7. Abstract Data Type (ADT)

ADT is a model for DS that defines the behaviour of the data types without specifying its implementation. It focuses on what operations a data type can perform rather than how it performs those operations.

Key Features of ADT:

- Encapsulation of Data and Operations - ADTs hide the internal representation of data. Users interact with the data only through a set of well-defined operations, which ensures that the data is safe from unintended interference.
- Separation of Interface and Implementation - The ADT defines an interface (a set of operations), but how those operations are implemented is not exposed. This allows for flexibility in implementation.
- Reusability - Since the implementation is hidden, ADTs can be easily reused or changed without affecting the rest of the code that relies on them.

Example :

Array has many operations :

Add
Remove
Fetch

Here ADT will show only behaviour, but how these operations were performed ADT will not show implementations.

Stack (LIFO) :

Push
Pop
Peek -> Return the item at the top of stack, without removing it.
isEmpty

Here ADT will show only how this DS i.e. stack behaves, but implementation will be hidden.