

Abstract Data Types, Data Structures

Data Structures and Algorithms

Concepts

A data type is a set of values and a set of operations on those values.

Primitive Data types are defined for a programming language; Compiler for the Programming language evolves based on machine representation.. int char double etc.

Composite data types are built on top of primitive data types and are understood by the compiler of the language based on its grammar/syntax.

Concepts ..

Primitive Data types

int, char, boolean, float, double, byte, short (Java)

int, char, bool, float, double, long, unsigned long, long long, double, long double etc (C / C++)

complex (Python)

Composite Data types in general

Array, structure, class, union, (C/C++/Java)

List, tuple (Python)

Concepts ..

Most times we want to represent data at a higher level of abstraction (instead of using primitive/composite data types)

An abstract data type (ADT) is a data type you *design* and
define operations on data.

define data organization and representation.

An abstract data type (ADT) is a data type you *use* to

Implement API or Applications

Develop/Enhance other ADT, eg. implement a Priority Q based on Q.

Concepts..

When a programmer works on a problem, he/she thinks abstractly about data and its properties, designs operations on the data.

Data storage, how the data is accessed and the implementations of the operations are not considered at this stage in the design of the solution.

ADT is a specification for a group of values and operations on those values.

Concepts ..

When using an ADT, we focus on the operations specified in the API and pay no attention to the data representation; when implementing an ADT, we focus on the data, then implement operations on that data.

– Robert Segdewick

Users/Clients of API are provided with ADT implementation in the form of classes, interfaces, structures, libraries etc.

Concepts ..

Role of Programmers :

Clients of ADT – Clients are those entities that are interested in operations the ADT provides. Client do not care about how the ADT is implemented.

ADT developer – Understand Client problem, Design ADT and Hide ADT representation from client. Provide operations required for the problem.

Learning :

Using a ADT versus Implementing a ADT. (understand your learning role)

Get familiar with Data Structures/API available in the language/Ecosystem.
(ie you get to use Data Structures and become glib in them)

Concepts ..

ADT's can be classified as

Value Oriented ADT, ie ADT puts emphasis on the value of the items it holds to define operations eg. sorted list

Position Oriented ADT ie ADT puts emphasis on the position of the item that it holds to help define operations, eg : stack, queue, a list

Concepts ..

Data Structures are used to implement ADTs. Data Structures are an implementation of ADT within a programming language.

- List, Singly linked, doubly linked, linear, circular
- Set, HashSet
- Stack, BackStack
- Queue, Priority Q, Deque
- Tree, Binary, BST, Heap, Red-Black
- Associative Array, Map, HashMap, Hash table, Dictionary
- Graph, Directed, Undirected, acyclic, cyclic etc.

