



# Section 10.4 - Introduction to Abstract Data Types (ADT)

## Layer 6: High-Order Language



### Syllabus Content Section 10: Data Types and Structure

 S10.4.1 Show understanding that an ADT is a collection of data and a set of operations on those data 

---

**Abstract Data Type:** a collection of data with associated operations

- create a new instance of the data structure
- find an element in the data structure
- insert a new element into the data structure
- delete an element from the data structure
- access all elements stored in the data structure in a systematic manner.

 S10.4.2 Show understanding that a stack, queue and linked list are examples of ADTs 

- Describe the key features of a stack, queue and linked list and justify their use for a given situation

---

--	--

Stacks	<div> <div> 7 6 5 4 3 2 1 0 </div> <div> <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div> </div> <div> ← BaseOfStackPointer ← TopOfStackPointer </div> </div> <div> <div> 7 6 5 4 3 2 1 0 </div> <div> <div></div><div></div><div></div><div></div><div>D</div><div>C</div><div>B</div><div>A</div> </div> <div> ← TopOfStackPointer ← BaseOfStackPointer </div> </div>
Queues	<div> <div> 0 1 2 3 4 5 6 7 </div> <div> <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div> </div> <div> ← FrontOfQueuePointer ← EndOfQueuePointer </div> </div> <div> <div> 0 1 2 3 4 5 6 7 </div> <div> <div>A</div><div>B</div><div>C</div><div>D</div><div>E</div><div></div><div></div><div></div> </div> <div> ← FrontOfQueuePointer ← EndOfQueuePointer </div> </div>
Circular Queues	<div> <div> 0 1 2 3 4 5 6 7 </div> <div> <div>I</div><div>J</div><div>K</div><div></div><div></div><div>F</div><div>G</div><div>H</div> </div> <div> ← EndOfQueuePointer ← FrontOfQueuePointer </div> </div>

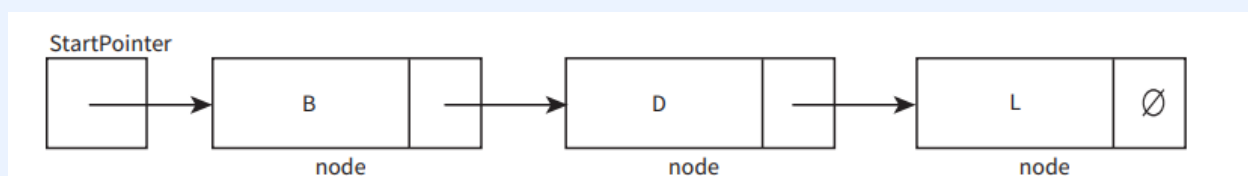
## Linked lists

**Node**: an element of a list

**Pointer**: a variable that stores the address of the node it points to

**Null pointer**: a pointer that does not point at anything

**Start pointer**: a variable that stores the address of the first element of a linked list



#### S10.4.3 Use a stack, queue and linked list to store data

- Candidates will not be required to write pseudocode for these structures, but they should be able to add, edit and delete data from these structures
- 

#### S10.4.4 Describe how a queue, stack and linked list can be implemented using arrays

---