

## C++FA 2.1 Hash\_Map Class Example

This simple example shows the inner workings of a Hash\_Map class. The Hash\_Map class maintains an array of pairs of template types—K (key) and V (value). An Iterator is a smart pointer that can point to each pair in the array sequentially. This is a very simple example and is not extensible, but it should give you the basic idea. This example fits in section C++FA 2.1 of *C++ with Financial Applications* by Ben Van Vliet, available at [www.benvanvliet.net](http://www.benvanvliet.net).

### Hash\_Map.h

```
template< class K, class V >
class Hash_Map
{
private:
    pair< K, V > *m_Table;
    const int SIZE;

    unsigned int get_hashcode( const K &key )
    {
        // Modification to hash algorithm by Justin Sobel
        unsigned int hash = 1315423911;
        for( size_t i = 0; i < key.length(); i++ )
        {
            hash ^= (( hash << 5 ) + key[ i ] + ( hash >> 2 ));
        }
        return hash % 100;
    }
public:
    Hash_Map() : SIZE( 100 )
    {
        m_Table = new pair< K, V >[ SIZE ];
    }
    ~Hash_Map()
    {
        delete [] m_Table;
    }
    void insert( pair< K, V > &val )
    {
        m_Table[ get_hashcode( val.first ) ] = val;
    }
    pair< K, V > *find( K &key )
    {
        return &m_Table[ get_hashcode( key ) ];
    }
    pair< K, V > *begin()
    {
        int i = 0;
        while( m_Table[ i ].second == 0 )
        {
            i++;
        }
    }
};
```

```

        return &m_Table[ i ];
    }
pair< K, V > *end()
{
    return &m_Table[ 99 ];
}

class iterator
{
private:
    pair< K, V > *current;
    pair< K, V > *end;

public:
    void operator=( pair< K, V > *temp )
    {
        current = temp;
    }
    bool operator!=( pair< K, V > *temp )
    {
        end = temp;
        return current != temp;
    }
    void operator++()
    {
        do
        {
            ++current;
        } while( current->second == 0 && current != end );
    }
    pair< K, V > *operator->()
    {
        return current;
    }
    pair< K, V > &operator*()
    {
        return *current;
    }
};
};

```

### Main.cpp for < string, double > Hash\_Map

```

#include <iostream>
#include "Hash_Map.h"
using namespace std;

int main()
{
    Hash_Map < string, double > m_Map;
    Hash_Map < string, double >::iterator m_Iter;

    typedef pair< string, double > MyPair;

```

```
m_Map.insert( MyPair( "IBMDP", 10.0 ) );
m_Map.insert( MyPair( "IBMEP", 15.0 ) );
m_Map.insert( MyPair( "IBMDO", 20.0 ) );
m_Map.insert( MyPair( "IBMMX", 25.0 ) );

for ( m_Iter = m_Map.begin(); m_Iter != m_Map.end(); m_Iter++ )
{
    cout << m_Iter->first << "\t" << m_Iter->second << endl;
}

string sym( "IBMDP" );
cout << m_Map.find( sym )->second << endl;

return 0;
}
```