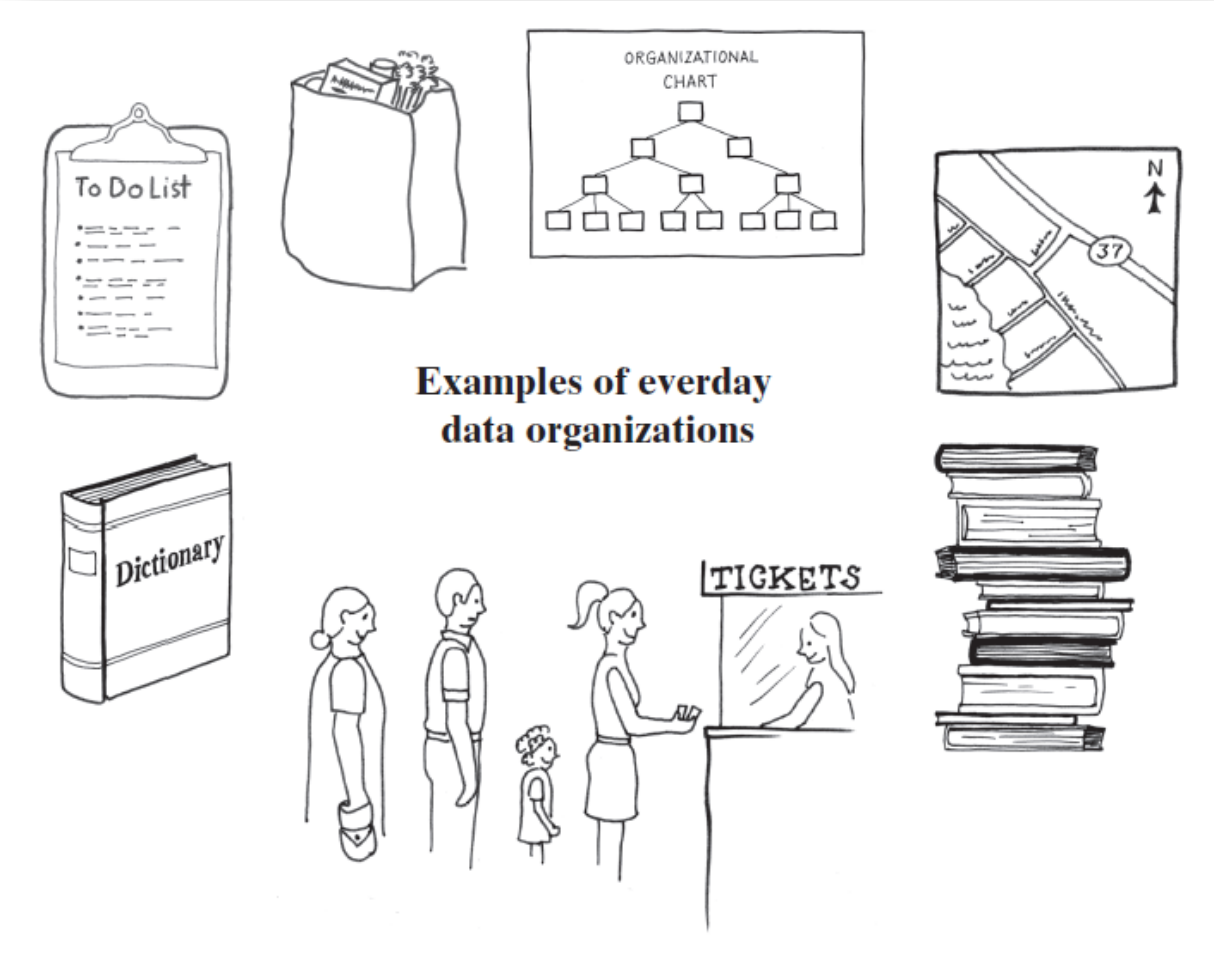
Abstract Data Type (ADT)

WIA1002/ WIB1002 :

Data structure

Data Organization



© 2016 Pearson Education, Ltd. All rights reserved.

Data Organization in Life

- Standing in a line
- Stack of books
- To-Do list
- Dictionary
- Folders, directories on your computer
- Road map

© 2016 Pearson Education, Ltd. All rights reserved.

Computer Data Organization

- In ways similar to the real-world examples (list, stack, dictionary, etc)
- These ways of organizing data are represented by abstract data types (ADTs)
- **Abstract data type (ADT)**
  - is a specification that describes a data set and the operations on that data that is defined conceptually and independently of any programming language.
  - specifies what data is stored and what the operations on the data do.
  - does not indicate how to store the data or how to implement the operations
- **Data structure** is an implementation of an ADT within a programming language.

• concept that is free from any programming lang.
• A process to organize data

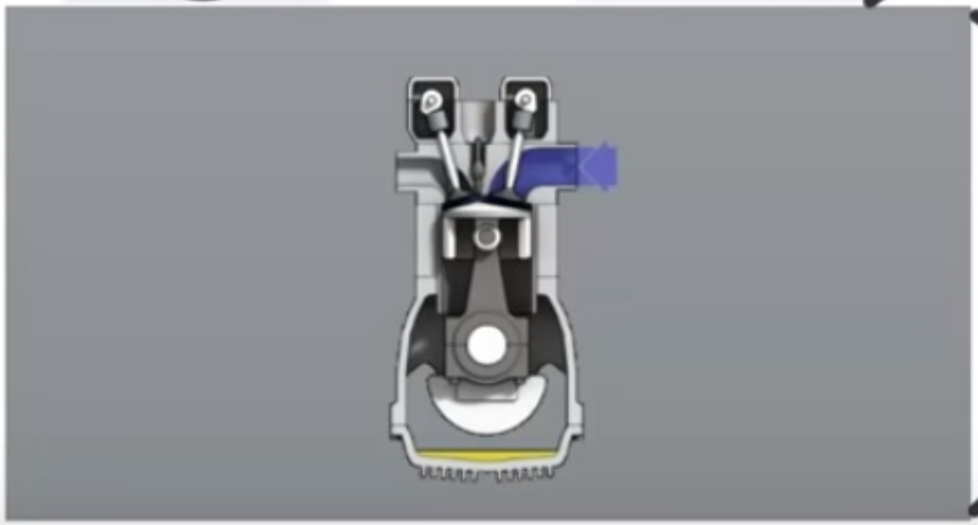
© 2016 Pearson Education, Ltd. All rights reserved.

Computer Data Organization

- A **collection** is a general term for an ADT that contains a group of objects.
- A **container** is a class that implements a collection. Some people use the terms “container” and “collection” interchangeably

© 2016 Pearson Education, Ltd. All rights reserved.

Interface



Implementation

Example

- Cinema Reservation System

(https://www.youtube.com/watch?v=HcxqzYsiJ3k)

  - Data?
    - Seats
    - Seats reserved or available
  - Operations ?
    - Determine availability of seats
    - Reserve a seat
    - Cancel a reservation
    - Find a block of available seats
  - Implementation?
    - How we write the program

What is ADT?

- A type in which
  - **What** the operations of the type are is **public**
    - That is, public to the client
  - **How** the type is implemented is **private**
    - That is, private to the client
- A type defines data items and associated operations, but not implementation (i.e., interface)
  Abstract = irrelevant details are ignored
  - The ADT is abstract because how the ADT is implemented is ignored

An abstract data type (ADT) is a model of a data structure that specifies:

- the characteristics of the collection of data
- the operations that can be performed on the collection

It's abstract because it doesn't specify how the ADT will be implemented.

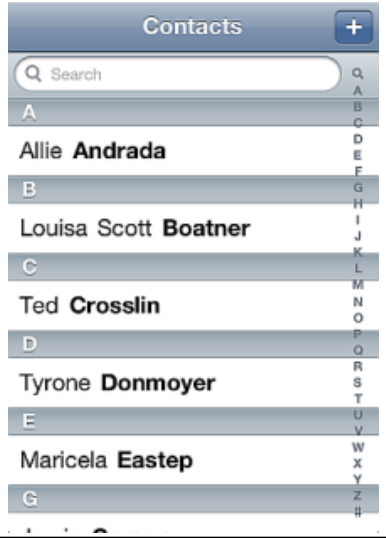
A given ADT can have multiple implementations.

ADT Examples

The following are unrelated items :

- A deck of playing cards
- A set of index cards containing birthday information

- Telephone numbers stored in your cellular phone



What do they share in common?

- Each one is a collection of elements.
- There is a first element.
- There is a second element, third element, and so on.
- There is a last element.
- Given an element other than the last element, there is a ‘‘next’’ element.
- Given an element other than the first element, there is a ‘‘previous’’ element.
- An element can be removed from the collection.
- An element can be added to the collection.
- A specified element can be located in the collection by systematically going through the collection.

Advantage of ADTs

- Shares data and operations.
- Information hiding. That is, ADT hides the implementation details of the operations and the data from the users of the ADT. Users can use the operations of an ADT without knowing how the operation is implemented

Reference

- *Introduction Chapter and Chapter 1, Data Structures and Abstractions with Java, 4e,* Frank Carrano
- https://www.youtube.com/watch?v=HcxqzYsiJ3k
- http://www.radford.edu/~nokie/classes/320/Abstract_Data_Types.html