start exploring how to use conditional formatting and data validation to make your spreadsheets more powerful and accurate.</p>	<p>You understand basic sorting algorithms and can use simple computational logic in programs. To improve, practice implementing more efficient sorting methods and start exploring how different algorithms solve problems differently.</p>	<p>You can design basic interfaces that work well and look clean. To improve, start paying more attention to details like colour schemes, typography, and user feedback to enhance the overall experience.</p>	<p>You understand basic Python syntax and can write simple programs. To improve, practice using loops, functions, and lists in more complex programs, and start learning about object-oriented programming and error handling.</p>

<p>Developing</p>	<p>You are starting to understand the importance of online safety and respectful behaviour. To move forward, practice creating strong passwords and learn about privacy settings on social media.</p>	<p>You can handle basic tasks and are familiar with essential Photoshop tools. To secure your skills, practice regularly, explore intermediate tutorials, and start working on small projects to build confidence.</p>	<p>You can create and format simple spreadsheets and use basic formulas. To move forward, focus on learning how to create charts, use more functions, and organise your data better to make your spreadsheets more useful and informative.</p>	<p>You know how to use basic sorting techniques and apply simple logic to solve problems. To move forward, concentrate on learning about different sorting algorithms and practice using them to sort various data sets.</p>	<p>You understand the basics of creating simple and functional interfaces. To move forward, practice arranging elements in a user-friendly way and learn about the importance of consistency and accessibility in design.</p>	<p>You know the basics of Python and can write very simple programs. To move forward, focus on learning about loops, conditionals, and functions, and practice combining these elements to create more useful programs.</p>
<p>Emerging</p>	<p>You are learning the basics of staying safe online and treating others kindly. To progress, focus on recognizing unsafe websites and understanding the importance of keeping personal information private.</p>	<p>You are just starting to learn Photoshop and can perform simple edits. To develop further, focus on learning the basic tools, follow beginner tutorials, and practice by working on easy projects to build a strong foundation.</p>	<p>You are beginning to learn about spreadsheets and can enter and organise data. To progress, concentrate on understanding basic formulas and functions, and practice formatting your spreadsheets to make the data clearer and easier to read.</p>	<p>You are beginning to learn about sorting and computational logic and can follow simple examples. To progress, focus on understanding the basics of sorting data and start practising simple logical steps to solve small problems.</p>	<p>You are learning the fundamentals of user interface design and can create basic layouts. To progress, focus on understanding the needs of users and start experimenting with different design elements to improve usability and aesthetics.</p>	<p>You are beginning to learn Python and can understand basic commands. To progress, concentrate on learning the core syntax and practising writing small programs to understand how variables, loops, and conditionals work.</p>