

Term	Week	Focus	Summary	Learning Outcomes	Learning skills
Term 2.1	1	Mocks			
	2				
	3				
	4	Computational Thinking and Advanced data representation	Backus-Naur Form, Big O Notation The binary number system	To explore, describe and evident computation thinking techniques Ability to extend the range and accuracy of binary numbers available within a byte including floating point numbers	Analysing
	5	Networking	The Internet, Internet security	To understand the workings of the internet including URL, ports and NAT's	Research
	6	Functional Programming	Writing functional programs	Ability to use higher order functions including map, filter and fold	Problem Solving
Term 2.2	1	Non-Exam Assessment – Scope	Investigation into the project. (Interview, Quiz, Observation. A description of the problem A List of SMART Objectives.	Identify key aspects of the problem for further analysis.	Problem Solving Research
	2	Non-Exam Assessment – Analysis	Flowchart ERD Consider Possible Solutions	Develop Entity relationship diagrams Evaluation and compare solutions Formulate a comprehensive understanding of the problem	Analysing

	3	Non-Exam Assessment – Design and development	ERD DFD Data Dictionaries IPSO Chart OOP Class design User Interface Design Top-Down Diagram	Utilise an IPSO chart for planning and organising. Design object-orienteed Programming diagram Create a top-down diagram for system representation	Creativity Critical Thinking
	4	Non-Exam Assessment - Development Skelton Code Preparations	Coding the solution, implementing design principles, and adhering to coding conventions. It emphasizes systematic testing, debugging, and optimizing for efficiency.	Implement design principles and coding conventions Emphasise systematic testing, debugging and optimisation. Implement through testing	Creativity Critical Thinking Problem Solving
	5	Non-Exam Assessment – Testing Skelton Code Preparations	Coding the solution, implementing design principles, and adhering to coding conventions. It emphasizes systematic testing, debugging, and optimizing for efficiency.	Implement design principles and coding conventions Emphasise systematic testing, debugging and optimisation. Implement through testing	Creativity Critical Thinking Problem Solving