Learning Programme

Fundamentals of data structures – A Level

Topic/Content	Objectives/Skills	Homework	Assessment	Success Criteria	Stretch & Challenge (Thirst for Learning)
Data structures and abstract data types			Past paper 1 exams	Based on past paper grade boundaries	Creation of software to be able to hash and salt a password to make its encryption almost uncrackable

Abstract data types/data structures	• Be familiar with the concept and uses of a:
	  ○ Queue
	  ○ Stack
	  ○ Graph
	  ○ Tree
	  ○ hash table
	  ○ dictionary
	  ○ vector
	• Be able to distinguish between static and dynamic structures and compare their uses, as well as explaining the advantages and disadvantages of each.
	• Describe the creation and maintenance of data within:
	  ○ queues (linear, circular, priority)
	  ○ stacks
	  ○ hash tables.
		Topic Tests
		Past paper questions			

Queues	• Be able to describe and apply the following to linear queues, circular queues and priority queues: ○ add an item				

	○ remove an item ○ test for an empty queue ○ test for a full queue				
Stacks	• Be able to describe and apply the following operations: ○ Push ○ Pop ○ peek or top ○ test for empty stack ○ test for stack full				
Graphs	• Be aware of a graph as a data structure used to represent more complex relationships. • Be familiar with typical uses for graphs. • Be able to explain the terms: ○ graph ○ weighted graph ○ vertex/node ○ edge/arc ○ undirected graph ○ directed graph. • Know how an adjacency matrix and an adjacency list may be used to represent a graph. • Be able to compare the use of adjacency matrices and adjacency lists.				
Trees (including binary trees)	• Know that a tree is a connected, undirected graph with no cycles	Creation of a simple tree Topic tests			

	• Know that a rooted tree is a tree in which one vertex has been designated as the root. A rooted tree has parent-child relationships between nodes. The root is the only node with no parent and all other nodes are descendants of the root. • Know that a binary tree is a rooted tree in which each node has at most two children. • Be familiar with typical uses for rooted trees.				
Hash tables	• Be familiar with the concept of a hash table and its uses. • Be able to apply simple hashing algorithms. • Know what is meant by a collision and how collisions are handled using rehashing.	Creation of a hash program			
Dictionaries	• Be familiar with the concept of a dictionary. • Be familiar with simple applications of dictionaries, for example information retrieval, and have experience of using a dictionary data structure in a programming language.				
Vectors	• Be familiar with the concept of a vector and the following				

	notations for specifying a vector: ○ [2.0, 3.14159, -1.0, 2.718281828] ○ 4-vector over $\mathbb{R}$ written as $\mathbb{R}4$ ○ function interpretation ▪ $0 \mapsto 2.0$ ▪ $1 \mapsto 3.14159$ ▪ $2 \mapsto -1.0$ ▪ $3 \mapsto 2.718281828$ ▪ $\mapsto$ means maps to ○ That all the entries must be drawn from the same field, eg $\mathbb{R}$. • Dictionary representation of a vector. • List representation of a vector. • 1-D array representation of a vector. • Visualising a vector as an arrow. • Vector addition and scalar-vector multiplication. • Convex combination of two vectors, u and v. • Dot or scalar product of two vectors. • Applications of dot product.				