| **#include<iostream>;**  **#include <list>;**  **using namespace std;**    **class Hashing**  **{**  **int hash\_bucket; // No. of buckets**    **// Pointer to an array containing buckets**  **list<int> \*hashtable;**  **public:**  **Hashing(int V); // Constructor**    **// inserts a key into hash table**  **void insert\_key(int val);**    **// deletes a key from hash table**  **void delete\_key(int key);**    **// hash function to map values to key**  **int hashFunction(int x) {**  **return (x % hash\_bucket);**  **}**    **void displayHash();**  **};**    **Hashing::Hashing(int b)**  **{**  **this->hash\_bucket = b;**  **hashtable = new list<int>[hash\_bucket];**  **}**    **//insert to hash table**  **void Hashing::insert\_key(int key)**  **{**  **int index = hashFunction(key);**  **hashtable[index].push\_back(key);**  **}**    **void Hashing::delete\_key(int key)**  **{**  **// get the hash index for key**  **int index = hashFunction(key);**    **// find the key in (inex)th list**  **list <int> :: iterator i;**  **for (i = hashtable[index].begin();**  **i != hashtable[index].end(); i++) {**  **if (\*i == key)**  **break;**  **}**  **// if key is found in hash table, remove it**  **if (i != hashtable[index].end())**  **hashtable[index].erase(i);**  **}**    **// display the hash table**  **void Hashing::displayHash() {**  **for (int i = 0; i < hash\_bucket; i++) {**  **cout << i;**  **for (auto x : hashtable[i])**  **cout <<" > " << x;**  **cout << endl;**  **}**  **}**  **// main program**  **int main() {**  **// array that contains keys to be mapped**  **int hash\_array[] = {11,12,21, 14, 15};**  **int n = sizeof(hash\_array)/sizeof(hash\_array[0]);**    **Hashing h(7); // Number of buckets = 7**    **//insert the keys into the hash table**  **for (int i = 0; i< n; i++)**  **h.insert\_key(hash\_array[i]);**  **// display the Hash table**  **cout<<"Hash table created:"<<endl;**  **h.displayHash();**    **// delete 12 from hash table**  **h.delete\_key(12);**    **// display the Hash table**  **cout<<"Hash table after deletion of key 12:"<<endl;**  **h.displayHash();**    **return 0;**  **}** |
| --- |

**Output:**

**Hash table created:**

**0 –> 21 –> 14**

**1 –> 15**

**2**

**3**

**4 –> 11**

**5 –> 12**

**6**

**Hash table after deletion of key 12:**

**0 –> 21 –> 14**

**1 –> 15**

**2**

**3**

**4 –> 11**

**5**

**6**