**Task 6:**

**1.What is the purpose of the initial capacity and load factor parameters for each Collection class?**

**What are their default values?**

The instance of HashMap has 2 important parameters initial capacity and load factor, the initial capacity defines the capacity of the hash table created ,Load factor is the number of **buckets** in hash table( the hash table uses hash function to convert index into array of buckets or slots , basically a bucket contains pairs of keys and values), the load factor is a measure to calculate the size of the hash table , whenever the entries exceed the product of load factor and initial capacity increases the table needs to be rehashed and the capacity of the table increases.

The default load factor is 0.75 that is 75% of current map size.

Default capacity of Collection varies like for array list it is 10, hashSet and hashMap it is 16.

**2. What is the difference between HashMap and Concurrent HashMap ?**

The Concurrent is an alternative to Hash Table, the differences are Hash map is not synchronized whereas Concurrent Hash Map is synchronized , hash map is not thread safe whereas Concurrent hash map is thread safe, Concurrent hash maps are supported in multi threaded and concurrent environment , whereas hash map is not . Concurrent hash maps are more scalable and performance is more when compared to hash maps.

**3. Arrange the following in the ascending order (performance):**

**HashMap , Hashtable , ConcurrentHashMap and SynchronizedMap.**

HashMap , Hash table, Synchronized Map, Concurrent Hash Map

**4. What is Blocking Queue in Java Collections Framework?**

It is an interface in collection framework , it is a queue which additionally supports operations like maintaining some space in queue while receiving elements and maintain the queue non-empty when retrieving an element.

The operations are:

**Insert-** add elements to the queue

**Remove-** removes elements from the queue

**Display🡪** Displays the elements in queue🡪peek()

This queue do no accepts null values, throws NullPointerException whenever we try to add null values or offer them, Elements can be added without blocking means the capacity of the queue increases , it is used for producer consumer queues, these implementations are thread safe

**Methods :**

**Add—**boolean add(E e)

This method adds elements in the queue, it returns true if an element is added and returns an IllegalStateexception when element is not added as there is no space in the queue

**Offer—**boolean offer(E e)

This method adds elements in the queue, it returns true if an element is added and returns false when the element is not added , in a capacity restricted queue this method is used than add method as this method return a Boolean value rather than throwing an exception

**Put---**void put(E e) throws InterruptedException

This method inserts the specified element in the queue , this method waits until there is space in the queue

**Offer**--- boolean offer([E](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/BlockingQueue.html) e, long timeout, [TimeUnit](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/TimeUnit.html) unit)

throws [InterruptedException](https://docs.oracle.com/javase/7/docs/api/java/lang/InterruptedException.html)

This method inserts the specified element in the queue, It is waited for a specified time if necessary until space is available.

**Take-** [E](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/BlockingQueue.html) take() throws [InterruptedException](https://docs.oracle.com/javase/7/docs/api/java/lang/InterruptedException.html)

This method is used to retrieve or remove the element from the queue, waits until the element is available to remove.

**5. What is IdentityHashMap ?**

This class implements the Map , serializable and clone interfaces extends abstract map class.

This class is used by the user when they should compare the elements via reference.