Advanced C++ with Examples

This blog aims to explain Advanced C++ concepts with simple examples. I found that sometimes programmers can learn the concepts very easily with a well written and commented code rather then going through hundreds of book pages.

Home About this Blog Other websites with C++ code examples

Thursday, 2 April 2009

An example of C++ Multimap

The following is a simple example of C++ Multimap class.

```
//Program tested on Microsoft Visual Studio 2008 - Zahid Ghadialy
//This program shows use of multi-maps
//We input multiple phone numbers for different people
#include<iostream>
#include<map>
#include <string>

using namespace std;

//forward declaration
void printer(multimap<string, int> pN);

int main()
{
  multimap<string, int> phoneNums;

  //Insert key, value as pairs
  phoneNums.insert(pair<string, int>("Joe",123));
  phoneNums.insert(pair<string, int>("Will",444));
  printer(phoneNums);

  //Insert duplicates
  phoneNums.insert(pair<string, int>("Joe",369));
  phoneNums.insert(pair<string, int>("Smith",567));
  phoneNums.insert(pair<string, int>("Joe",888));
  phoneNums.insert(pair<string, int>("Will",999));
  printer(phoneNums);

  //Checking frequency of different keys
  cout<<"\n\nFrequency of different names"<<endl;
  cout<<"Number of Phones for Joe = "<<phoneNums.count("Joe")<<end
l;
  cout<<"Number of Phones for Will = "<<phoneNums.count("Will")<<e
ndl;
```

```
  cout<<"Number of Phones for Smith = "<<phoneNums.count("Smith")<<endl;
  cout<<"Number of Phones for Zahid = "<<phoneNums.count("Zahid")<<endl;

  //Print all Joe from the list and then erase them
  pair<multimap<string,int>::iterator, multimap<string,int>::iterator> ii;
  multimap<string, int>::iterator it; //Iterator to be used along with ii
  ii = phoneNums.equal_range("Joe"); //We get the first and last entry in ii;
  cout<<"\n\nPrinting all Joe and then erasing them"<<endl;
  for(it = ii.first; it != ii.second; ++it)
  {
    cout<<"Key = "<<it->first<<"    Value = "<<it->second<<endl;
  }
  phoneNums.erase(ii.first, ii.second);
  printer(phoneNums);

  return 0;
}

//This method prints the vector
void printer(multimap<string, int> pN)
{
  cout<<"\n\nMultimap printer method"<<endl;
  cout<<"Map size = "<<pN.size()<<endl;
  multimap<string, int>::iterator it = pN.begin();
  while(it != pN.end())
  {
    cout<<"Key = "<<it->first<<"    Value = "<<it->second<<endl;
    it++;
  }
}
```

The output is as follows:

niazi wrote on 2016-02-26 at 14:43 to 'auto' and 'register' variables (4)

fapmachado wrote on 2016-01-28 at 04:46 to Reading and Writing from Windows Registry (3)

Anonymous wrote on 2016-01-14 at 16:28 to An example of C++ maps (2)

Anonymous wrote on 2015-01-19 at 17:32 to An example of C++ Multimap (9)

Srinivasareddy Bade wrote on 2015-01-06 at 02:13 to Example of C++ class invariant

Zahid Ghadialy wrote on 2014-09-16 at 13:53 to Synchronous and Asynchronous calls in C++ (3)

Anonymous wrote on 2014-09-16 at 13:39 to Synchronous and Asynchronous calls in C++ (3)

Anonymous wrote on 2013-03-08 at 10:00 to C++ example of Proxy Design Pattern (5)

Stanislav Zlatopolsky wrote on 2012-12-12 at 17:46 to C++ example for Prototype Design Pattern (2)

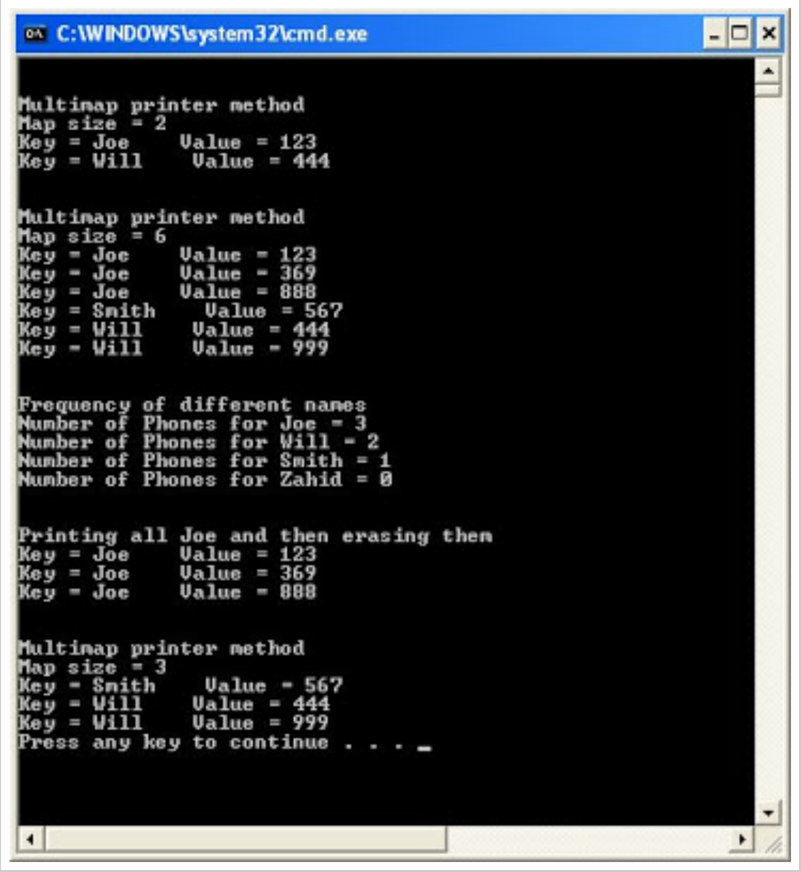
Zahid Ghadialy wrote on 2012-11-23 at 08:22 to Reading and Writing from Windows Registry (3)

Next ↑

Based on a Widget by Review Of Web

About Me

ZAHID GHADIALY



```
Multimap printer method
Map size = 2
Key = Joe     Value = 123
Key = Will     Value = 444

Multimap printer method
Map size = 6
Key = Joe     Value = 123
Key = Joe     Value = 369
Key = Joe     Value = 888
Key = Smith     Value = 567
Key = Will     Value = 444
Key = Will     Value = 999

Frequency of different names
Number of Phones for Joe = 3
Number of Phones for Will = 2
Number of Phones for Smith = 1
Number of Phones for Zahid = 0

Printing all Joe and then erasing them
Key = Joe     Value = 123
Key = Joe     Value = 369
Key = Joe     Value = 888

Multimap printer method
Map size = 3
Key = Smith     Value = 567
Key = Will     Value = 444
Key = Will     Value = 999
Press any key to continue . . . _
```

Posted by Zahid Ghadialy at 04:48

G+1 Recommend this on Google

Labels: Forward Declarations, Multimap, STL

9 comments:

Anonymous *14 August 2009 at 14:17*

Hi

Actually i have one question...

i just want to konw how can i instantiate a multimap if a declare it inside a calss..

Reply

Zahid Ghadialy *14 August 2009 at 17:06*

You can instantiate it in the constructor or in a method. I will post a simple example later today.

Reply

Anonymous *17 November 2010 at 15:19*

Mobile Telecoms Consultant, C++ Coder, Technology Analyst, Trainer, Technologist, Strategic Recruiter, Blogger @ 3g4g, Webmaster at 3g4g.co.uk, Managing Director @ eXplanoTech

View my complete profile

Blog Archive

- ► 2013 (1)
- ► 2012 (2)
- ► 2011 (24)
- ► 2010 (51)
- ▼ 2009 (77)
  - ► December (4)
  - ► November (3)
  - ► October (4)
  - ► September (8)
  - ► August (11)
  - ► July (8)
  - ► June (9)
  - ► May (4)
  - ▼ April (8)
    - Abuse of try and catch in exception handling
    - Using Const Cast
    - Example showing Pass by Value and Pass by Referenc...
    - An example of using Typename with Templates
    - Functions, Methods and

I need to use multi-map in C to store my data. I can't use the C++ Multimap.
is there any suggestions?

Reply

Zahid Ghadialy *17 November 2010 at 15:47*

You can try using linked-lists.

Reply

Anonymous *9 January 2011 at 08:22*

Thanks for this. I need to make use of every one of these examples you've shown. Thanks again

Reply

Anonymous *18 March 2011 at 14:14*

Great blog...good examples which are clear and concise. Keep up the good work.

Reply

Anonymous *12 May 2011 at 18:45*

Can you use class objects as the dataType? e.g.

multimap borrowedVideos;

Reply

Zahid Ghadialy *21 May 2011 at 11:39*

Yes you can but not as the way you have described. Suppose you have a class called Videos then you can have Videos borrowedVideos;

multimap is a STL container and you cant use it the way you have written.

Reply

Anonymous *19 January 2015 at 17:32*

Thanks.. A very helpful example.

Reply

Operations

A simple example of C++ Exceptions

A Word and Letter Counter program example

An example of C++ Multimap

► March (18)

Labels

Abstract Class (2)

Algorithms (5)

Algorithms copy (1)

Algorithms equal_range (1)

Algorithms lower_bound (1)

Algorithms min (1)

Algorithms set_intersection (1)

Algorithms set_union (1)

Algorithms sort (1)

Algorithms upper_bound (1)

Arguments (2)

Array (3)

Assignment Operator (1)

Asynchronous and Synchronous (1)

Auto Pointer (1)

Auto Variables (1)

Base Class (5)

BitSets (1)

BitString (1)

Bitwise Operators (1)

Break (1)

C String (2)

C++0x (1)

Enter your comment...

Comment as: Google Accour

Publish Preview

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

- Callbacks (1)
- Cast (6)
- Challenging Problems (10)
- Cin (3)
- Class (4)
- Client Server (1)
- Command-line arguments (1)
- Compiler Warnings (2)
- Const Cast (2)
- Const correctness (6)
- Const Functions (3)
- Const Pointer (2)
- Const References (1)
- Constructor (3)
- Copy (1)
- Copy Constructor (3)
- Cout (1)
- Date and Time (2)
- Debugging (3)
- Deep Copy and Shallow Copy (1)
- Derived Class (6)
- Design Pattern Abstract Factory (1)
- Design Pattern Adapter (1)
- Design Pattern Bridge (1)
- Design Pattern Builder (1)
- Design Pattern Composite (1)
- Design Pattern Decorator (1)
- Design Pattern Facade (1)
- Design Pattern Factory Method (1)
- Design Pattern Flyweight (1)
- Design Pattern Observer (1)
- Design Pattern Prototype (1)

Design Pattern Proxy (1)

Design Pattern Singleton (3)

Design Pattern State (1)

Design Patterns (15)

Destructor (2)

Directory Operations (1)

Divide By Zero (1)

Dynamic Cast (1)

Enum (2)

Environment Variables (1)

Event (1)

Exceptions (5)

File Operations (5)

Find (3)

Find First Not Of (1)

Find If (1)

Flushing Buffers (1)

Forward Declarations (4)

Friend (1)

Function Pointers (3)

Functions (6)

Functors (2)

General (1)

Goto (1)

Hexadecimal (1)

HexString (1)

Inheritance (2)

Instantiation (1)

Interface Class (1)

Interview Questions (22)

Invariants (1)

Labels (2)

Linked Lists (1)

Lists (3)

Macro (2)

Maps (5)

Mathematical Operations (4)

Memory Allocation (5)

Memory Leaks (3)

Memory Management (1)

Multi-Threading (3)

Multimap (3)

Multiset (2)

Mutable (1)

Mutex (3)

New and Delete (7)

OOPS (1)

Operator Overloading (5)

Operator Precedence (1)

Order of Evaluation (1)

Overloading (4)

Pair (1)

Pass by Reference (1)

Pointer to Const (3)

Pointers (3)

Preprocessor (5)

Print Formatting (2)

Priority Queue (3)

Procedures (1)

Pure Virtual (2)

Queue (1)

Redefinition (1)

References (3)

Register Variables (1)

Reinterpret Cast (3)

Return from Functions (1)

Reverse Engineering (1)

Scope (1)

Search (1)

Set (3)

Singleton (3)

Size (3)

Size_t (1)

SizeOf (2)

Sleep (1)

Smart Pointers (1)

Socket (1)

Sorting (1)

Static Cast (2)

STL (13)

String (17)

String Convert (1)

String Erase (1)

String Length (1)

String Replace (2)

String Size (1)

String Split (1)

String Stream (5)

Struct (1)

Switch Case (1)

Templates (4)

Temporary Objects (1)

Terminology (1)

Test and Debug (1)

Thread Synchronisation (5)

Timers (4)

Transform (1)

Typedef (1)

Typeid (1)

Typename (1)

Union (1)

Variadic Functions (1)

Vector (10)

Virtual (3)

Virtual Destructor (1)

Virtual Inheritance (1)

Void (1)

Volatile (1)

Wide Charachters (2)

Windows Registry (1)

Xerces (3)