

# Set ADT

# Set

- A collection of elements with no strict ordering
  - Example:
- Are duplicates allowed in a set?

# Specification

- Elements: Set elements can be of any type, but we will assume SetElement
- Structure: Any mechanism for tracking the items

# Operations on Sets

- Think back to Discrete Mathematics
- What operations can be performed on sets?

# Set Operations

- function create (s: Set, isCreated: boolean)  
results: if s cannot be created, isCreated is false; otherwise, isCreated is true, the set is created and is empty
- function terminate (s: Set)  
results: Set s no longer exists

# Set Operations

- function isEmpty (s: Set)  
results: returns true if the Set is empty;  
otherwise, false is returned
- function contains (s: Set, e: SetElement, b:  
boolean)  
results: set b to true if e is in the Set;  
otherwise set b to false

# Set Operations

- function insert (s: Set, e: SetElement)  
requires: contains (s, e) is false  
results: element e is added to the set
- function remove (s: Set, e: SetElement)  
requires:  
results:

# Set Operations

- function union (s1: Set, s2: Set; result: Set)  
results:
- function: intersection (s1: Set, s2: Set; result: Set)  
results:

# Set ADT

- Can we use a Linked List to build this data structure?
- What other operations would be useful?
- Can we print every element of the set to the screen?

# Bag

- A bag is similar to a Set but duplicates are allowed in the Bag.
- Also called a multiset.