

## Lab Hash Table

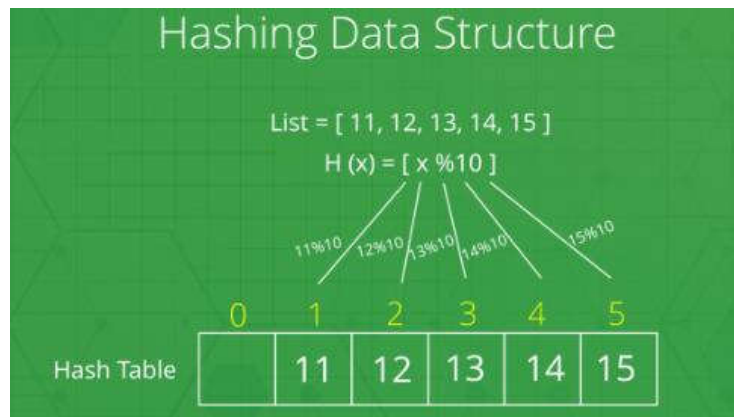
CLO:  
02,03

### Objectives:

In this section we will attempt to go one step further by building a data structure that can be searched in  $O(1)$  time. This concept is referred to as hashing.

### Hashing:

Hashing is an important Data Structure which is designed to use a special function called the Hash function which is used to map a given value with a particular key for faster access of elements. The efficiency of mapping depends of the efficiency of the hash function used. Let a hash function  $H(x)$  maps the value  $x$  at the index  $x\%10$  in an Array. For example, if the list of values is  $[11,12,13,14,15]$  it will be stored at positions  $\{1,2,3,4,5\}$  in the array or Hash table respectively.



The ideal hash table structure is merely an array of some fixed size, containing the items. A stored item needs to have a data member, called key, that will be used in computing the index value for the item. Key could be an integer, a string, etc

For e.g. a name or Id that is a part of a large employee structure

The size of the array is TableSize. The items that are stored in the hash table are indexed by values from 0 to TableSize - 1. Each key is mapped into some number in the range 0 to TableSize - 1. The mapping is called a hash function.

The hashing algorithm is called the hash function-- probably the term is derived from the idea that the resulting hash value can be thought of as a "mixed up" version of the represented value. In addition to faster data retrieval, hashing is also used to encrypt and decrypt digital signatures (used to authenticate message senders and receivers). The digital signature is transformed with the hash function and then both the hashed value (known as a message-digest) and the signature are sent in separate transmissions to the receiver. Using the same hash function as the sender, the receiver derives a message-digest from the signature and compares it with the message-digest it also received. (They should be the same.)

**Applications:** There are many other applications of hashing, including modern day cryptography hash functions. Some of these applications are listed below:

- Message Digest
- Password Verification
- Data Structures (Programming Languages)
- Compiler Operation

- Rabin-Karp Algorithm
- Linking File name and path together

### **Sample Code**

#### **Hash Function**

```
int hash(const string &key, int tableSize)
{
    int hasVal = 0;
    for (int i = 0; i < key.length(); i++)
        hashVal += key[i];
    return hashVal % tableSize;
}
```

### **Lab Tasks**

1. You have to implement the hash table Data Structure using Chaining.
  - Insert
  - Hashfun
  - Remove
  - Search