

Unordered Associative Containers

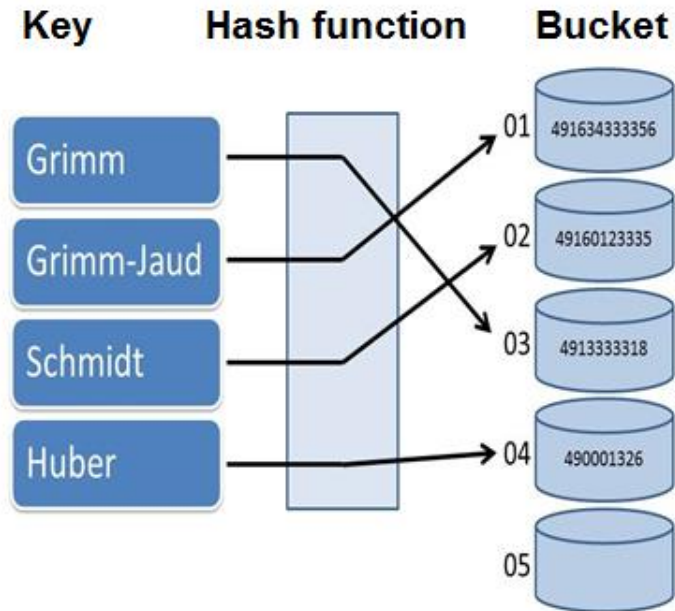
`std::unordered_set`, `std::unordered_multiset`,
`std::unordered_map`, `std::unordered_multimap`

- Are also known as dictionary, associative array, or hash table.
- Have a data type(s), an allocator (default), a hash function (default), and an equal comparison function (default).
- Are quite similar to the ordered associative containers.

Unordered Associative Containers	Value Available	More Identical Keys	Header
<code>std::unordered_set</code>	no	no	<code><unordered_set></code>
<code>std::unordered_multiset</code>	no	yes	<code><unordered_set></code>
<code>std::unordered_map</code>	yes	no	<code><unordered_map></code>
<code>std::unordered_multimap</code>	yes	yes	<code><unordered_map></code>

`unorderedMap.cpp`

Unordered Associative Containers



Classical use case:

- Key: family name
- Value: telephone number

```
std::unordered_map<std::string, int> um{ {"Grimm", 4916343333}, {"Grimm-Jaud", 491601233}, {"Schmidt", 491333318}, {"Huber", 4900013} };
```



The hash function maps the key in constant time to its index.

Unordered Associative Containers

- **Collision:**
 - Unordered associative containers store their keys in the buckets.
 - Different keys with the same hash value can be stored in the same bucket.
 - The access time of the bucket is constant, the search in the bucket is linear.
- **Capacity:**
 - Number of buckets
- **Load factor:**
 - Average number of elements of each bucket.
- **Rehashing:**
 - New buckets are created per default if the load factor is bigger than 1.