



Detail

Set in C++ Standard Template Library

(Training materials for students)

Sets are a type of associative containers in which each element has to be unique, because the value of the element identifies it. The value of the element cannot be modified once it is added to the set, though it is possible to remove and add the modified value of that element:

Some basic functions associated with Set:

- **begin()** – Returns an iterator to the first element in the set.
- **end()** – Returns an iterator to the theoretical element that follows last element in the set.
- **size()** – Returns the number of elements in the set.
- **max_size()** – Returns the maximum number of elements that the set can hold.
- **empty()** – Returns whether the set is empty.

Methods of set:

- **begin()** – Returns an iterator to the first element in the set.
- **end()** – Returns an iterator to the theoretical element that follows last element in the set.
- **rbegin()** – Returns a reverse iterator pointing to the last element in the container.
- **rend()** – Returns a reverse iterator pointing to the theoretical element right before the first element in the set container.
- **crbegin()** – Returns a constant iterator pointing to the last element in the container.
- **crend()** – Returns a constant iterator pointing to the position just before the first element in the container.
- **cbegin()** – Returns a constant iterator pointing to the first element in the container.
- **cend()** – Returns a constant iterator pointing to the position past the last element in the container.
- **size()** – Returns the number of elements in the set.
- **max_size()** – Returns the maximum number of elements that the set can hold.
- **empty()** – Returns whether the set is empty.
- **insert(const i)** – Adds a new element 'i' to the set.
- **iterator insert (iterator position, const i)** – Adds a new element 'i' at the position pointed by iterator.

- **erase(iterator position)** – Removes the element at the position pointed by the iterator.
- **erase(const i)** – Removes the value 'i' from the set.
- **clear()** – Removes all the elements from the set.
- **key_comp()** / **value_comp()** – Returns the object that determines how the elements in the set are ordered ('<' by default).
- **find(const g)** – Returns an iterator to the element 'g' in the set if found, else returns the iterator to end.
- **count(const g)** – Returns 1 or 0 based on the element 'g' is present in the set or not.
- **lower_bound(const g)** – Returns an iterator to the first element that is equivalent to 'g' or definitely will not go before the element 'g' in the set.
- **upper_bound(const g)** – Returns an iterator to the first element that is equivalent to 'g' or definitely will go after the element 'g' in the set.
- **equal_range()**– The function returns an iterator of pairs. (key_comp). The pair refers to the range that includes all the elements in the container which have a key equivalent to k.
- **emplace()**– This function is used to insert a new element into the set container, only if the element to be inserted is unique and does not already exists in the set.
- **emplace_hint()**– Returns an iterator pointing to the position where the insertion is done. If the element passed in the parameter already exists, then it returns an iterator pointing to the position where the existing element is.
- **swap()**– This function is used to exchange the contents of two sets but the sets must be of same type, although sizes may differ.
- **operator=** – The '=' is an operator in C++ STL which copies (or moves) a set to another set and set::operator= is the corresponding operator function.
- **get_allocator()**– Returns the copy of the allocator object associated with the set.

--- The End ---