Ursuline College Curriculum Implementation: **Computer Science & IT Year 7**

Curriculum Intent	*'I have come so that they may have life and have it to the full.'* John 10: Chapter 10. V 10 Inspired by the Gospel values of love, joy, and compassion, we nurture and develop the whole child so that each unique individual grows in confidence and has the knowledge and skills to enable them to achieve their true potential. Our Computing curriculum aims to develop students' computational thinking and problem-solving abilities, equipping them with the skills needed for future success. Aligned with our core values of respecting, helping, and bearing with each other, we strive to foster resilience and creativity, enabling students to become critical thinkers who can tackle any challenge. Our broad and balanced curriculum prepares students for a technological world, teaching them to use various Computing and coding software. Through exposure to key computational concepts, students gain the knowledge, skills, and understanding to stay safe online, comprehend how computers work, and confidently use technology. This curriculum nurtures resilient learners capable of solving problems and learning from their mistakes. **Respecting each other, helping each other and bearing with each other: Getting better never stops.**	
Year 7 Intent	The intent for Year 7 is to give students the core digital skills, e-safety awareness and computing knowledge they need to work safely and efficiently throughout their secondary education and beyond. Students begin with school systems and modernised e-safety, then explore how computers work through hardware and binary, and are introduced to the world of computing careers. They go on to build strong programming foundations through computational thinking, MicroPython on the micro:bit and Python, before finishing the year with user interface and web development. This gives them a secure footing for the next phase of their computing education, including programming and understanding how computers work.	
	Cycle 1	
	Term 1 2.9.26 – 23.10.26 7 weeks	Term 2 2.11.26 – 18.12.26 7 weeks
Focus & Skills	School systems: Logging in, Teams, OneDrive & passwords E-safety (modernised): Online bullying & reporting Scams, phishing & social engineering AI, deepfakes & misinformation Hardware: What is a computer, inside a computer & the CPU	Careers in Computing: Introduction to computing careers Tech job families: software, cyber, data, creative tech Skills employers look for My skills audit & computing pathway A day in the life of a software engineer Showcase: my computing career pathway
Links to CST	**Preferential option for the poor and vulnerable:**	

	Making school systems and technology easy to access reflects the preferential treatment of the poor, ensuring students with less access to computing at home are not left behind. Opening students' eyes to computing careers reflects the preferential treatment of the poor, widening opportunity for those who might not otherwise see these pathways as open to them.	
Explicit Skill Content	How to log in and use Teams, OneDrive, and Office 365, and set secure passwords How to recognise and report online bullying How to spot scams, phishing, and social engineering How AI deepfakes and misinformation work and how to stay critical Defining what a computer is; input and output devices Identifying the main internal components How the CPU works – the basics of the Fetch-Decode-Execute cycle	What careers exist in computing and what people in tech do Understanding the main tech job families Identifying the skills employers value Completing a personal skills audit Researching a day in the life of a software engineer Presenting a personal computing pathway plan
Homework	• Keeping accounts safe – secure password paragraph • Design a reward postcard for Computing • Literacy keywords & activity sheets	• Research one tech job: what do they do day-to-day? • Complete skills audit started in lesson • Literacy keywords
Assessment	Showcase presentation and MS Forms assessment at end of cycle Regular use of R&R to assess prior learning	
Disciplinary Literacy Vocab	Login Password Account Digital Footprint Phishing Scam Social Engineering Deepfake Misinformation Input Output Processor Component CPU	Career Pathway Software Cyber Data Creative tech Apprenticeship Skills Audit Employer Portfolio Engineer
Texts/ Resources	Arbor Office 365 Teams	Office 365 Careers case studies / visiting speakers

KS4 Links	These essentials give students the correct start for all of their subjects. E-safety links to KS4 through cyber security and social engineering, while hardware and the FDE cycle underpin GCSE Computer Science and BTEC DIT.	Careers education gives students the context to make informed KS4 option choices, linking directly to GCSE Computer Science and BTEC / Cambridge Nationals pathways.
	Cycle 2	
	Term 3 4.1.27 – 12.2.27 6 weeks	Term 4 23.2.27 – 25.3.27 5 weeks
Focus & Skills	Hardware: The CPU (deep dive) Binary & data representation: Converting binary to and from denary ASCII Computational thinking: Introduction to algorithms Micro:bits: MicroPython from day one	Micro:bits (MicroPython): Variables & buttons – Magic 8 Ball Selection & loops – reaction game Mini-project – name badge or pong Python – Turing Lab: Farmbot programming Transition towards Thonny
Links to CST	**Preferential option for the poor and vulnerable:** Building strong digital foundations for every student embodies the preferential treatment of the poor, giving those with fewer resources the knowledge to thrive in a technological world. Programming empowers students to create solutions to real problems, including tools that can serve and uplift the poor and marginalised.	
Explicit Skill Content	How the CPU works in more depth Converting binary numbers to denary Converting denary numbers to binary Representing characters using ASCII Decomposing problems and writing simple algorithms Writing first MicroPython programs on the micro:bit	Using variables and button inputs Using selection and loops to build interactive programs Planning and building a mini project Writing Python in Turing Lab (Farmbot) using sequence, selection and iteration
Homework	• Binary practice worksheet • Predict-the-output: 5 MicroPython snippets (PRIMM) • Literacy keywords	• Game design worksheet • Complete Turing Lab • Literacy keywords
Assessment	Vocab test and Seneca learning R&R's to assess prior learning	Assessment of micro:bit mini project R&R's to assess prior learning
Disciplinary Literacy Vocab	• Processor • Fetch • Decode • Execute	• Variable • Selection • Loop • Iteration

	• Binary • Denary • ASCII • Bit • Byte • Algorithm • Decomposition • Sequence • MicroPython	• Input • Output • Button • Sensor • Sequence • Function • Algorithm • Debug
Texts/ Resources	Seneca learning Office 365 Micro:bit Classroom & micro:bits	Micro:bit Classroom & micro:bits Turing Lab Office 365
KS4 Links	Binary, ASCII and the FDE cycle are essential for foundation marks at GCSE. Algorithms and MicroPython lay the groundwork for GCSE CS programming and BTEC DIT.	Physical computing and Python build the programming foundations for GCSE CS (algorithms, writing and tracing code) and BTEC DIT Component 3.
	Cycle 3	
	Term 5 12.4.27 – 28.5.27 7 weeks	Term 6 7.6.27 – 16.7.27 6 weeks
Focus & Skills	Python – Thonny: Syntax & debugging Selection Iterative loops (FOR) Conditional loops (WHILE) & mini challenges End of year assessment	User Interfaces: What makes a good UI? Design & showcase Web Development: What is HTML? History of the web Building basic pages Design, style & accessibility
Links to CST	**Preferential option for the poor and vulnerable:** Learning to program equips students to build technology that can benefit disadvantaged communities, putting the needs of the poor first. Designing accessible interfaces and websites reflects the preferential treatment of the poor, ensuring technology works for those with disabilities, low incomes or limited access.	
Explicit Skill Content	Writing and debugging Python in Thonny Using selection to control program flow Using FOR loops for iteration Using WHILE loops for conditional repetition Combining concepts in mini challenges	What makes a good, accessible user interface Designing a simple UI Understanding HTML and the history of the web Building basic web pages Styling pages and designing for accessibility
Homework	• Python PRIMM quiz • Literacy keywords & Seneca	• What UI would you like to create? Design it at home • Plan your website – purpose & structure

		• Literacy keywords
Assessment	End of year MS Forms / programming assessment R&R's to assess prior learning	Final outcome from the UI / web project assessed R&R's to assess prior learning
Disciplinary Literacy Vocab	• Syntax • Debug • Variable • Selection • Iteration • For • While • Condition • Function • Algorithm • Input • Output	• Interface • Accessibility • Usability • Navigation • HTML • Tag • Element • Web page • Browser • Style • Layout
Texts/ Resources	Thonny Office 365	PowerPoint HTML editor Office 365
KS4 Links	Python in Thonny covers variables, loops, selection and inputs, directly supporting GCSE CS algorithms and code tracing, and BTEC DIT Component 3.	UI design and HTML link to BTEC DIT Component 1 (user interface design) and support GCSE CS understanding of how software and the web work.