# Member Functions of the list Class

| | |
|---|---|
| constructors | Create lists |
| operator= | Copy the contents of a list |
| assign | Assign elements to a list |
| back | Access the last element of a list |
| begin | Return the iterator pointing to the beginning of a list |
| clear | Erase all elements of a list |
| empty | Test whether a list is empty |
| end | Return the iterator pointing to the end of a list |
| erase | Erase elements of a list |
| front | Access the first element of a list |
| insert | Insert elements into a list |
| max_size | Return the largest possible size of a list |
| merge | Merge sorted lists |
| pop_back | Remove the last element of a list |
| pop_front | Remove the first element of a list |
| push_back | Insert an element at the end of a list |
| push_front | Insert an element at the beginning of a list |
| rbegin | Return the reverse_iterator pointing to the beginning of a reversed list |
| remove | Remove all elements of a list with a specific value |
| remove_if | Remove all elements of a list fulfilling with a specified condition |
| rend | Return the reverse_iterator pointing to the end of a reversed list |
| resize | Change the size of a list |
| reverse | Reverse the order of elements of a list |
| size | Return the size of a list |
| sort | Sort the elements of a list |
| splice | Move elements from list to list |
| swap | Swap the contents the contents of two lists |
| unique | Remove duplicate elements of a list |

## Function Prototypes

| | |
|---|---|
| constructors | Create lists |
| | list () – create an empty list |
| | list (size_type n, const T& value =T()) – create a list from n copies of value |
| | template <class II> list (II first, II last) – create a list from a copy of the elements starting from the element referred by the input iterator first to the element right before the one referred by the input iterator last |
| | list (const list<T>& l) – create a copy of the list l |
| destructor | Destroy a list |
| | ~list() – deallocate all the storage capacity allocated by a list |
| operator= | Copy the contents of a list |
| | list<T>& operator= (const list<T>& l) – assign a copy of the list l to a list |
| assign | Assign elements to a list |
| | void assign (size_type n, const T& x) – assign n copies of the element x to a list, replacing its current content |
| | template <class II> void assign (II first, II last) – assign a copy of the elements, starting from the element referred by the input iterator first to the element right before the element referred by the input iterator last, to a list, replacing its current content |
| back | Access the last element of a list |
| | T& back () – return a reference to the last element of a list |
| | const T& back () const – const version of the function |
| begin | Return the iterator pointing to the beginning of a list |
| | iterator begin () – return an iterator to the beginning of a list |
| | const_iterator begin () const – const version of the iterator |
| clear | Erase all elements of a list |
| | void clear () – set a list content to an empty list |
| empty | Test whether a list is empty |
| | bool empty () const – return whether a list is empty |
| end | Return the iterator pointing to the end of a list |
| | iterator end () – return an iterator referring to the end of a list |
| | const_iterator end () const – const version of the iterator |
| erase | Erase elements of a list |
| | iterator erase (iterator p) – erase the element of a list at the position referred by the iterator p |
| | iterator erase (iterator first, iterator last) – erase all the elements of a list between the positions referred by the iterators first and last |
| front | Access the first element of a list |
| | T& front () – return a reference to the first element of a list |
| | const T& front () const – const version of the function |
| insert | Insert elements into a list |
| | iterator insert (iterator i, const T& x) – insert a copy of the element x at the position referred by the iterator i into a list and return an iterator referring to the insert position |
| | void insert (iterator i, size_type n, const T& x) – insert n copies of the element x at the position referred by the iterator i into a list |
| | template <class II> void insert (iterator i, II first, II last) – insert a copy of the elements, |

| | |
|---|---|
| | starting from the element referred by the input iterator first to the element right before the one referred by the input iterator last, at the position referred by the iterator i into a list |
| max_size | Return the largest possible size of a list |
| | size_type max_size () const – return the maximum number of elements that a list can hold |
| merge | Merge sorted lists |
| | void merge (list<T>& l) – merge the list l into a list at their respective ordered positions and empty l |
| | template <class C> void merge (list<T>& l, C cmp) – merge the list l into a list at their respective ordered positions in which the order is determined by the class object cmp for all pairs of elements |
| pop_back | Remove the last element of a list |
| | void pop_back () – remove the last element of a list |
| pop_front | Remove the first element of a list |
| | void pop_front () – remove the first element of a list |
| push_back | Insert an element at the end of a list |
| | void push_back (const T& x) – add a new element at the end of a list |
| push_front | Insert an element at the beginning of a list |
| | void push_front (const T& x) – add a new element at the beginning of a list |
| rbegin | Return the reverse_iterator pointing to the beginning of a reversed list |
| | reverse_iterator rbegin () – return a reverse iterator referring to the last element of a list |
| | const_reverse_iterator rbegin () const – const version of the reverse iterator |
| remove | Remove all elements of a list with a specific value |
| | void remove (const T& x) – remove all the elements of a list with the value of x |
| remove_if | Remove all elements of a list fulfilling with a specified condition |
| | template <class P> void remove_if (P pred) – remove all the elements of a list with the values that the predicate pred returns true |
| rend | Return the reverse_iterator pointing to the end of a reversed list |
| | reverse_iterator rend () – return a reverse iterator referring to the element right before the first element of a list |
| | const_reverse_iterator rend () const – const version of the reverse iterator |
| resize | Change the size of a list |
| | void resize (size_type n, T x =T()) – resize the list content to n elements, and if n is greater than the current size of the list, its content is expanded by filling of the copies of the element x |
| reverse | Reverse the order of elements of a list |
| | void reverse () – reverse the order of elements in a list |
| size | Return the size of a list |
| | size_type size () const – return the number of elements in a list |
| sort | Sort the elements of a list |
| | void sort () – sort the elements of a list in ascending order |
| | template <class C> void sort (C cmp) – sort the elements of a list in which the order is determined by the class object cmp for all pairs of elements |
| splice | Move elements from list to list |
| | void splice (iterator i, list<T>& l) – move all the elements from the list l to the position |

| | |
|---|---|
| | referred by the iterator i of a list and remove them from l |
| | void splice (iterator i, list <T>& l, iterator j) – move the element at the position referred by the iterator j of the list l to the position referred by the iterator i of a list and remove it from l |
| | void splice (iterator i, iterator first, iterator last) – move the elements between the position referred by the iterators first and last at the position referred by the iterator i of a list |
| swap | Swap the contents the contents of two lists |
| | void swap (list<T>& l) – swap the contents of a list with the list l |
| unique | Remove duplicate elements of a list |
| | void unique () – remove all but the first element from every consecutive group of equal elements in a list |
| | template <class P> void unique (P pred) – remove the first element of all pairs of two consecutive elements in a list for which the predicate pred returns true |