



4.2.1 Data Structure & ADTs

Data structures

A data structure is a way of organising, storing, and managing data so that it can be accessed and modified efficiently.

Static data structures

- Have a fixed size (storage size is determined at compile-time).
- Data is stored in consecutive memory locations.
- Can waste storage space if the number of data items stored is small relative to the memory reserved

Dynamic data structures

- Can change size (storage size can grow/shrink during run-time).
- Data is not stored in consecutive memory locations.
- Require points to be stored with the data to point to the next item location in memory.
- Only take up the amount of storage space required for the data currently in use.

Array

- Is a static data structure.
- Can only hold one data type.
- Is ordered and uses indexes for access.
- Is mutable.

List

- Is a dynamic data structure.
- Can hold multiple data types.
- Is ordered and uses indexes for access.
- Is mutable.

Tuple

- Is a static data structure.
- Can hold multiple data types.
- Is ordered and uses indexes for access.
- Is immutable.





Abstract data types

An ADT is a logical description of a data structure that defines what operations can be performed on the data, without specifying how those operations are implemented.

Example:

A stack defines operations push and pop but could be implemented in a number of ways, such as using an array or a linked list.

A **string** is a sequence of characters.

String-handling operations allow programs to manipulate and process text.

