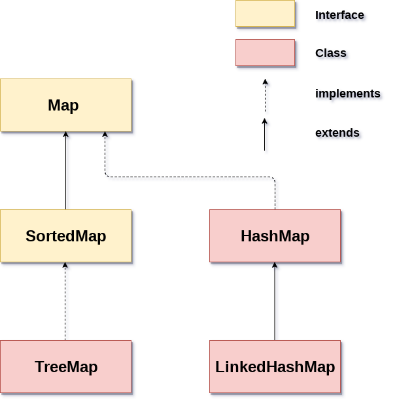
4TH FEBRUARY 2021

MAPS:

* Contains values based on key i.e., key and value pair. Each key and value pair is known as an entry.
* Contains unique keys.
* Used to search, update, or delete values on basis of keys.
* Two interfaces to implements Map in java: Map and SortedMap.
* Three classes: HashMap, LinkedHash Map, and TreeMap.
* Map allows heterogeneous elements



* Map cannot have duplicate keys but can have duplicate values.
* HashMap and Linked HashMap allow null keys and values, but Tree Map does not allow any null key and value.
* HashMap & HashTable follows Random order. But Hashtable methods are Synchronized. But HashMap is not Synchronized.
* LinkedHashMap follows insertion order. It is the implementation of Map and inherits HashMap class.
* HASHMAP:
  + Class implements the Map Interface which allows us to store key and value pair, where keys should be unique.
  + Has no order.
  + Put() method inserts the elements in the map
  + Map.Entry interface contains the getKey() and getValue()Methods. entrySet() method of Map interface to get the instance of Map.Entry.



* ITERATOR: An interator is an interface used for iterating over a collection. It is used in the place of Enumeration in Java Collections Framework. Iterator can traverse both legacy and non-legacy elements. But enumeration traverse only legacy elements.

INTERVIEW QUESTIONS:

1. Hashing : Hashing is a process of converting an object into a integer value. The integer value helps in indexing and faster searches.
2. 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 hashCode() method.
3. 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

* PACKAGES:
  + Packages contains set of related classes.
  + In otherwords, it is a mechanism to encapsulate a group of classes, subpackages and interfaces.
  + Two types of packages :
    - Built-in Packages (Packages from Java API)
    - User-defined Packages



* ACCESS SPECIFIERS :

1. Private: accessible within the class where they are defined, cannot access outside the package.
2. Public : accessible from any class
3. Protected: accessible within the package and outside the package through child class, it cannot be accessed from outside the package.
4. Default : when no access specifier is specified, cannot access outside package.