20.3

1. **Collection** is the interface, it is also a data structure, an object that store the references of the other objects.
2. **Collections** is a Class that can implements the interface collection.
3. **Comparator** is a interface. Comparator provide compare() method to sort elements. Comparator found in java.util package.
4. **List**  an interface. An ordered collection that can contain duplicate elements. List can be implemented by using Linklist, Arraylist and vector classes.
5. **Load Factor** is the measure of how full the Hash table is allowed to get before its capacity is automatically increased.
6. **Collision** is a situation when two or more data elements in the data set U. maps to the same location in the Hash table ,
7. **Hash Map** is a class. It is not threat safe and synchronized. It fast and allow one null key.

20.4

1. **add** method add a new element at the end of the vector.
2. **set** method replace the element at the specified position in the vector with specified element.
3. **remove** method remove the first occurrence of the specified element in the vector.
4. **removeAllElements** methodremoves all element in the vector and set it size to zero.
5. **removeElementAt** remove the element at the specified index.
6. **firstElement** returned the first element at index 0 in the vector.
7. **lastElement** returned the last element of the vector.
8. **contains** returns true if the vector contains the specified elements.
9. **indexOf** gives the index of first occurrence of the specified element in the vector otherwise return -1.
10. **size** gives the number of element in the vector.
11. **capacity** return the current capacity of this vector.

20.7

1. **SET Map**

Set is an un-ordered collection In map we use to store data

which does not allow duplicate. in key and value pairs, we may

We can iterate the values by have duplicates values but no

calling iterator() method duplicate keys.

In map we don’t have iterator() method, but we can get the key by calling method key set()

1. **W**e can not add the primitive type in the collection because collection only store the reference of the objects . Primitive type don’t have objects.
2. Yes we can print all the elements in a collection without using an Iterator. We can use enhance for loop to print all elements of the collection.

20.8

1. **Iterator** can do remove operation only on elements
2. **hasNext** returns true if the iteration has more element.
3. **next** gives the next element in the iteration.

20.9

1. **put** associate the specified value with the specified key
2. **get** returned the value according to the specified key
3. **isEmpty** return true if map contain no key-value mapping.
4. **containKey** return true if map contain mapping for the specified key.
5. **keySet** return the set of keys in the map.

20.10

1. true
2. true
3. true

20.11