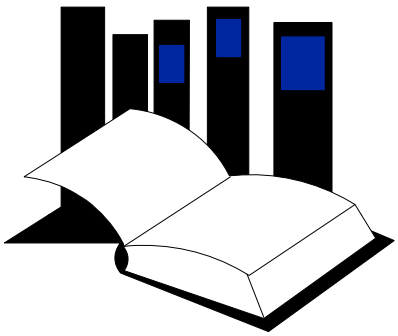
Container ADTs



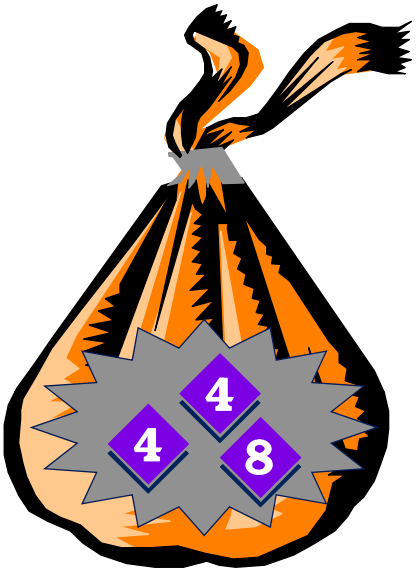
- A ***container ADT*** is a data type that is capable of holding a ***collection of items***.
- In C++, container ADTs can be implemented as a **`class`**, along with member functions to *add* items, *remove* items, *examine* items, *etc*.

Bags

- For an example, think about a bag.

Bags

- For an example, think about a bag.
- Inside the bag are some numbers.



Initial State of a Bag

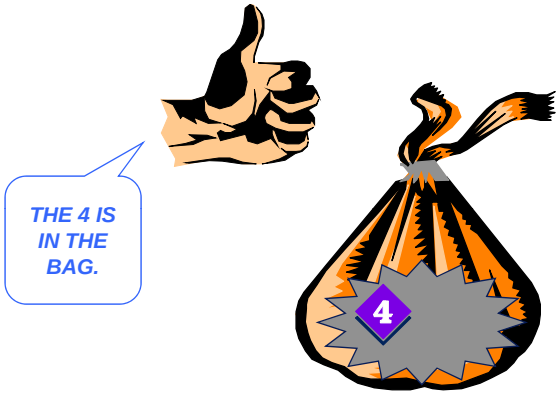
- When you first begin to use a bag, the bag will be empty.
- We count on this to be the ***initial state*** of any bag that we use.

Inserting Numbers into a Bag

- Numbers may be inserted into a bag.

Inserting Numbers into a Bag

- Numbers may be inserted into a bag.



Inserting Numbers into a Bag

- Numbers may be inserted into a bag.
- The bag can hold many numbers.

Inserting Numbers into a Bag

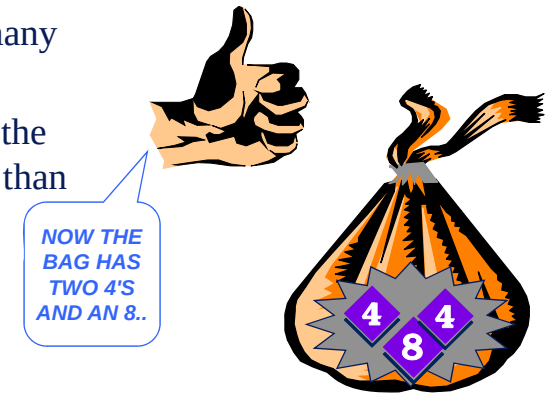
- Numbers may be inserted into a bag.
- The bag can hold many numbers.

Inserting Numbers into a Bag

- Numbers may be inserted into a bag.
- The bag can hold many numbers.
- We can even insert the same number more than once.

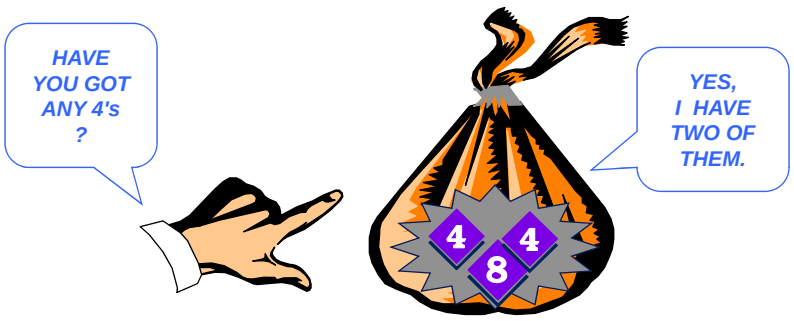
Inserting Numbers into a Bag

- Numbers may be inserted into a bag.
- The bag can hold many numbers.
- We can even insert the same number more than once.



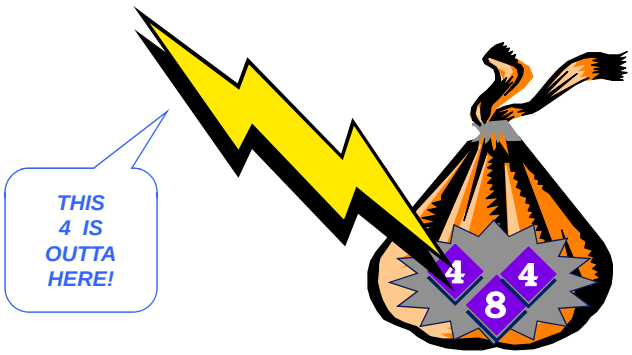
Examining a Bag

- We may ask about the contents of the bag.



Removing a Number from a Bag

- We may remove a number from a bag.

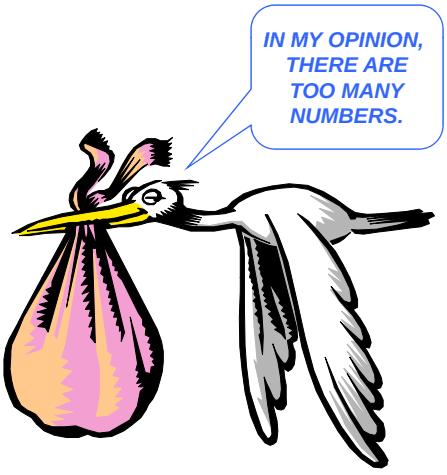


Removing a Number from a Bag

- We may remove a number from a bag.
- But we remove only one number at a time.

How Many Numbers

- Another operation is to determine how many numbers are in a bag.



Summary of the Bag Operations

- A bag can be put in its *initial state*, which is an empty bag – ***constructor***
- Numbers can be *inserted* into the bag – ***insert(…)***
- You may check how many *occurrences* of a certain number are in the bag – ***count(…)***
- Numbers can be *removed* from the bag – ***erase(…)*** and ***erase_one(…)***
- You can check *how many* numbers are in the bag – ***size()***

Sequence

- Another container class
- Also (more traditionally) called ***list***
- Similar to bag – contains bunch of items
- Different from bag – items arranged in an order
  - Which item comes first, which item comes next, …, and which item comes last
  - Items are *not* necessarily in sorted order
    - If items are required to be in sorted order → *sorted sequence*

Textbook Readings

- Chapter 3
  - Section 3.1
  - Section 3.2
- Chapter 4
  - Section 4.3
  - Section 4.4
- Chapter 6
  - Section 6.2