

Sackville School Computer Science Curriculum - Year 12 Computing



TERM	WHAT? (Is delivered?)	WHY? (Is this important?)	WHY NOW? (Why is this taught now?)	IMPACT? (What is the impact at the end of this half term?)	ASSESSMENT
Aut 1	Software Types and Categories Software Development Models	<ul style="list-style-type: none"> Exploring the different software categorisation and their implementation/use in different scenarios to form a foundation for the courses learning. Investigating different software development models to equip us with the knowledge needed to start software development. 	<ul style="list-style-type: none"> The process of software development is such that students need to understand the term application software and the different methodologies used to develop software before they begin their development phases. 	<ul style="list-style-type: none"> Students will be able to articulate the different software types and discuss the different models used in early software development. 	Knowledge Checkers Scenario Focused Assessment
Aut 2	Roles and Industry Standards Implementation and Maintenance Models	<ul style="list-style-type: none"> Exploring the different roles in the application development industry and the different standards that are de facto to this sector. Investigating different software implementation models to equip us with the knowledge needed to start software development. 	<ul style="list-style-type: none"> The process of software development requires students to understand the different models of implementation and their benefits and uses. There are different types of industry roles and students need to know these before they begin their projects 	<ul style="list-style-type: none"> Students will be able to apply for a role and understand a role requirements Students will be able to discuss the differing implementation models and their 'fitness' to be used 	Knowledge Checkers Scenario Focused Assessment
Spr 1	UX and UI Storage	<ul style="list-style-type: none"> Developing the User Experience (UX) and User Interface (UI), which covers defining screen layouts, navigation flows, and visual component designs. Determining the necessary data structures and persistence strategy (e.g., cloud database or local storage) required to manage all application data. 	<ul style="list-style-type: none"> A robust design phase minimizes costly errors, delays, and scope creep during the subsequent development and testing stages. Effective design ensures the final application is intuitive, accessible, and meets the user's intended goals, driving adoption and satisfaction. 	<ul style="list-style-type: none"> You must establish non-functional requirements (e.g., security, performance) and data structures before beginning any coding or prototyping in the next unit. This unit directly informs the creation of the Statement of Intent (SOI) and initial project brief, which need sign-off before detailed work can commence. 	Knowledge Checkers Application Checker
Spr 2	Network Security Legislation	<ul style="list-style-type: none"> Network Security: Defining security models, access control protocols (e.g., strong authentication and authorisation), and necessary encryption methods to protect data during transmission and storage. Legislation: Researching and documenting the application's mandatory compliance with relevant data protection acts (e.g., GDPR) and ethical guidelines regarding data 	<ul style="list-style-type: none"> Cyber threats evolve rapidly, making security a fundamental, non-negotiable requirement that must be addressed from the initial design phase of any application. It is critical for designing robust, real-world applications (a core AAQ focus) that can withstand common security vulnerabilities and attacks. 	<ul style="list-style-type: none"> Learners can correctly identify potential security vulnerabilities in their application design and specify appropriate protection measures (e.g., input validation, secure session management). Students can justify their security architecture choices in documentation based on assessed risks, aligning with industry best practices for secure application development. 	Knowledge Checkers Mock Exam

		use and storage.			
Sum 1	Protocols Project Delivery	<ul style="list-style-type: none"> To ensure universal compatibility, allowing different hardware and software systems to communicate reliably and exchange data without errors. They define error-checking and routing mechanisms essential for maintaining stable performance and ensuring data integrity in networked applications. 	<ul style="list-style-type: none"> Modern application design is inherently networked (client-server models); the choice of protocol is a fundamental design decision that dictates data format and security requirements. It is critical for configuring API interactions, ensuring efficient data transfer, and securing the data link layer of the application. 	<ul style="list-style-type: none"> Learners can select and justify the appropriate protocol stack (e.g., HTTPS over TCP) required for their application's specific communication tasks. Students can define the structure of data packets/messages (e.g., JSON payloads) that adhere to the chosen protocol standards for reliable transmission. 	Model exam questions and assignment questions
Sum 2	NEA Delivery Time				

Links to L4L Curriculum and Gatsby Benchmarks:
Job roles and professional bodies in IT -