



Collections

Kevin Buffardi
kbuffardi@vt.edu

Collections

A group of (zero or more) data items with a shared significance and organization

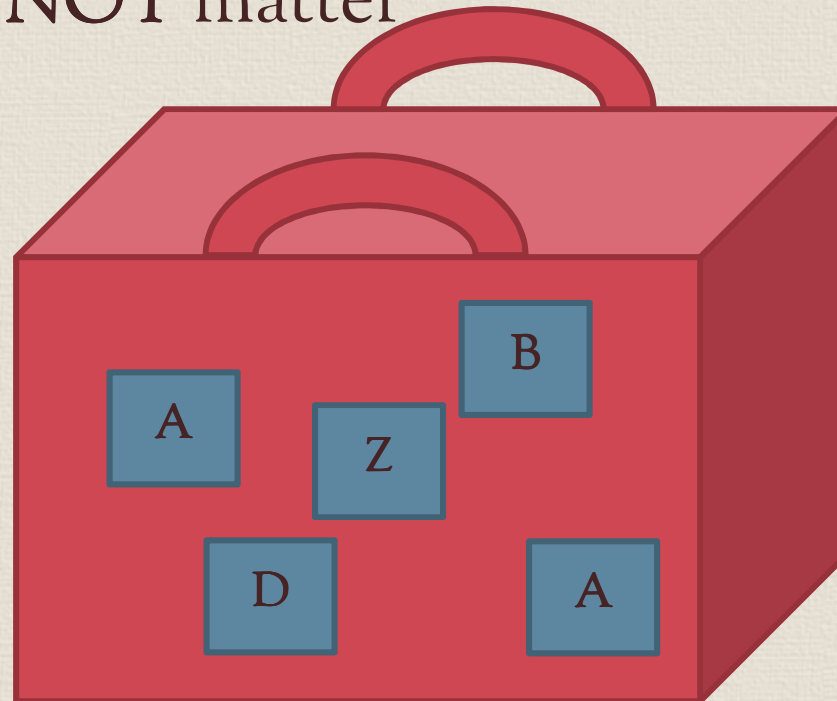
- **Associative:** Bags, Sets, Maps/Dictionaryes
- **Linear:** Lists, Stacks, Queues
- **Non-linear/other:** Trees, Graphs

Collection Interface

```
public interface Collection <T>{  
  
    public void add(T item);  
  
    public void remove(T item);  
  
    public boolean contains(T item);  
  
}
```

Bag

- Associative Collection
- Allows duplicate items
- Order does **NOT** matter



- *Applications?*

Implement ArrayBag

```
public class ArrayBag<T>
    implements Collection<T>{
    //Fields?
    //Constructor?
    public void add(T item)
    public void remove(T item)
    public boolean contains(T item)

}
```

Set

- Similar to Bag
 - Collection
 - Order does **NOT** matter
- Different from Bag
 - Duplicate items **NOT** allowed
- *How should we implement a Set?*
 - *DRY: Don't repeat yourself*
 - Very similar to Bag, but needs to be adapted to disallow duplicate items
 - What is the relationship between Bag and Set?
 - Is-A? Has-A?

Implement ArraySet

```
public class ArraySet<T>
    implements Collection<T> {

    private ArrayBag<T> contents; // "has a"

    public void add(T item) {}
    public void remove(T item) {}
    public boolean contains(T item) {}
}
```

Implement ArraySet

```
public ArraySet() //constructor
{
    contents = new ArrayBag<T>();
}
public void add(T item)
{ //no duplicates!
    if( !contents.contains(item) )
        contents.add(item);
}
```


Implement ArraySet

```
public void remove(T item)
{
    contents.remove(item);
}
```

```
public boolean contains(T item)
{
    return contents.contains(item);
}
```