CSCI 241
Scott Wehrwein
The Map ADT
Goals

Know the purpose and operations of the Map ADT.
Map

• In math, a map is a function.

• What is a function, anyway?
In math, **map** is another word for **function**.

If $F$ is a map then $F(a) \rightarrow b$ means "$a$ maps to $b$".

A map $F$ has a:

- **domain** - set of values $F$ maps **from**
- **range** - set of values $F$ maps a domain element **to**
- **codomain** - the set of **all** values of the range's type whether or not any element in the domain maps to it
Arrays are great!
Arrays are great!

Arrays are a special case of maps: `Thing[] a = new Thing[10];`

- **Domain**: 0..a.length
- **Range**: all elements in the array
- **Codomain**: the array's type

In arrays, we get to choose the **codomain**.

what if we could choose the **domain** too?
The Map ADT

- The **Map ADT** represents a mapping from **keys** to **values**.

- We get to choose the type of both:
  - the **keys** (domain)
  - the **values** (codomain)
Example Uses of Maps

Map<String, Integer> wordCounts;

Map<Student, Character> grades;

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;to&quot;</td>
<td>2</td>
</tr>
<tr>
<td>&quot;be&quot;</td>
<td>2</td>
</tr>
<tr>
<td>&quot;not&quot;</td>
<td>1</td>
</tr>
<tr>
<td>&quot;or&quot;</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>'B'</td>
</tr>
<tr>
<td>String name</td>
<td>&quot;Mal&quot;</td>
</tr>
<tr>
<td>Student</td>
<td>'A'</td>
</tr>
<tr>
<td>String name</td>
<td>&quot;Inara&quot;</td>
</tr>
<tr>
<td>Student</td>
<td>'B'</td>
</tr>
<tr>
<td>String name</td>
<td>&quot;Zoe&quot;</td>
</tr>
<tr>
<td>Student</td>
<td>'C'</td>
</tr>
<tr>
<td>String name</td>
<td>&quot;Jayne&quot;</td>
</tr>
</tbody>
</table>
The Map Interface

```java
public interface Map<K,V> {
    /** Returns the value to which the specified key *
     * is mapped, or null if this map contains no *
     * mapping for the key. */
    V get(Object key);

    /** Associates the specified value with the *
     * specified key in this map */
    V put(K key, V value);

    /** Removes the mapping for a key from this map *
     * if it is present */
    V remove(Object key);

    // more methods
}
```