Predefine Stack Methods

In C++, the `stack` class provides various methods to perform different operations on a stack.

Operation	Description
`push()`	adds an element into the stack
`pop()`	removes an element from the stack
`top()`	returns the element at the top of the stack
`size()`	returns the number of elements in the stack
`empty()`	returns `true` if the stack is empty

C++ Predefined Queue Methods

In C++, the `queue` class provides various methods to perform different operations on a queue.

Methods	Description
`push()`	inserts an element at the back of the queue
`pop()`	removes an element from the front of the queue
`front()`	returns the first element of the queue
`back()`	returns the last element of the queue
`size()`	returns the number of elements in the queue
`empty()`	returns `true` if the queue is empty

Basic operations on List

C++ STL provides various functions that we can use to perform different operations on lists. Let's look at some commonly used list functions to perform the following operations:

1. Add elements
2. Access elements
3. Remove elements

---

1. Add Elements to a List in C++

We can add values in a list using the following functions:

- `push_front()` - inserts an element to the beginning of the list
- `push_back()` - adds an element to the end of the list

2. Access List Elements

We can access list elements using the following functions:

- `front()` - returns the first element of the list
- `back()` - returns the last element of the list

3. Delete List Elements

We can delete list elements using the following functions:

- `pop_front()` - removes the element at the beginning of the list
- `pop_back()` - removes the element at the end of the list

Other List Functions in C++

While there are many functions that can be used with lists, we will only look at some of the functions in the table below:

Function	Description
`reverse()`	Reverses the order of the elements.
`sort()`	Sorts the list elements in a particular order.
`unique()`	Removes consecutive duplicate elements.
`empty()`	Checks whether the list is empty.
`size()`	Returns the number of elements in the list.
`clear()`	Clears all the values from the list
`merge()`	Merges two sorted lists.