**Internally use Hashing Technique.**

**equals():**

It checks the equality of two objects.

It compares the Key, whether they are equal or not.

It is a method of the Object class. It can be overridden.

If you override the equals() method, then it is mandatory to override the hashCode() method.

**hashCode():**

This is the method of the object class.

It returns the memory reference of the object in integer form.

The value received from the method is used as the bucket number.

The bucket number is the address of the element inside the map. Hash code of null Key is 0.

**Buckets:**

Array of the node is called buckets.

Each node has a data structure like a LinkedList.

More than one node can share the same bucket.

It may be different in capacity.

* When we create Map Objet then it will create Bucket.
* Initial capacity of map is 16 then Default Bucket structure is 16 means 0 – 15.
* This all bucket is consider as a LikedList.
* This LinkedList contains Node.
* Node has a **Hash Key Value next**
* Now when call put(K, V) method it will calculate hashCode using key and then calculate bucket index value using hashcode & (length - 1) for this hasValue.
* Now it will put this **hashValue Key Value next** in calculated bucket index.
* Now next is null because no new node is insert in bucket yet.
* If next map object put method is call and this object calculated index of bucket is same as present one then hashing collision happens.
* Collision means if same bucket having multiple nodes.
* Now when Collision happens it will call internally equals() method.
* It will check and compare present node key and current node key is same or not using equals() method.
* If it is different then immediately it will store this current node in same bucket.
* Now present reference node next value having this new node reference.
* If Key as Null then it will go with zeroth bucket.

**What is enhancement of HashMap in Java 8?**

This HashMap is use bucket as a LinkedList but in certain threshold it will convert as balance tree mechanism.

In Java 8 they use Balance tree mechanism instead of LinkedList in certain threshold.

**What is Fail-Fast and Fail-Safe?**

**Fail-fast** is like throw ConcurrentModificationException while the two threads are try to modifying simultaneously then it will throw the exception.

**Fail-safe** is like working on clone of its collection and it can’t throw any exception.

What is callable interface?

Comparable vs Comparator and which will be use?

What Contract in Equals() and HashCode() method?

What is ConcurrentHashMap?

What is ConucrrentLinkedQueue?

What is Marker Interface and Inbuilt Marker Interface?

Spring Bean LifeCycle and Scope?(init(), destroy())

What is Wrapper Class??

What is Singleton and Write Singleton class and also how to break and prevent Singleton Class?

Why did you use @RestController and why not @Controller?

What is Load Balancer?