Hash Map in Python

- A hash map makes use of a hash function to compute an index with a key into an array of buckets or slots.
- A hashmap is a data structure that maps keys to their respective value pairs. It makes it easy to find values that are associated with their keys.
- Its value is mapped to the bucket with the corresponding index.
- The key is unique and immutable.
- Hash function is the core of implementing a hash map.
- It takes in the key and translates it to the index of a bucket in the bucket list.
- Ideal hashing should produce a different index for each key.
- When hashing gives an existing index, we can simply use a bucket for multiple values by appending a list or by rehashing.
- In Python, dictionaries are examples of hash maps.

❖ Advantages of HashMaps

- Fast random memory access through hash functions
- Can use negative and non-integral values to access the values.
- Keys can be stored in sorted order hence can iterate over the maps easily.

❖ Disadvantages of HashMaps

- Collisions can cause large penalties and can blow up the time complexity to linear.
- When the number of keys is large, a single hash function often causes collisions.

❖ Applications of HashMaps

- These have applications in implementations of Cache where memory locations are mapped to small sets.
- They are used to index tuples in Database management systems.
- They are also used in algorithms like the Rabin Karp pattern matching