1. Hashing : Hashing is a technique or process of mapping keys, values into the hash table by using a hash function. It is done for faster access to elements.

2. Hash Table: it is data structure for storing and retrieving data quickly. Every entry in the hash table is made using hash function

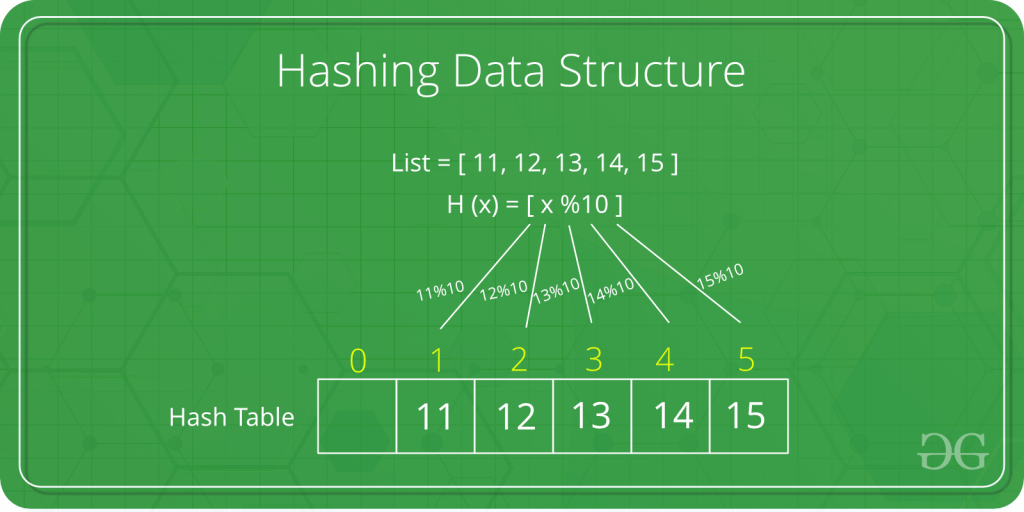
Hash tables support functions that include the following:

* insert (key, value)
* get (key)
* delete (key)

3. Hash Function:

A hash function is any function that can be used to map data of arbitrary size to fixed-size values. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes. The values are usually used to index a fixed-size table called a hash table

Let a hash function H(x) maps the value  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.



4. Bucket : A **bucket** is simply a fast-access location (like an array index) that is the the result of the **hash** function.  Hash table stores entries(key, value) pairs in buckets

Assume you have a housing colony in which there are lots of buildings or apartments and if asked the address of your house, you can answer that I live in XYZ building. So, a building or apartment is a bucket, housing colony is a hash table and your house is an entry.

5. Collision:

It is situation in which hash function returns the same address for more than one record.

The situation where a newly inserted key maps to an already occupied slot in the **hash** table is called **collision**

**Eg. Hash function as key mod 5**

**25, 33, 54 , 55**

6. Probe: Each Calculation of an address and test for success is known as probe

7. Synonym : the set of keys that has to the same location are called synonyms.

Eg, in above hash table 25 and 55 are synonyms.

8. Overflow: When hash table is becomes full and new records to be inserted then it is called overflow.

9. Perfect hash function: a function that maps distinct key elements into hash tables with No Collision

10. Applications:

* Message Digest
* Password Verification
* Data Structures(Programming Languages)
* Compiler Operation
* Rabin-Karp Algorithm
* Linking File name and path together

.

**11. Open or external hashing: When we allow records to be stored in potentially unlimited space, it is called as open or external hashing**

12. **Closed or internal hashing**

When we use fixed space for storage eventually limiting the number of records to be stored, it is called as closed or internal hashing

13 **Load density**

**The maximum storage capacity that is maximum number of records that can be accommodated is called as loading density**

**14. Load factor**

**Load factor is the number of records stored in table divided by maximum capacity of table, expressed in terms of percentage**

15. **Rehashing**

To handle collision, we use strategy to choose a sequence of alternative locations Hash1(Key1), Hash2(Key1), … within the bucket table so as to place the record with Key1