A hashMap, also known as a hash table or dictionary in some programming languages, is a data structure that stores key-value pairs. It provides a way to map keys to values and allows for efficient retrieval, insertion, and deletion operations.

Here are some key characteristics of hash maps:

1. **Key-Value Pairs**: A hash map stores elements as key-value pairs, where each key is unique within the map and is associated with a corresponding value.
2. **Hashing**: Hash maps use a hashing function to convert keys into indices of an array. This allows for constant-time (O(1)) access to elements based on their keys.
3. **Collision Handling**: Since multiple keys may hash to the same index (known as a collision), hash maps implement collision resolution techniques to handle such cases. Common approaches include chaining (using linked lists or other data structures to store multiple elements at the same index) or open addressing (probing for alternative indices until an empty slot is found).
4. **Dynamic Resizing**: Hash maps typically dynamically resize themselves as the number of elements stored in them increases or decreases. This helps maintain a good balance between space and time complexity for operations.
5. **Efficient Operations**: Hash maps provide efficient operations for insertion, retrieval, and deletion of key-value pairs. On average, these operations have a time complexity of O(1) when the hash function distributes elements evenly across the array.
6. **Unordered**: Hash maps do not guarantee any specific order of elements. The order in which elements are stored and retrieved may vary depending on factors such as hash function, collision resolution strategy, and resizing behavior.
7. **Common Use Cases**: Hash maps are commonly used for implementing associative arrays, symbol tables, caching mechanisms, and in various applications requiring fast lookup operations.