8.2. Create, Copy, and Destroy Operations

Containers provide the following constructors and destructors. In addition, most constructors allow you to pass an allocator as an additional argument, covered in Section 8.10, page 430. See Section 7.1.2, page 254, for a general discussion about initializing containers.

```
container::container ()
```

- The default constructor.
- Creates a new empty container.
- For arrays, the operation is implicitly defined and creates a nonempty container where the elements might have undefined values (see Section 7.2.1, page 262).
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

```
explicit container::container (const CompFunc& cmpPred)
```

- Creates a new empty container with *cmpPred* used as the sorting criterion (see Section 7.7.5, page 328, and Section 7.8.6, page 351, for examples).
- The sorting criterion must define a *strict weak ordering* (see Section 7.7, page 314).
- Provided by set, multiset, map, multimap.

Click here to view code image

```
explicit container::container (size_type bnum)
explicit container::container (size_type bnum, const Hasher& hasher)
explicit container::container (size_type bnum, const Hasher& hasher,
                               const KeyEqual& eqPred)
```

- Create a new empty container with at least *bnum* buckets, *hasher* used as hashing function, and *eqPred* used as criterion to identify equal values.
- If *eqPred* is not passed, the default equivalence criterion of the container type is used (see Section 7.9.2, page 366, for details).
- If *hasher* is not passed, the default hashing function of the container type is used (see Section 7.9.2, page 363, for details).
- Provided by unordered set, unordered multiset, unordered map, unordered multimap.

```
container::container (initializer-list)
```

- Creates a new container that is initialized by the elements of *initializer-list*.
- For arrays, the operation is implicitly defined (see Section 7.2.1, page 262).
- Available since C++11.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

```
container::container (initializer-list, const CompFunc& cmpPred)
```

- Creates a container that has the sorting criterion *cmpPred* and is initialized by the elements of *initializer-list*.
- The sorting criterion must define a *strict weak ordering* (see Section 7.7, page 314).
- Available since C++11.
- Provided by set, multiset, map, multimap.

Click here to view code image

```
container::container (initializer-list, size_type bnum)
container::container (initializer-list, size_type bnum,
                      const Hasher&  hasher)
container::container (initializer-list, size_type bnum,
                      const Hasher&  hasher, const KeyEqual&  eqPred)
```

- Create a container with at least *bnum* buckets, *hasher* used as hashing function, and *eqPred* used as criterion to identify equal values, initialized by the elements of *initializer-list*.
- If *eqPred* is not passed, the default equivalence criterion of the container type is used (see Section 7.9.2, page 366, for details).
- If *hasher* is not passed, the default hashing function of the container type is used (see Section 7.9.2, page 363, for details).
- Provided by unordered set, unordered multiset, unordered map, unordered multimap.

container::*container* (const *container*& *c*)

- The copy constructor.
- Creates a new container as a copy of the existing container *c*.
- Calls the copy constructor for every element in *c*.
- For arrays, the operation is implicitly defined.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container::*container* (*container*&& *c*)

- The move constructor.
- Creates a new container initialized with the elements of the existing container *c*.
- After this call, C is valid but has an unspecified value.
- For arrays, the operation is implicitly defined.
- Available since C++11.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

explicit **container**::*container* (size_type *num*)

- Creates a container with *num* elements.
- The elements are created by their default constructor.
- Provided by vector, deque, list, forward list.

container::*container* (size_type *num*, const T& *value*)

- Creates a container with *num* elements.
- The elements are created as copies of *value*.
- T is the type of the container elements.
- For strings, *value* is not passed by reference.
- Provided by vector, deque, list, forward list, string.

container::*container* (InputIterator *beg*, InputIterator *end*)

- Creates a container initialized by all elements of the range [*beg*, *end*).
- This function is a member template (see Section 3.2, page 34). Thus, the elements of the source range may have any type convertible into the element type of the container.
- See Section 7.1.2, page 256, for examples and a discussion of a problem resulting from the fact that this is a member function.
- Provided by vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container::*container* (InputIterator *beg*, InputIterator *end*, const CompFunc& *cmpPred*)

- Creates a container that has the sorting criterion *cmpPred* and is initialized by all elements of the range [*beg*, *end*).
- This function is a member template (see Section 3.2, page 34). Thus, the elements of the source range may have any type convertible into the element type of the container.
- The sorting criterion must define a *strict weak ordering* (see Section 7.7, page 314).
- Provided by set, multiset, map, multimap.

Click here to view code image

```
container::container (InputIterator  beg, InputIterator  end,
size_type  bnum)
container::container (InputIterator  beg, InputIterator  end,
size_type  bnum,
                      const Hasher&  hasher)
container::container (InputIterator  beg, InputIterator  end,
size_type  bnum,
                      const Hasher&  hasher, const KeyEqual&  eqPred)
```

- Create a container with at least *bnum* buckets, *hasher* used as hashing function, and *eqPred* used as criterion to identify equal values, which is initialized by all elements of the range [*beg*, *end*).
- If *eqPred* is not passed, the default equivalence criterion of the container type is used (see Section 7.9.2, page 366, for details).
- If *hasher* is not passed, the default hashing function of the container type is used (see Section 7.9.2, page 363, for details).

- Provided by unordered set, unordered multiset, unordered map, unordered multimap.

container::~*container* ()

- The destructor.
- Removes all elements and frees the memory.
- Calls the destructor for every element.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.