Prince William School
Computer Science & IT Curriculum Overview

Why Teach COMPUTER SCIENCE?

We believe that Computer Science is a vital subject for pupils to learn within the modern world, where computing and technology are embedded in everyday life. By learning Computer Science learners will study:-

- How to recognise online dangers and stay safe within the online world

- How computers work

- How they are programmed

- Computational thinking skills they can use in many other areas

- How to use key programs to present and share work

- Understand how digital technology is vital for modern careers

Disciplinary Big Ideas

Within Computer Science we develop learner's disciplinary knowledge by the following techniques to help build their ability to fully apply the substantive knowledge.

The Big Substantive Ideas of the Computer Science Curriculum

Curriculum maps detail the sequencing of substantive knowledge from Computer Science to enable pupils to build schemata of important concepts over time through 4 'big ideas'.

Learning for Life and Careers

Employability skills

Learning to program builds characteristics such as collaboration, communication, creativity, critical thinking, and resilience. Additionally, this subject develops numeracy and literacy, presentation skills, and the ability to analyse and evaluate

Linking the curriculum to careers

Careers in computing, engineering, IT, data management and security.

Examples of qualification pathways

At KS4 we offer the BTEC Tech Award in Digital Information Technology (DIT) as well as the GCSE Computer Science course and offer A level Computer Science at KS5. BTEC Tech Award in Digital Information Technology (DIT) progresses well to the Level 3 Extended Certificates, which in turn progresses well to study at Higher Nationals. The Computer Science GCSE progresses naturally to the Computer Science A level or professional qualifications such as CCNA, which in turn can lead to further study at degree level.
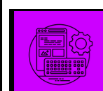
Prince William School
Computer Science & IT Curriculum Map by Term

Legend: Programming | E-Safety | Technology | Data

Term	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
Autumn 1	Computer Usage; E-Safety; IT: Uses of e mail	E-Safety; Computational Thinking	E-Safety; Python Review	Data storage-Numbers; Computational thinking Designing, creating and refining algorithms; Programming fundamentals: Intro & Review; Data types; Programming fundamentals: Sequence & Selection	Architecture - CPU Performance; Data storage-Images Data storage-Sound Data storage-Compression	Programming basics: Selection, Iteration, Arrays, Subroutines, Files, Exceptions; Computational Thinking; Structured programming; Algorithms; Testing; Abstraction, Automation	Databases, ERDs, Normalisation, SQL Project skills – SQLite Systemic approach Data structures – Queues, Lists, Stacks, Hash tables; Communications methods Network topologies Client-server and a peer-to-peer networks Wireless networks Communications & privacy
Autumn 2	History and Future of Computing; Computational Thinking; IT: Research skills and use of PowerPoint	Games Programming in Scratch	Intermediate Python	Programming fundamentals: Selection Programming fundamentals: Iteration - Count Programming fundamentals: Iteration - Condition	Embedded Systems Memory Storage	Finite state Machines; Social, legal and cultural issues; Number Systems, Binary & Units, Binary Arithmetic, Floating point Images, Sound, Compression	Data structures –Graphs, Trees, Vectors; Recursion; Big-O Searching / Sorting Graph traversal Optimisation Algorithms
Spring 1	Kodu – an introduction to programming; IT: Uses of word processing	Scratch Project	Intermediate Python	Additional programming - Arrays Additional programming techniques - Sub programs	Internet / WWW LANs Wireless Client-server and a peer-to-peer networks	Hardware & software Operating systems, Language classification, Translators Logic gates, Adders, D-type flip-flops Boolean algebra Computer Architecture	Limits of computation; Sets, Regular expressions, Mealy machines, Turing machine
Spring 2	Kodu – an introduction to programming; IT: Uses of graphics software	Introduction to programming in Python	Ethics of computing; Office Skills - Spreadsheets	Data storage-Characters; Additional programming techniques - String manipulation Additional programming techniques - File Handling	Protocols Network threats Preventing vulnerabilities	CPU,; Instruction sets Assembly Language OOP basics OOP Design Principles	BNF, Reverse Polish The internet, packet switching, routers; Functional programming Big Data
Summer 1	Computer System Basics; IT: Uses of infographics	Introduction to programming in Python	Office Skills - Spreadsheets, Presentations	Searching - Binary & Linear Sorting - Merge & Insertion Additional programming techniques - Record/SQL Defensive design Testing Languages Boolean logic	Operating Systems Utilities	I/O devices Secondary storage Communications methods Network topologies Client-server and a peer-to-peer networks	Network security TCP & protocols IP addresses Client-server model
Summer 2	Office skills- Spreadsheets; IT: Project to combine skills learnt during the year	Computational Thinking: Binary; Hardware & I/O devices	Office Skills – Presentations, Word Processing	Ethical, legal, cultural and environmental impact; Languages The Integrated Development Environment (IDE)		Wireless networks Communications & privacy; Project introduction Project skills – TK, Pygame	

Programming	E-Safety	Technology	Data

Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
Kodu – an introduction to programming: *learning the basics of programming such as sequence and selection* Kodu Project: *applying the skills learnt to make their own game* Computational Thinking: *understand the power of problem solving and have an appreciation of the rapidly changing technologies that have brought the computer to where it is today.*	Games Programming in Scratch: *building on the Yr7 work to include iteration and variables* Scratch Project: *applying the skills learnt to make their own game* Introduction to programming in Python: *building on the previous programming knowledge/skills and learning about a text-based programming language and further programming features including* • *variables* • *I/O* • *Sequence* • *Selection – IF / ELSE* • *Iteration - FOR* Computational Thinking: *understand the power of problem solving and the different methods that Computer Scientists use to tackle problems.* *An introduction to binary and how a digital processor works.*	HTML: *learning about websites and some HTML basics* Python Review / Intermediate Python: *recapping & building on the previous work to include* • *Selection – ELIF* • *Iteration – While* *Ext:* • *Arrays* • *Subroutines*	*Putting the KS3 work into context and extending it.* Computational thinking Designing, creating and refining algorithms Programming fundamentals – *recapping the KS3 work and extending knowledge, skills and complexity:* Intro & Review Data types Sequence Selection Iteration - Count Iteration - Condition *Learning & applying additional programming techniques:* Arrays Sub programs String manipulation File Handling Record/SQL Standard algorithms – *learning about the standard algorithms and how the skills they've learnt are applied to them:* Searching - Binary & Linear Sorting - Merge & Insertion *Learn more about the creation, testing and development environment:* Defensive design Testing Language facilities	*Building on the Boolean work so far and formalising it:* Boolean logic *Practice of skills learned in Yr10*	*A recap and applying the GCSE skills in more complex situations* Programming basics: • Selection • Iteration • Arrays • Subroutines • Files • Exceptions *Extending the knowledge to do with programming theory, logic and data structures used:* Computational Thinking Structured programming Algorithms Testing Abstraction, Automation Instruction sets Assembly Language *Learning a different programming paradigm - Objects* OOP basics OOP Design Principles *Understand the expectation for the NEA project and develop further suitable skills* Project introduction Project skills – TK, Pygame	*Further programming theory and applications* Recursion Big-O Searching / Sorting Graph traversal Optimisation Algorithms Limits of computation Mealy machines Turing machine *A further programming paradigm* Functional programming

Programming		E-Safety		Technology		Data

Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
Cyberbullying *what it is and what to do about it* Social Networking *issues and guidance for use* Digital Footprint *what it is and how to be mindful about it*	Searching/Sources *tips for searching techniques and understanding issues on the reliability of sources, esp. online* Grooming (Sexual) *what it is and what to do about it* Sexting *what it is and what to do about it*	Grooming (Extremism) *what it is, things to watch out for and what to do about it* Digital Citizen *being a good digital citizen and understanding how being online can be used for good (incl. "Upskirting")*	*Know and understand the* Impacts of digital technology on wider society including: • Ethical issues • Legal issues • Cultural issues • Environmental issues • Privacy issues	*Know and understand the impact of* Legislation relevant to Computer Science: • The Data Protection Act 2018 • Computer Misuse Act 1990 • Copyright Designs and Patents Act 1988 • Software licences (i.e. open source and proprietary)	*Know and understand the issues around the* Consequences of uses of computing - • Individual (moral), social (ethical), legal and cultural issues and opportunities: • Awareness of current individual (moral), social (ethical), legal and cultural opportunities and risks of computing.	Big Data - *understand what "Big Data" is, it's characteristics and how it is collected and analysed.*
Computer Usage *know about the operating system, file and folder management and networked resources* Computer System Basics *know the basics components of a computer system* *E mail Use:* *Know the purpose of e mail and how to use effectively*	Languages *understand basic features of programming languages* The Integrated Development Environment (IDE) *know about basic features of the programming development environments*	Hardware review & I/O devices *know the basic system components, esp. input & output devices* Ethics: Investigate some of the wider ethical issues in computing, including the moral, environmental and legal issues that can arise in the digital age.	Languages *understand key features of programming languages* The Integrated Development Environment (IDE) *know about key features of the programming development environment*	*Know about computer systems, hardware software and networked systems:* Architecture - CPU Performance Embedded Systems Memory Storage Internet / WWW LANs Wireless Client-server and a peer-to-peer networks# Protocols Network threats Preventing vulnerabilities Operating Systems Utilities	*Know about computer systems, hardware software and networked systems in more detail:* Hardware & software Operating systems, Language classification, Translators Logic gates, Adders, D-type flip-flops Boolean algebra Computer Architecture CPU I/O devices Secondary storage	*Know about networked systems in more detail:* Communications methods Network topologies Client-server and a peer-to-peer networks Wireless networks Communications & privacy The internet, packet switching, routers Network security TCP & protocols IP addresses Client-server model

Programming		E-Safety		Technology		Data

Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
Office skills- Spreadsheets *learn basic formatting, charts, formula & functions* *IT: use of word processing, presentation and graphics software to produce appropriate resources.*	Scratch & Python programming *Identify variables and values to be stored for different scenarios*	Office Skills - Intermediate Spreadsheets *review & learn further spreadsheet skills* Presentations, Word Processing *learn more intermediate and some advanced features of a word processor and presentations software*	Data storage-Numbers *Data representation in a computer system:* Units	*Data representation in a computer system:* Data storage-Characters Data storage-Images Data storage-Sound Data storage-Compression	*Build on and extend the GCSE knowledge of data representation* Number Systems, Binary & Units, Binary Arithmetic, Floating point Images, Sound, Compression *learn about new methods of working with data* Finite state Machines	*learn more about how data is stored and interrogated* Databases, ERDs, Normalisation, SQL *Link theory to practical skills that could be used for the NEA project* Project skills – SQLite; TKinter Systemic approach *learn about new methods of working with data* Data structures – Queues, Lists, Stacks, Hash tables Data structures –Graphs, Trees, Vectors Sets Regular expressions BNF, Reverse Polish Big Data

Prince William School
COMPUTER SCIENCEX Curriculum Map – Disciplinary Knowledge Progression

	Good Online Practices		Computer Programming		Computer Systems and Networks		ITC Literacy

Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13

• Create strong passwords
• Not to share passwords with others and the consequences of doing so
• Reading EULAs
• Organise work with folder structures and files
• Identify cyberbullying and how to deal with it
• Identify personal information and how to be careful with it
• Report issues about using the internet to trusted adults
	• learn online searching skills
	• understand the reliability of online sources and how to check them
	• understand what sexual grooming and sexting is and how to deal with it
	• understand what sexting is and how to deal with it
	
	Report issues about using the internet to trusted adults
		• understand what extremism and radicalisation are and how to deal with them
		
		Understand how to be a good Digital Citizen and the drawbacks and benefits of being online
			• Understand the impacts of digital technology on wider society including:
			  • Ethical issues
			  • Legal issues
			  • Cultural issues
			  • Environmental issues
			  • Privacy issues
			• Create secure programs that are safe for users to use
				• Learn about legislation relevant to Computer Science and understand it relevance:
				  • The Data Protection Act 2018
				  • Computer Misuse Act 1990
				  • Copyright Designs and Patents Act 1988
				  • Software licences (i.e. open source and proprietary)
				
				Identify flaws in cybersecurity within networks and suggest improvements
					• Understand the consequences of uses of computing -
					  • Individual (moral), social (ethical), legal and cultural issues and opportunities:
					  • Awareness of current individual (moral), social (ethical), legal and cultural opportunities and risks of computing.
					• Create secure programs that are safe for users to use
					  •
						• Understand what "Big Data" is, it's characteristics and how it is collected and analysed and it's impact
						• Develop secure networks which allow network users to be safe

• Put together instructions in a logical order
• Understand the basic programming constructs of sequence and selection
• Drag and drop blocks to create programs in Kodu
	• Analyse problems
	• Break down problem into smaller parts
	• Introducing binary numbers and how to convert from denary
	• Hoe binary is used to represent images
	• Understand the basic programming constructs of sequence, selection and count-controlled iteration
	• Drag and drop blocks to create programs in EduBlocks
	• Using increasing amount of text-based coding by learning Python.
		• Increase the use of text-based programming
		• Further develop the use of programming constructs of sequence, selection and count-controlled and condition-controlled iteration in Python
			• Understand and use Computational thinking
			• Design, create and refine algorithms
			• Review and further practice the programming fundamentals:
			  • Data types
			  • Sequence
			  • Selection
			  • Iteration - Count
			  • Iteration – Condition
			• Learn additional programming techniques:
			  • Arrays
			  • Sub programs
			  • String manipulation
			  • File Handling
			• Records/SQL
				• Further develop the practice of programming for real-life situations	• Review and further develop the practice of programming skills
					• Systems analysis
					• Understand the features and development of Structured programming
					• Learn Algorithms
					• Understand instruction sets and the basics of machine code
					• Learn the development and use of assembly language
					• Learn new programming paradigm of Object-Oriented Programming (OOP)
					• Finding solutions to meet 'client' requirements
						• Learn about recursive programs
						• Analyse the time-complexity of programs using Big-O notation
						• Be able to write Searching / Sorting algorithms
						• Program Graph traversal & Optimisation Algorithms
						• Learn about the limits of computation
						Learn functional programming and it's application to Big Data

Prince William School
COMPUTER SCIENCEX Curriculum Map – Disciplinary Knowledge Progression Continued

	Good Online Practices		Computer Programming		Computer Systems and Networks		ITC Literacy

Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
• Learn about computer system basics • Understand key hardware components, what they do and how they link together • Learn about computers in the past, present and future • Learn how to write an effective email • Learn how to manage e mail effectively	• Understand key storage devices and how they work	• How you use technology responsibly and ethically • Technology and the environment • Technology and the law	• Understand how data is represented in a computer and it's impact on programs, such as data types	• Understand the architecture of a computer, especially the CPU, it's key components, such as registers and cache, and how they affect system performance • Understand the different memory types of RAM & ROM • Understand the need for and the different types of storage – magnetic, solid state and optical and choose the appropriate one for specific scenarios • Understand network types e.g. LAN/WAN or C-S/P2P and recommend appropriate use • Understand the issues of network threats and use appropriate techniques to prevent vulnerabilities	• Understand Hardware & software • Describe Operating systems, Language classification & Translators • Work with Logic gates, Adders, D-type flip-flops for specific scenarios • Use Boolean algebra to rearrange or simplify expressions • Know Computer Architecture • Be able to describe in detail the specifics of certain I/O or secondary storage devices	• Choose the appropriate network topology and components for different situations • Understand how the internet works with packet switching and routers • Understand the issues of network security and use appropriate techniques to prevent vulnerabilities • Understand the need for protocols and the TCP/IP stack • Know how IP addressing works and use subnet masks to create subnets for specific situations
• Learn the basics of spreadsheets • Be able to use simple and intermediate spreadsheet formulae and functions • Create graphs/charts • Learn and use the basics of word processing • Learn and use the basics of graphics software • Learn and use the basics of presentation software		• Further develop Office skills • Learn about Presentation techniques • Learn how to create Presentations that are fit for purpose • Learn Word Processing techniques • Be able to use intermediate and advanced spreadsheet formulae and functions • Create documents of appropriate type and content	• Use key programming techniques to begin to create their own programs • Use appropriate programs to create and organise resources	• Use key programming techniques to begin to create their own programs • Use appropriate programs to create and organise resources	• Use key programming techniques to begin to create their own programs • Use appropriate programs to create and organise resources	• Develop their own app or program to for fill the needs of a specific user • Cite references for research they have used

Prince William School
COMPUTER SCIENCE Key Vocabulary

	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
Autumn 1	**Computer Usage, E-Safety**	**E-Safety**	**E-Safety, Hardware – I/O**	**Data storage-Numbers, Programming fundamentals:**	**Boolean logic, Data representation**	**Programming Basic, Computational thinking**	**Databases, Data structures**
	Organisation	Internet	Grooming	Place value	Boolean	Exceptions	Wireless / Wi-F-
	Files	World wide web (WWW)	Extremism	Denary / decimal	Operators / gates / circuits	Imperative	NIC
	Folders	Searching	Radicalisation	Binary	Truth table	Declarative	CSMA/CA and RTS/CTS
	Cyber-bullying	Source	Rights	Hexadecimal	Units	Abstraction	SSID
	Social networking	Reliability	Responsibilities	Sequence	Characters	Algorithm	Whitelist
	Personal data	Cross-check	"Upskirting"	Data type / casting	Images	Modelling	Prototype
	Trusted adult	Grooming		Variables	Sound	Automation	Waterfall
	Outcome	Sexting		Input / output	Compression	Decomposition	Queues, Lists, Stacks, Hash tables
Autumn 2	**History and Future of Computing**	**Programming in Scratch**	**Websites- HTML, Python review**	**Programming fundamentals:**	**Standard algorithms, Impact**	**FSM, Impact of Computers, Data representation**	**Data structures, Algorithms**
	Word processing software	Sprite	Web server	Selection	Algorithm	Finite State Machine	Graphs, Trees, Vectors
	Templates	Costumes	HTML	If / else / elseif	Searching – linear / binary	Start / End state / Transition	Recursion / recursive
	Design cycle	Movement	Tags	Boolean operator	Sorting – bubble / insertion / merge	Legislation	Stop / base case
	Moore's law	Co-ordinates	Head / Body	Case	Ethical	Number bases	Traversal
	Transistors	Sequence	CSS	Count-controlled Iteration	Legislation	Floating point / Normalisation	Pre / in / post-order
	Microchip	Selection	Variables	For	Cultural	Bit depth	Time complexity
	Algorithm	Lives	Sequence	Condition-controlled Iteration	Environmental	Resolution	Polynomial
	Flowcharts	Variables	Comment	While	Impact	Run length encoding	
Spring 1	**Kodu**	**Scratch**	**Python Intermediate**	**Programming fundamentals:**	**Systems Architecture, Networks, Connections & Topologies**	**Computer Architecture**	**Algorithms**
	Program	Iteration	Selection	Array	Von Neumann	Hardware	Tractable
	Precise	Procedures (Broadcasts)	If	Index	ALU / CU / MAR / MDR /PC /ACC	Software	Heuristic
	Sequence	Randomisation	Else	Length	Primary / secondary storage	OS	Computable / non-computable
	Landscape	Interaction	Elseif	Append	LAN / WAN	Bespoke / Off-the-shelf	Functions / composition
	Navigation	Sounds	Indentation	Subroutine	Ethernet / wireless	Gates	Domain / co-domain
	Automatic	Criteria	Iteration	Procedure	Client server / Peer-to-peer	Boolean	First-class objects
	Manual	Testing	For	Function	Topology	Adders	Regular language
	Clones	Evaluation	Range	Return	DNS / Cloud	Flip-flop	Universal Turing Machine
Spring 2	**Kodu**	**Programming in Python**	**Ethics of Computing, Office Skills - Spreadsheets**	**Programming fundamentals:**	**Protocols and Security**	**Computer Architecture, OOP**	
	Creatables	IDE	Ethics	String	Encryption	Machine code	Backus-Naur Form
	Scoring	Syntax	Copyright	Character	IP / MAC addresses	Assembly language	Revers Polish notation
	Selection	Sequence	Data protection	Concatenate	TCP/IP, HTTP, FTP, POP, IMAP, SMTP	Object	IP address v4 / v6
	Indentation	Debugging	GDPR	Slice	Layers	Class	Domain Name System
	Pages	Parentheses	Formula	File	Malware	Instantiation	Internet Registries
	Depth	Variables	Relative / Absolute	Open / Close	Brute-force / DOS / SQL injection	Inheritance	Routing table
	Complexity	Data types	Function	Read	Anti-malware Software	Polymorphism	Packet switching
	Objectives	Integer	Nested	Write	Firewall	Overriding	Packet header / trailer

Prince William School
COMPUTER SCIENCE Key Vocabulary

	Computer System Basics	Programming in Python	Office Skills - Presentations	Programming fundamentals:	Software	Computer Architecture	
Summer 1	Computer	Float	Presentation	Defensive design	Operating system (OS)	Register	Firewall
	Input	String	Slide	Authentication	User Interface (UI)	Cache	Symmetric / asymmetric encryption
	Process	Input	Layout	Validation	OS functions	Bus	TCP/IP protocol stack
	Output	Arithmetic operators	Design	Record	Utility Software	I/O device	Ports
	Memory	Selection	Object	SQL – Select, From, Where	Encryption	Primary / Secondary storage	Subnet mask
	Storage	If	Master slide	Testing – iterative & final	Defragmentation	Optical	DHCP / NAT
	CPU	Indentation	View	Test data–normal, boundary, invalid	Data Compression	Magnetic	WebSocket
	Random Access Memory (RAM)	Else	Organise	Syntax and Logic errors		Solid State	CRUD / REST / JSON / XML

	Spreadsheets	Computational Thinking and Hardware	Office Skills – Word processing	Languages		Wireless networks, Project	
Summer 2	Data	Binary	Word processor	High & Low level languages		Entity	
	Cell	Decomposition	Paragraph	Translators		Attribute	
	Row	Variable	Section	Compilers		Tuple	
	Column	Testing	Break	Interpreter		Relationship	
	Formula	CPU	Format	IDE		Normal form	
	Function	Storage devices	Layout	Editor		Key	
	Relative / Absolute	Input devices	Mail merge	Diagnostics		NEA	
	Graph / chart	Output devices	Review	Run-time environment		Deadline	