Abstract Data Types (ADT)

Atul Gupta

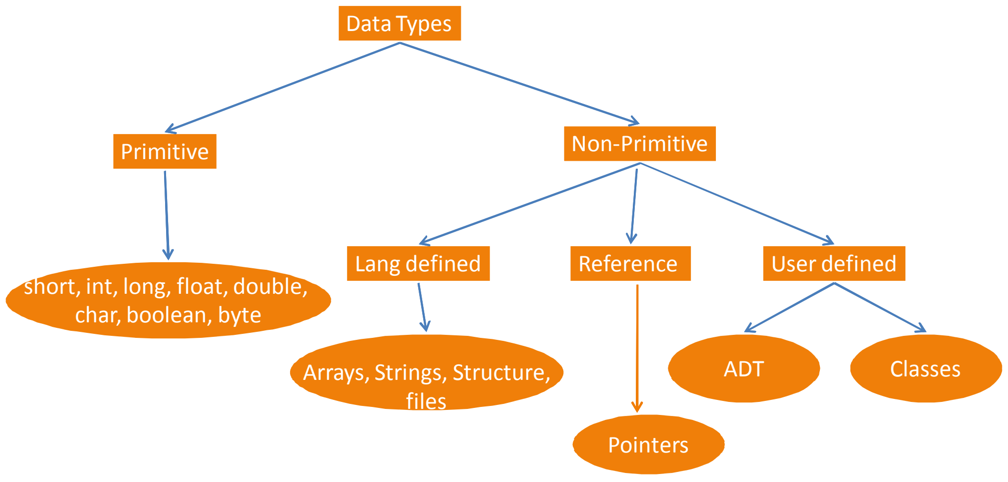
ES103: Course Summary

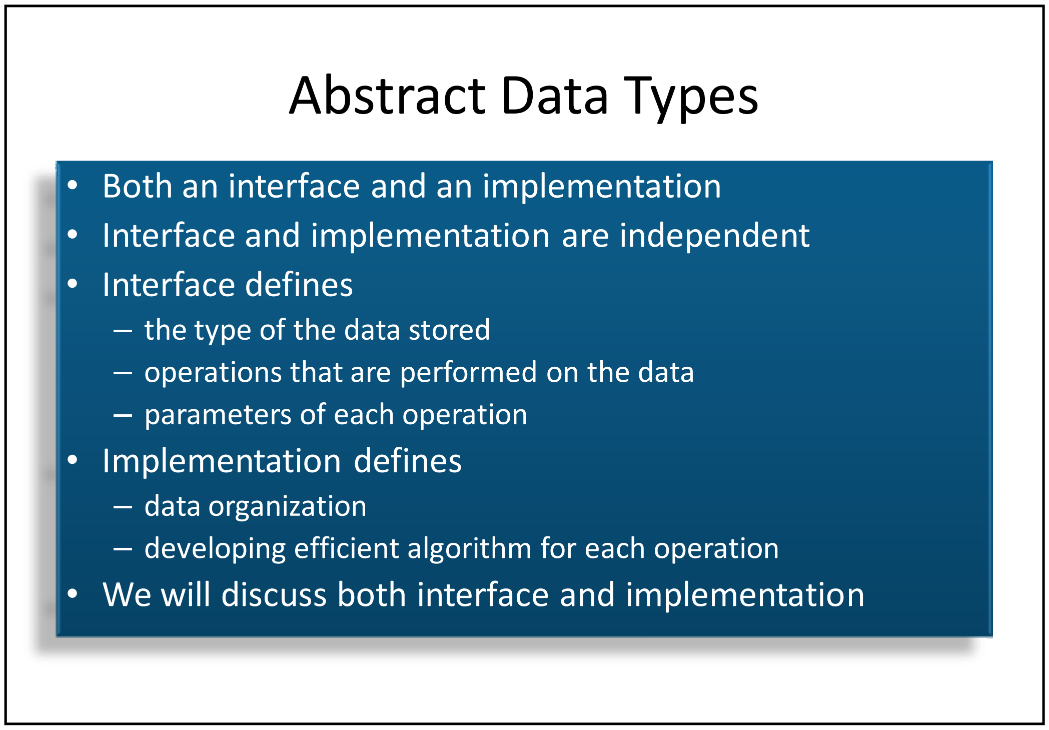
- Estimating algorithmic complexities
- Tools like Recursion, Iteration
- ADT
  - Linear data organizations like Arrays, Stacks, Queues, Linked lists
  - Non-linear data organizations like hashing, trees (and its many variances), graphs (and many variances)
  - Examples
- Standard Problems: Searching and Sorting
- Standard Problem Solving Approaches

Data Types

- Set of values
- Operations that can be performed on those values
  - Ex: **short int**
    - can take values (-32768 to 32767)
    - Operations are +, -, ×, /

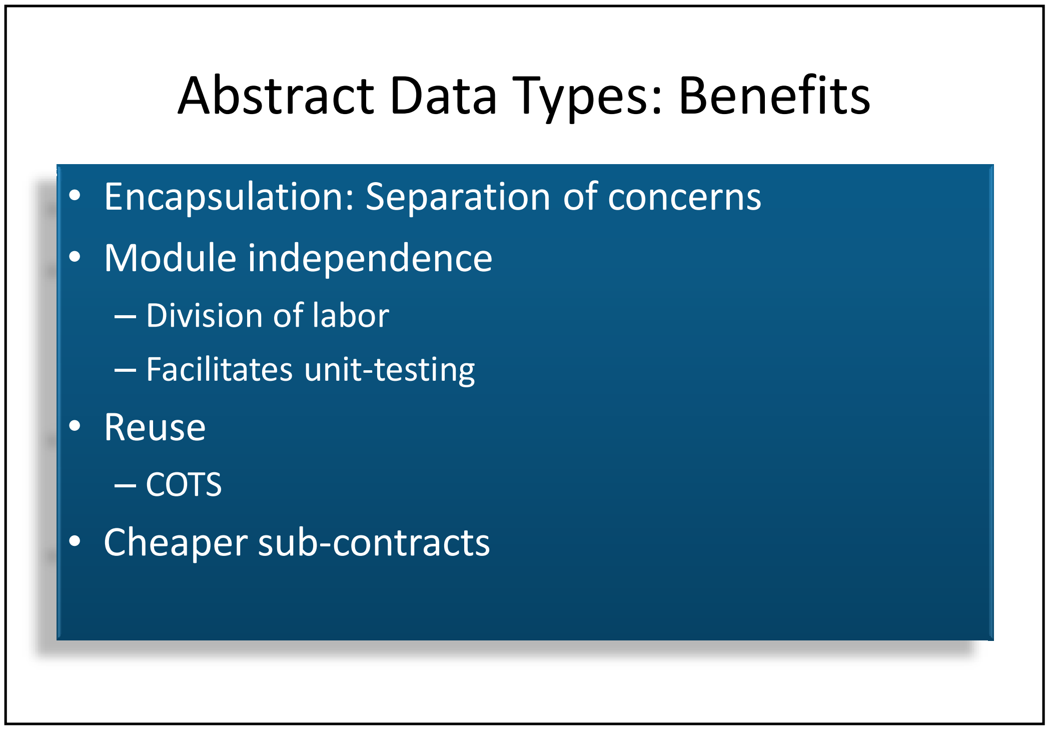
Data Types


Data Types
Primitive
Non-Primitive
short, int, long, float, double, char, boolean, byte
Lang defined
Reference
User defined
Arrays, Strings, Structure, files
Pointers
ADT
Classes



Abstract Data Types

- Both an interface and an implementation
- Interface and implementation are independent
- Interface defines
  - the type of the data stored
  - operations that are performed on the data
  - parameters of each operation
- Implementation defines
  - data organization
  - developing efficient algorithm for each operation
- We will discuss both interface and implementation



Abstract Data Types: Benefits

- Encapsulation: Separation of concerns
- Module independence
  - Division of labor
  - Facilitates unit-testing
- Reuse
  - COTS
- Cheaper sub-contracts

Basic Operations of a Collection ADT S

- insert(S, x)
- delete(S, x)
- search(S, x)
- findMin(S)
- findMax(S)
- findSuccessor(S, x)
- findPredecessor(S, x)

Collections ADTs

- Linear
- Non-linear

Linear ADTs

- Restricted Lists
  - Stack
  - Queue
    - Circular queue
    - Priority queue
- General Lists
  - Arrays
  - Linked list
  - Circular list
  - Doubly linked list

Non-linear ADTs

- Trees
  - Binary Trees and Types
  - Binary Search Trees and Variants
  - Threaded Binary Trees
  - Heaps
- Graphs
  - Undirected
  - Directed
- Hash Tables

ADTs: Summary

- Standard data collection organizations (Data Structures) with desired operations
- Described by an interface
- Many implementations are possible
- Facilitate reuse and easy extensibility
- Design issues are Time and Space Complexity