```
// HashMap = A data structure that stores key-value pairs
//          Keys are unique, but Values can be duplicated
//          Does not maintain any order, but is memory efficient
//          HashMap<Key, Value>
```

Method	Description	Example
put(K key, V value)	Adds or updates a key-value pair	map.put("name", "Alice");
get(K key)	Retrieves value for a key	map.get("name");
remove(K key)	Removes key-value pair	map.remove("name");
containsKey(K key)	Checks if key exists	map.containsKey("name");
containsValue(V value)	Checks if value exists	map.containsValue("Alice");
size()	Returns number of entries	map.size();
isEmpty()	Checks if map is empty	map.isEmpty();
clear()	Removes all entries	map.clear();
keySet()	Returns all keys	map.keySet();
values()	Returns all values	map.values();
entrySet()	Returns all key-value pairs	map.entrySet();

```java
package Map;

import java.util.HashMap;

public class HashMap1 {

    public static void main(String[] args) {
        HashMap<String, Integer> scores = new HashMap<>();

        // 2. put() - Add elements
        scores.put("Math", 90);
        scores.put("Science", 85);
        scores.put("English", 88);
        System.out.println("Initial HashMap: " + scores);

        // 3. get() - Access value
        System.out.println("Score in Math: " + scores.get("Math"));

        // 4. containsKey() and containsValue()
        System.out.println("Has Science? " + scores.containsKey("Science"));
        System.out.println("Has score 85? " + scores.containsValue(85));

        // 5. size() and isEmpty()
        System.out.println("Total subjects: " + scores.size());
        System.out.println("Is map empty? " + scores.isEmpty());

        // 6. remove() - Delete an entry
        scores.remove("English");
        System.out.println("After removing English: " + scores);

        // 7. keySet(), values(), entrySet()
        System.out.println("Subjects: " + scores.keySet());
        System.out.println("Scores: " + scores.values());
        System.out.println("Entries: " + scores.entrySet());

        // 8. clear() - Remove all entries
        scores.clear();
        System.out.println("After clearing: " + scores);

}
```
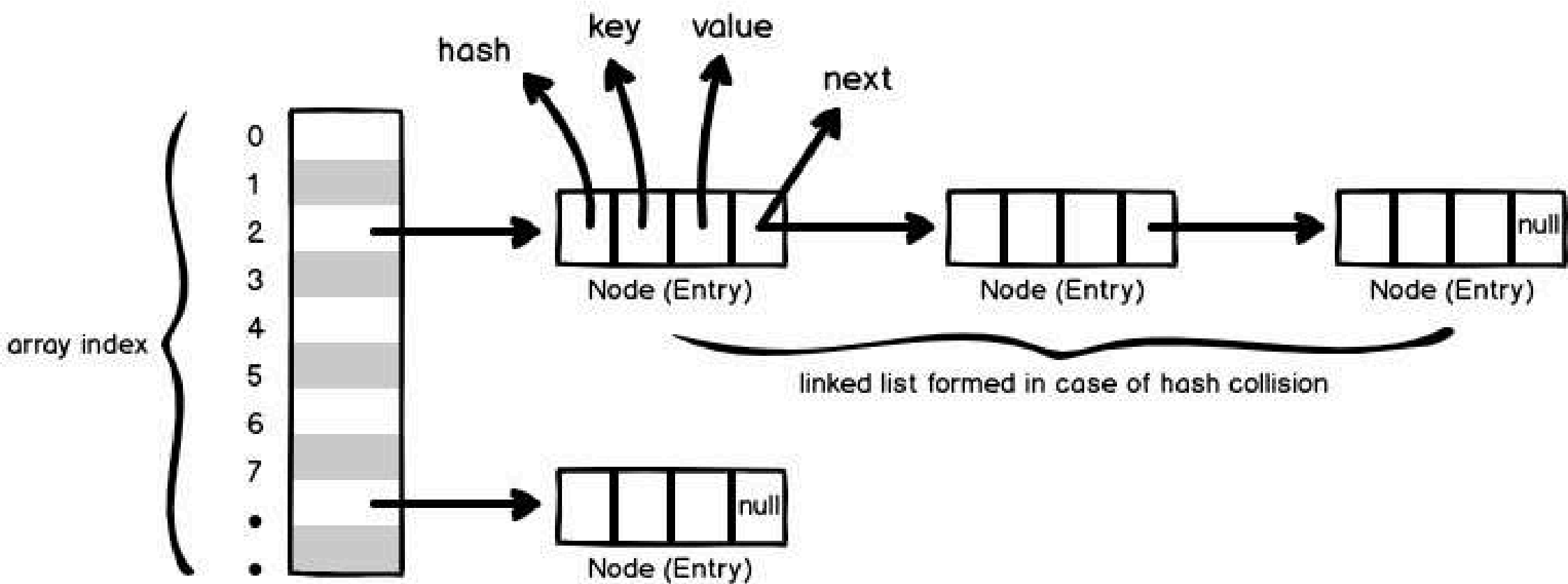
hash   key   value   next

Node (Entry)   Node (Entry)   Node (Entry)   null

linked list formed in case of hash collision

array index

0
1
2
3
4
5
6
7
.
.

Node (Entry)   null

Bucket (array) / Entry table

HashMap