STL Associative Containers
Overview of STL Associative Containers

- Associative containers are data structures that support efficient insertion, deletion, & find operations based on keys.
Overview of STL Associative Containers

- There are two types of STL associative containers
Overview of STL Associative Containers

- There are two types of STL associative containers
  - **Ordered associative containers** that support efficient operations on elements using keys ordered by `operator<`

<table>
<thead>
<tr>
<th>Category</th>
<th>Containers</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordered associative</td>
<td>set</td>
<td>Defines where the elements’ values are the keys and duplicates are not allowed. It has fast lookup using the key,</td>
</tr>
<tr>
<td></td>
<td>multiset</td>
<td>Defines where the elements’ values are the keys and duplicates are allowed. It has fast lookup using the key,</td>
</tr>
<tr>
<td></td>
<td>map</td>
<td>Key-to-value mapping where a single key can only be mapped to one value,</td>
</tr>
<tr>
<td></td>
<td>multimap</td>
<td>Key-to-value mapping where a single key can be mapped to many values.</td>
</tr>
</tbody>
</table>

See [en.wikipedia.org/wiki/Associative_containers](en.wikipedia.org/wiki/Associative_containers)
Overview of STL Associative Containers

- There are two types of STL associative containers
  - **Ordered associative containers** that support efficient operations on elements using keys ordered by `operator<`
    - Implemented as balanced binary trees

See [en.wikipedia.org/wiki/Red-black_tree](en.wikipedia.org/wiki/Red-black_tree)
Overview of STL Associative Containers

- There are two types of STL associative containers
  - **Ordered associative containers** that support efficient operations on elements using keys ordered by `operator<`
    - Implemented as balanced binary trees
    - Keys are const & can’t be changed via iterators
Overview of STL Associative Containers

- There are two types of STL associative containers
  - **Ordered associative containers** that support efficient operations on elements using keys ordered by `operator<`
  - **Unordered associative containers** that maintain data in structures suitable for fast associative operations

<table>
<thead>
<tr>
<th>Category</th>
<th>Containers</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unordered associative</td>
<td>unordered_set</td>
<td>Defines where the elements' values are the keys and duplicates <em>are not</em> allowed. It has fast lookup using the key,</td>
</tr>
<tr>
<td></td>
<td>unordered_multiset</td>
<td>Defines where the elements’ values are the keys and duplicates <em>are</em> allowed. It has fast lookup using the key,</td>
</tr>
<tr>
<td></td>
<td>unordered_map</td>
<td>Key-to-value mapping where a single key can only be mapped to one value,</td>
</tr>
<tr>
<td></td>
<td>unordered_multimap</td>
<td>Key-to-value mapping where a single key can be mapped to many values.</td>
</tr>
</tbody>
</table>

Overview of STL Associative Containers

- There are two types of STL associative containers
  - **Ordered associative containers** that support efficient operations on elements using keys ordered by `operator<`
  - **Unordered associative containers** that maintain data in structures suitable for fast associative operations
    - Implemented as hash tables

See [en.wikipedia.org/wiki/Hash_table](en.wikipedia.org/wiki/Hash_table)
Overview of STL Associative Containers

- There are two types of STL associative containers
  - **Ordered associative containers** that support efficient operations on elements using keys ordered by `operator<`
  - **Unordered associative containers** that maintain data in structures suitable for fast associative operations
    - Implemented as hash tables
    - Keys are const & can’t be changed via iterators