STANDARD TEMPLATE LIBRARY (STL)

(continue)

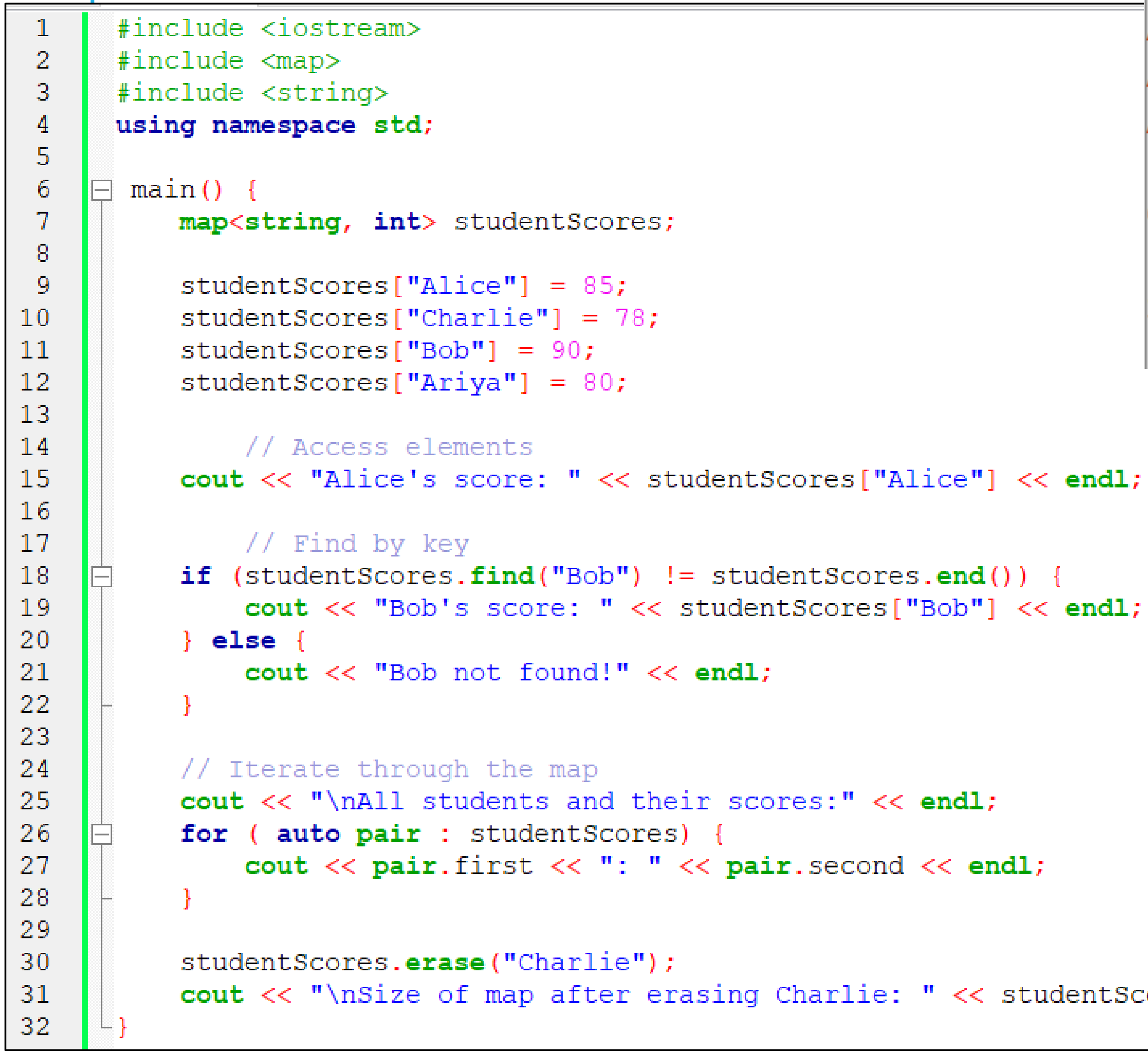
OUTLINE

- Map
- Unordered hash map
- Examples

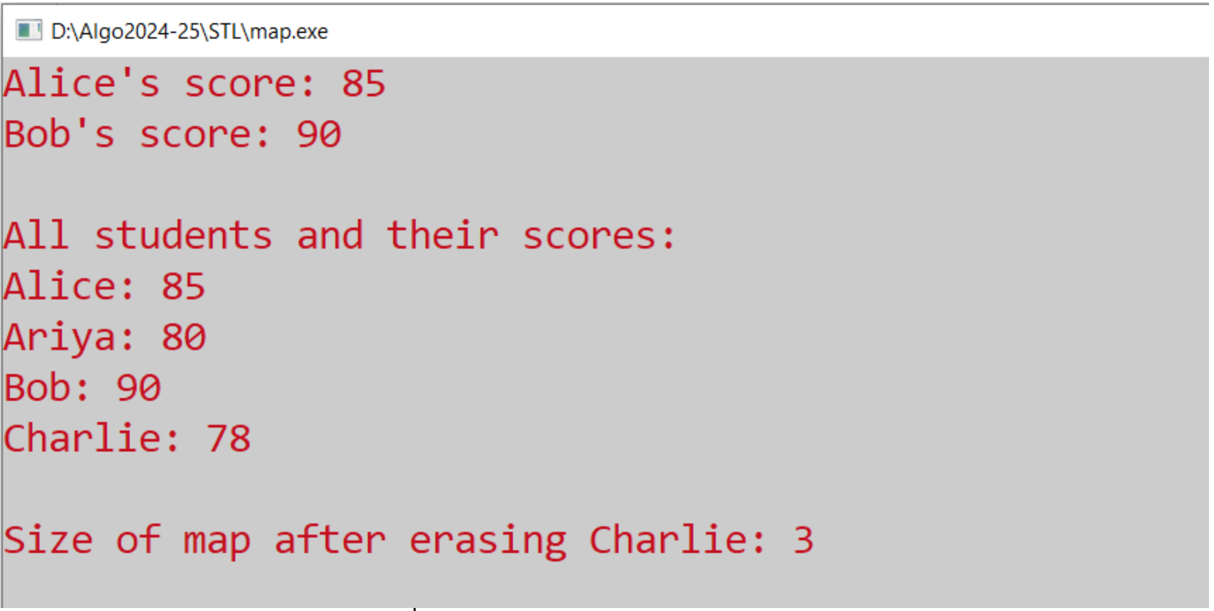
INTRODUCTION

Map	**Unordered hash map**
Uses a balanced **binary search tree** as the underlying data structure.	Uses **a hash table** as the underlying data structure.
When data needs to be sorted or accessed in order.	When quick lookups are needed, and the order of elements doesn't matter.
Example: Leaderboards, frequency counters with order, interval problems.	Example: Caching, counting elements, dictionary-like applications.
Time complexity: O(log n)	Time complexity: O(1) (Worst-case: O(n))

EXAMPLES: USING MAP



```cpp
#include <iostream>
#include <map>
#include <string>
using namespace std;

main() {
    map<string, int> studentScores;

    studentScores["Alice"] = 85;
    studentScores["Charlie"] = 78;
    studentScores["Bob"] = 90;
    studentScores["Ariya"] = 80;

        // Access elements
    cout << "Alice's score: " << studentScores["Alice"] << endl;

        // Find by key
    if (studentScores.find("Bob") != studentScores.end()) {
        cout << "Bob's score: " << studentScores["Bob"] << endl;
    } else {
        cout << "Bob not found!" << endl;
    }

    // Iterate through the map
    cout << "\nAll students and their scores:" << endl;
    for ( auto pair : studentScores) {
        cout << pair.first << ": " << pair.second << endl;
    }

    studentScores.erase("Charlie");
    cout << "\nSize of map after erasing Charlie: " << studentScores.size() << endl;
}
```

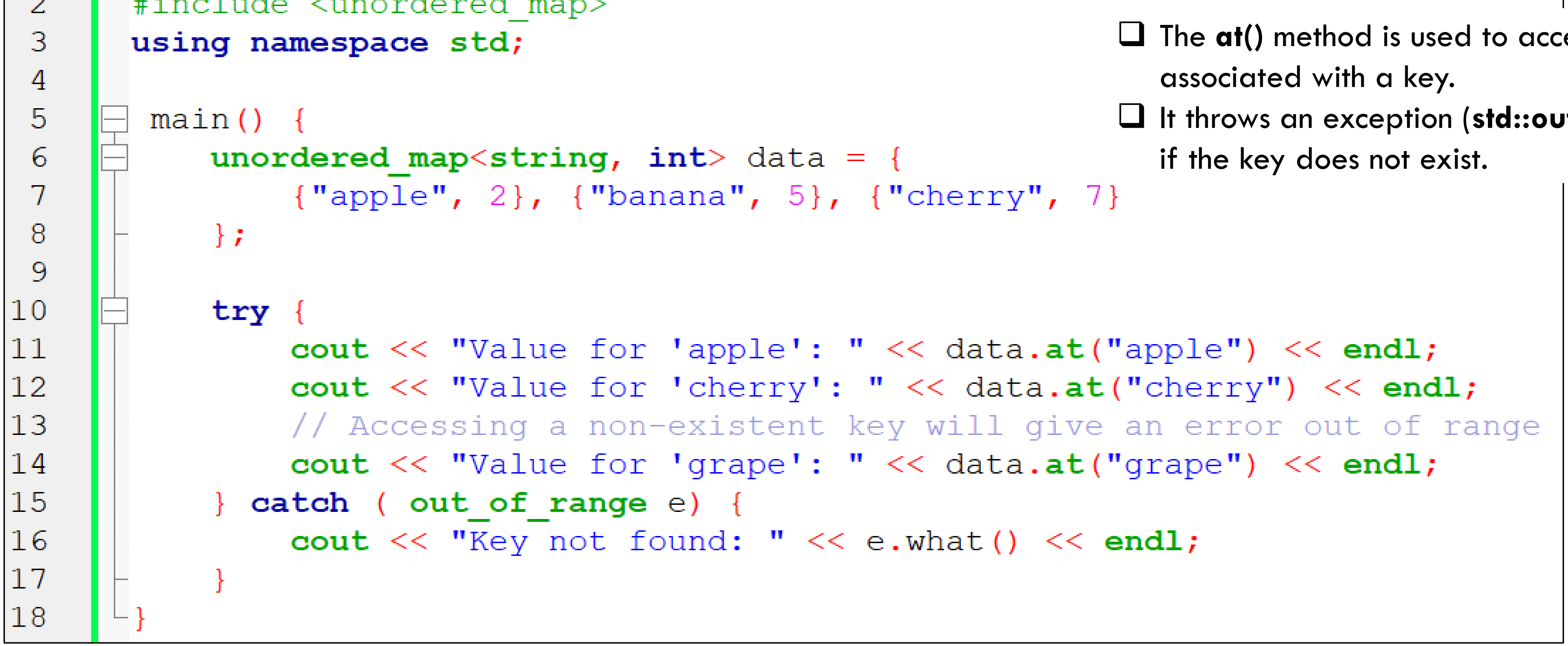


D:\Algo2024-25\STL\map.exe

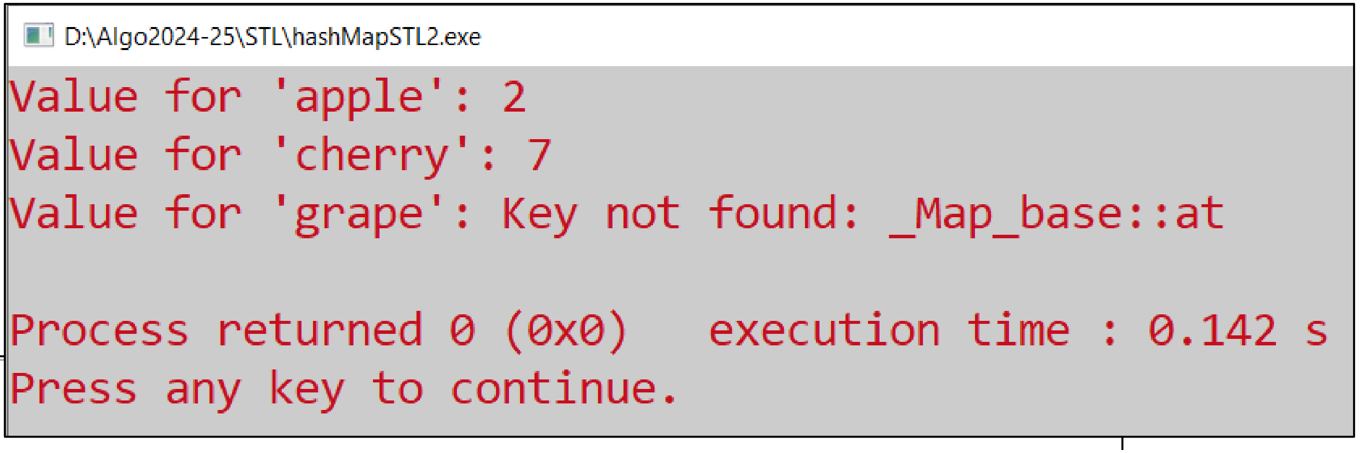
```
Alice's score: 85
Bob's score: 90

All students and their scores:
Alice: 85
Ariya: 80
Bob: 90
Charlie: 78

Size of map after erasing Charlie: 3
```

Using method **at( )**

EXAMPLES:
USING UNORDERED MAP



```cpp
#include <iostream>
#include <unordered_map>
using namespace std;

 main() {
    unordered_map<string, int> data = {
        {"apple", 2}, {"banana", 5}, {"cherry", 7}
    };

    try {
        cout << "Value for 'apple': " << data.at("apple") << endl;
        cout << "Value for 'cherry': " << data.at("cherry") << endl;
        // Accessing a non-existent key will give an error out of range
        cout << "Value for 'grape': " << data.at("grape") << endl;
    } catch ( out_of_range e) {
        cout << "Key not found: " << e.what() << endl;
    }
}
```

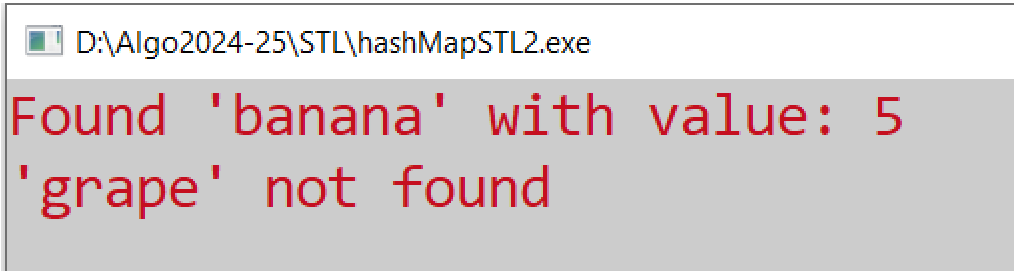


```
Value for 'apple': 2
Value for 'cherry': 7
Value for 'grape': Key not found: _Map_base::at

Process returned 0 (0x0)   execution time : 0.142 s
Press any key to continue.
```

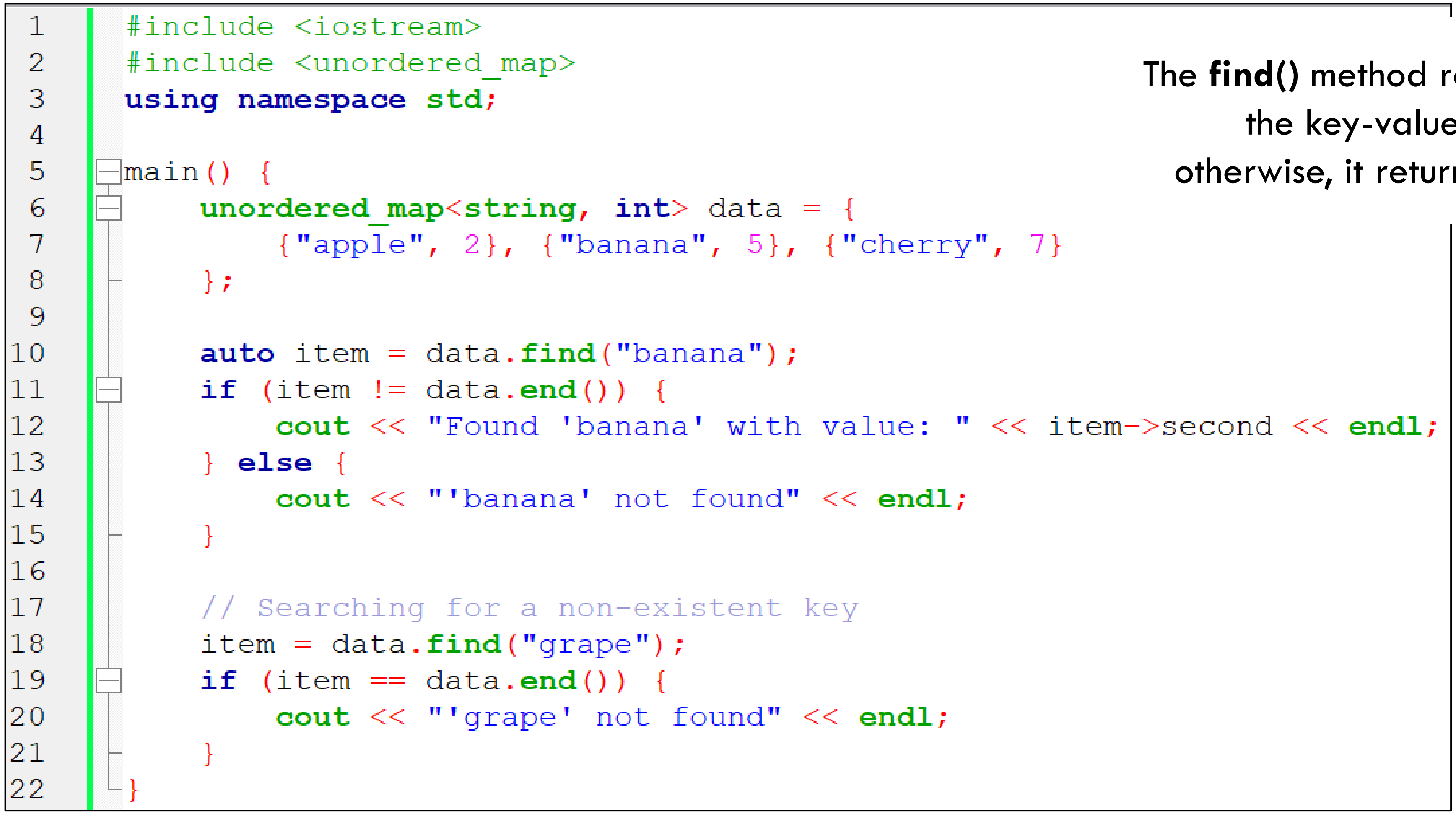
- ❑ The **at()** method is used to access the value associated with a key.
- ❑ It throws an exception (**std::out_of_range**) if the key does not exist.

Using method **find( )**

EXAMPLES:
USING UNORDERED MAP



```
Found 'banana' with value: 5
'grape' not found
```

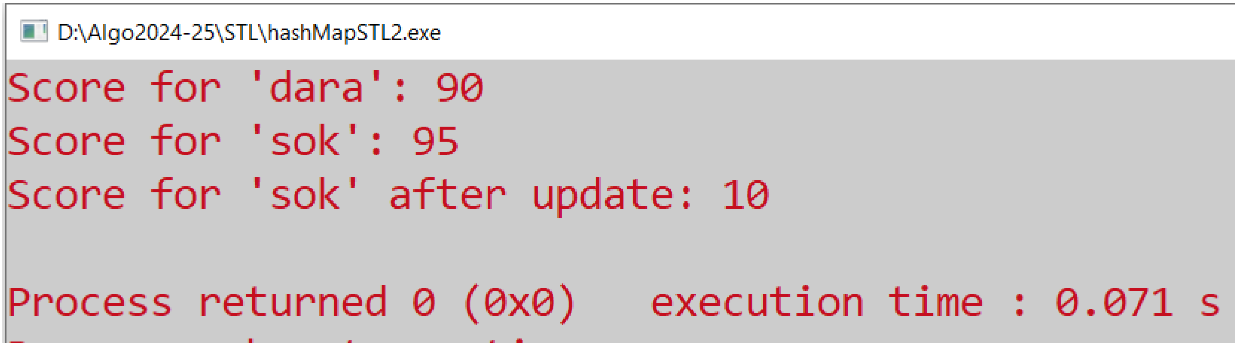


```
#include <iostream>
#include <unordered_map>
using namespace std;

main() {
    unordered_map<string, int> data = {
        {"apple", 2}, {"banana", 5}, {"cherry", 7}
    };

    auto item = data.find("banana");
    if (item != data.end()) {
        cout << "Found 'banana' with value: " << item->second << endl;
    } else {
        cout << "'banana' not found" << endl;
    }

    // Searching for a non-existent key
    item = data.find("grape");
    if (item == data.end()) {
        cout << "'grape' not found" << endl;
    }
}
```

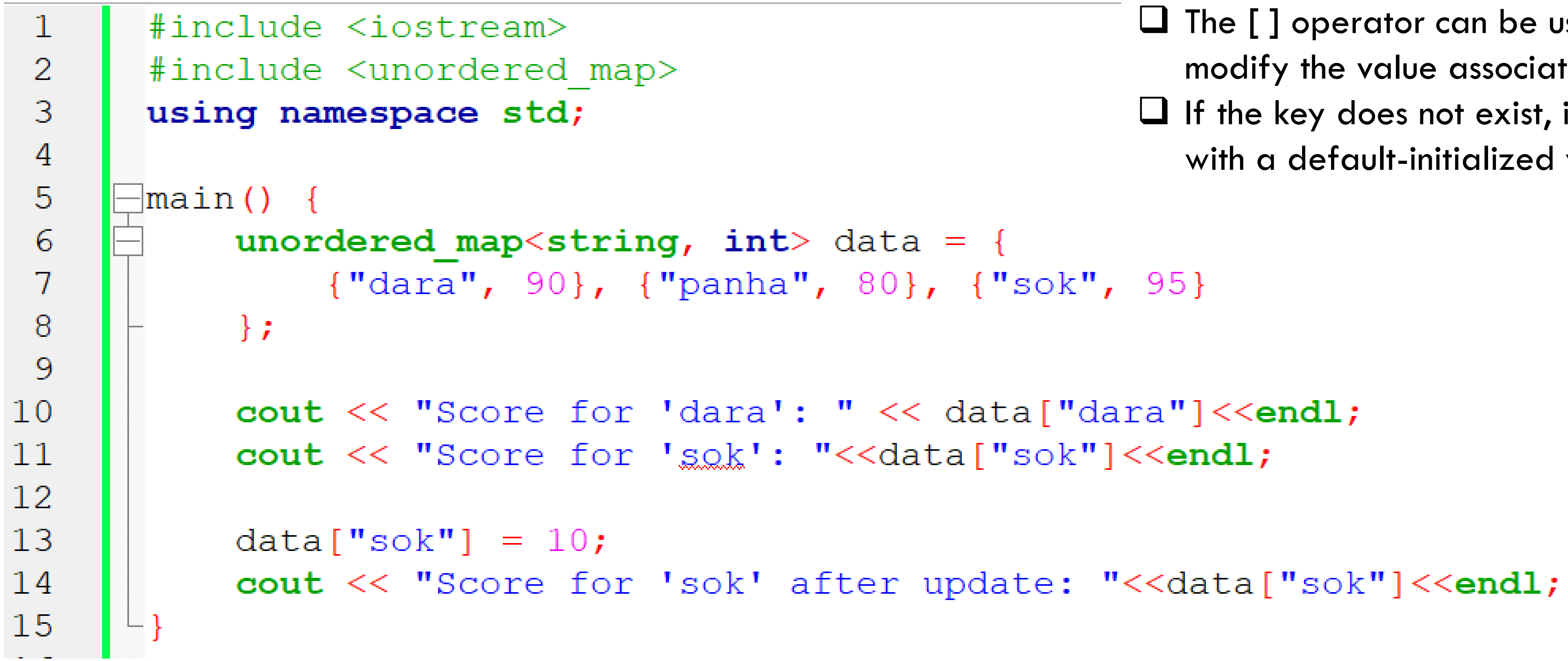
The **find()** method returns an iterator pointing to the key-value pair if the key exists; otherwise, it returns **unordered_map::end().**

Using operator [ ]

EXAMPLES:
USING UNORDERED MAP



```
Score for 'dara': 90
Score for 'sok': 95
Score for 'sok' after update: 10

Process returned 0 (0x0)   execution time : 0.071 s
```



```
#include <iostream>
#include <unordered_map>
using namespace std;

main() {
    unordered_map<string, int> data = {
        {"dara", 90}, {"panha", 80}, {"sok", 95}
    };

    cout << "Score for 'dara': " << data["dara"]<<endl;
    cout << "Score for 'sok': "<<data["sok"]<<endl;

    data["sok"] = 10;
    cout << "Score for 'sok' after update: "<<data["sok"]<<endl;
}
```

- ❑ The [ ] operator can be used to access or modify the value associated with a key.
- ❑ If the key does not exist, it inserts the key with a default-initialized value.

Q&A

Quiz

- ❑Quiz in Moodle
- ❑Quiz topic: STL and Time complexity
- ❑Duration: 30mn
  - ▪ Start: 4:00pm