Hash Function, Hash Table creation:

A function that transforms a key into a table index is called a hash function. If h is a hash function and

key is a key, h(key) is called the hash of key and is the index at which a record with the key key should he placed. If r is record whose key hashes into hr, hr is called the hash key of r. The hash function in the preceding example is h(k) = key/ 1000. The values that h produces should cover the entire set of indices in the table. For example, the function x%1000 can produce any integer between 0 to 999, depending on the value of x. It is a good idea for the table size to be somewhat larger than the number of records that are to be inserted.

- A good hash function is one that distributes keys evenly among all slots / index (locations).
- Design of a hash function is an art more than science.

KEY
HASH FUNCTION
INDEX

- Consider key elements as 34, 46, 72, 15, 18, 26, 93
- Hash function is **key mod 5.**
- Index value for the keys are generated using the given hash function
- 34 mod 5 = 4, 34 is stored at index 4.
- 46 mod 5 = 1, 46 is stored at index 1.
- 72 mod 5 = 2 , 72 is stored at index 2.
- 15 mod 5 = 0, 15 is stored at index 0.
- 18 mod 5 = 3, 18 is stored at index 3.

This technique is called **closed hashing. It is shown as below:**

Hash Table

Index / hash	DATA
0	15
1	46
2	72
3	18
4	34

Consider, additional elements 26, 93.

Since, the capacity of the hash table is full, it cannot insert the next element 26 and 93.

Let us say if the capacity is increased, it can accommodate 26 in the hash table.

Now calculate the hash value for 26 = 26 % 5 =1.

This results in collision as the location is not empty. Then search for the first empty location. Say it is at index 5. Now, 26 can be stored in it.

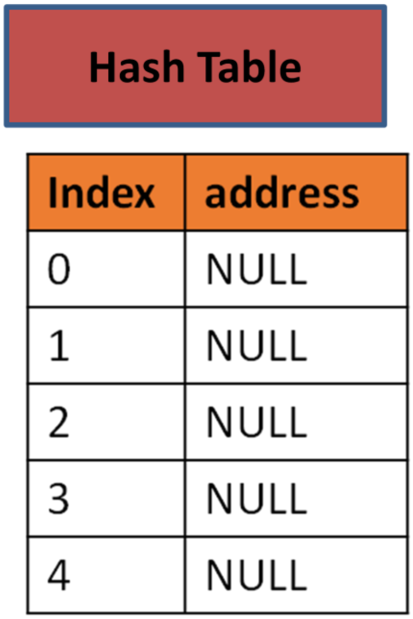
Later for 93, the location 3 is non empty. Searching for the empty location, it can find at location 6. 93 is stored at location with index 6.

Coming to the previous case of only 5 locations, collision occurs. This problem can be resolved by

- Increasing the Memory Capacity.
- Overcoming Collision using
  - Open Addressing / Separate Chaining
  - Closed Addressing :
    - Linear Probing
    - Quadratic Probing
    - Double Hashing

Open Addressing / Separate Chaining :

Initially the hash table contains all NULL values in the address field. This is as shown in the figure below.



Hash Table

Index	address
0	NULL
1	NULL
2	NULL
3	NULL
4	NULL

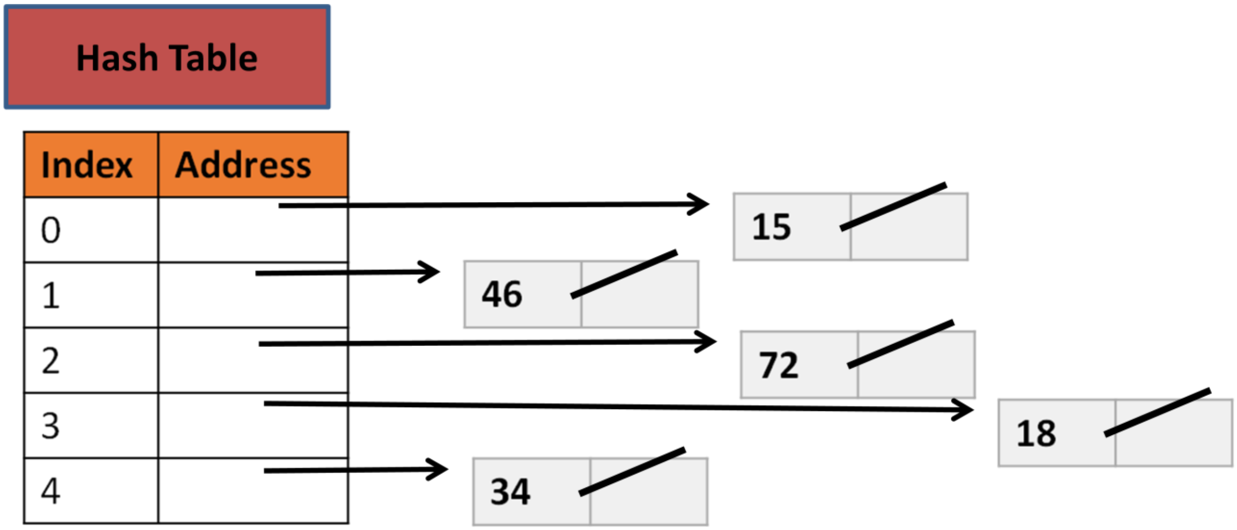
Let us consider again the same key elements as 34, 46, 72, 15, 18. If the hash function is

key mod 5. The series of locations generated is as shown.

- 34 mod 5 = 4, 34 is stored at index 4.
- 46 mod 5 = 1, 46 is stored at index 1.
- 72 mod 5 = 2 , 72 is stored at index 2.

- 15 mod 5 = 0, 15 is stored at index 0.
- 18 mod 5 = 3, 18 is stored at index 3.

The same is as shown below.


Hash Table
Index
Address
0
1
2
3
4
15
46
72
18
34