

# C++ Chapter 6: Unordered Maps

## Contents

C++ Chapter 6: Unordered Maps .....	1
DRY.....	1
Online Tutorial .....	1
Unordered Map .....	1
Basic Operations .....	2
Tutorial 1: Unordered Map Fun .....	2
Assignment 1: Fruit Inventory .....	4
Assignment Submission.....	5



No AI use.

Time required: 60 minutes

## DRY

Don't Repeat Yourself

## Online Tutorial

Go through the following tutorials before starting the tutorials

- [C++ Unordered Map](#)

## Unordered Map

An C++ unordered\_map stores data in key value pairs. This is very much like Java HashMaps or Python dictionaries.

- **Key-Value Pairs**
  - Elements are stored as pairs of keys and their associated values.
  - Keys must be unique in the map, and values can be duplicated.

- **Fast Lookup**
  - Provides rapid access to elements based on their keys.
- **No Specific Order**
  - Unlike map, unordered\_map does not maintain elements in a particular order based on keys.
- **Iterator Support**
  - Enables traversal through the unordered map using iterators or range-based for loops.
- **Dynamic Resizing**
  - Automatically adjusts its size to maintain efficient performance, resizing when necessary.

---

## Basic Operations

- **Insertion:** Add elements to the unordered map using the `[]` operator or the `insert()` method.
- **Accessing Elements:** Retrieve elements using their keys with the `[]` operator or `at()` method.
- **Modification:** Update the value associated with a key by assigning a new value.
- **Deletion:** Remove elements using the `erase()` method based on keys.

## Tutorial 1: Unordered Map Fun

```

1  /**
2   * Filename: unordered_map_fun.cpp
3   * Written by:
4   * Written on:
5   * C++ program to demonstrate functionality of unordered_map
6   * unordered_map is like a Java HashMap or a Python dictionary
7   * elements are stored in key value pairs
8   */
9
10 #include <iostream>
11 #include <unordered_map>
12
13 CodiumAI: Options | Test this function
14 int main()
15 {
16     // Declaring umap to be of <string, int> type
17     // key is will be of string type
18     // mapped value will be of int type
19     std::unordered_map<std::string, int> umap;
20
21     // Insert values by using [] operator
22     // Add key-value pairs to the unordered map
23     umap["ten"] = 10;
24     umap["twenty"] = 20;
25     umap["thirty"] = 30;
26
27     // Iterate through the unordered map using a for-each loop
28     // 'auto' is used to let C++ determine the data type
29     for (auto x : umap)
30     {
31         // Print key-value pairs
32         std::cout << x.first << " " << x.second << std::endl;
33     }
34
35     // Access value with a key
36     // Retrieve the value associated with the key "ten"
37     std::cout << "Value at key ten: " << umap.at("ten") << std::endl;
38
39     // Change the value associated with the key "ten"
40     umap["ten"] = 100;
41
42     // Traversing an unordered map after changing a value
43     for (auto x : umap)
44     {
45         // Printing updated key-value pairs
46         std::cout << x.first << " " << x.second << std::endl;
47     }
48
49     return 0;
50 }

```

Example run:

```
thirty 30
twenty 20
ten 10
Value at key ten: 10
thirty 30
twenty 20
ten 100
```

## Assignment 1: Fruit Inventory

This is the Java Fruit HashMap program back again for more fun.

1. Create a C++ program called **fruit\_inventory.cpp**
2. Change the method of element entry from hard coded to user entry.
3. Use a loop to continue adding elements.
4. Allow user to exit loop when they are done.
5. Print inventory for each loop, final inventory when they are done.

Example run:

```
--- Welcome to Bill's Fruit Inventory System ---
Enter a fruit name to add: Apple
Enter the quantity for Apple: 45

Updated Fruit Inventory:
Key: Apple, Value: 45

Do you want to add more fruits? (y/n): y
Enter a fruit name to add: Orange
Enter the quantity for Orange: 23

Updated Fruit Inventory:
Key: Apple, Value: 45
Key: Orange, Value: 23

Do you want to add more fruits? (y/n): y
Enter a fruit name to add: Pineapple
Enter the quantity for Pineapple: 23

Updated Fruit Inventory:
Key: Apple, Value: 45
Key: Pineapple, Value: 23
Key: Orange, Value: 23

Do you want to add more fruits? (y/n): n

Final Fruit Inventory:
Key: Apple, Value: 45
Key: Pineapple, Value: 23
Key: Orange, Value: 23
```

---

## Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.