Map

`#include <map>

std::map<KeyType, ValueType> mapName;`

Navigating Maps:

- Key-Value Chronicle: Pairs data in unique key-value sets.
- Uniqueness Rule: Each key maps to a single value, but values can repeat.
- Sorting Saga: Yields sorted output; unordered_map does not.
- Insert-Erase-Find: O(log n) complexity due to red-black tree in map; O(1) in unordered_map (hash table).
- Map's Toolkit: Methods include insert, erase, find, count, enriching data management.

Methods:

1. insert(std::pair<KeyType, ValueType>): Inserts a key-value pair into the map.
2. emplace(): Inserts a key-value pair in-place.
3. emplace_hint(): Inserts a key-value pair with a hint for where it should be positioned.
4. erase(iterator): Removes a key-value pair pointed to by the iterator.
5. erase(key): Removes the key-value pair with the specified key.
6. clear(): Removes all key-value pairs from the map.
7. size(): Returns the number of key-value pairs in the map.
8. empty(): Checks if the map is empty (i.e., if its size is zero).
9. find(key): Finds a key in the map and returns an iterator to the corresponding key-value pair.
10. count(key): Counts the occurrences of a key in the map (1 if present, 0 otherwise).

11. lower_bound(key): Returns an iterator to the first key-value pair that is not less than a specified key.

12. upper_bound(key): Returns an iterator to the first key-value pair that is greater than a specified key.

13. equal_range(key): Returns a pair of iterators representing the range of key-value pairs with the specified key.

14. begin(): Returns an iterator to the beginning of the map.

15. end(): Returns an iterator to the end of the map.

16. rbegin(): Returns a reverse iterator to the reverse beginning of the map.

17. rend(): Returns a reverse iterator to the reverse end of the map.