1. //What is the collection hierarchy in java language?

**Solution:**

**Collection(I)**

**List(I)** **Set(I)**   **Queue**

**ArrayList(C) LinkedList(C) Vector(C) HashSet SortedSet LinkedList**  **Priority Queue**

**LinkedHashSet TreeSet**

The LinkedList class extends AbstractSequentialList and implements the List interface. It provides a linked-list data structure.

**//2. Demonstrate the creation of collection Array-list, Linked-List and Vector by adding string objects, integer objects and a custom object of your choice and perform add, remove, clear, get with index, get with object, contains, looping through, for the list created with custom object with both iterator, list-iterator and for-each.**

**//2. Demonstrate the creation of collection HashSet, LinkedHashset and TreeSet for strings, integers, custom objects (Employee or Student) and perform add, remove, clear, contains, looping through for-each and iterator**

**//3. Create a program which contains a list1 with values 1,2,3,4,5 and list2 with 3,4,5,6,7 and I would like to print the output 1,2,6,7. So you will have to print out all the numbers which are not common to both the lists.**

**//4. Create a program which contains a list1 with values 1,2,3,4,5 and list2 with 4,5,6,7 and I would like to print the output 1,2,3,4,5,6,7 which are unique values.**

**//5. Create a list and add employee objects with firstname, lastname, middlename and salary and sort the list by salary**

**//6.Sort the same list you created above by firstname**

**//7.Sort the list you created at 5 in the opposite order.**

**//8. write a program to copy of list of integers to another list.**

**//9. Write a program to find the length of longest string in a arraylist**

**//10. Write a program to find the largest number in a arraylist ?**

//11.What is the difference between an arraylist and a LinkedList and when should I choose one over the other?

**Solution:**

|  |  |
| --- | --- |
| **ArrayList** | **LinkedList** |
| 1. Array list is an array based collection. 2. Accessing by index is faster. 3. Addition and removals are slower. | 1. Linked List is a node based collection. 2. Index retrievals are linear hence slower. 3. Addition and removals are faster |

//12. What is the difference between a ArrayList vs Vector?

Solution: The differences between the arraylist and vector are, vector is thread safe whereas arraylist is not. Vector grows by 10 every time, whereas the arraylist grows 1.5 times the current size.

//13. For a contains operation which data structure performs better when comparing a ArrayList vs HashSet?

Solution: ArrayList allows duplicate values in it therefore contains methods has O(N) time complexity whereas HashSet does not allow duplicate values therefore its contains method has an O(1) time complexity.

//14. What is the difference between a Set Vs List and explain when do we get a concurrent modification exception?

Solution:

List maintains insertion order and it does allows duplicates whereas Set does not maintain insertion order and it does not allow duplicates.

When we use fail fast iterators such as for-each and iterator to add an element into a data structure we get concurrent modification exception.

//15. How does Set data-structure work in java?

Solution: Set does not allow duplicates. First it calls the hash code method to check if two objects have the same hash code. If the objects have the same hash code it calls the equals method to make sure they are really equal. It the objects has different hashcode, it treats them as different objects and does not call the equals method.

//16. What is the internal data structure for a ArrayList, HashSet, TreeSet, HashMap and LinkedHashMap?

Solution: TreeSet internally makes use of SortedSet. HashSet internally makes use of HashMap. HashMap internally makes use of an array and linked list.

//17. What is the difference between a HashSet and LinkedHashSet?

Solution: HashSet does not maintain insertion order, whereas LinkedHashSet maintains insertion order, underneath it uses linked list to maintain the order.

//18. When do we use a TreeSet?

Solution: When we want to sort the contents of the data structure we use the TreeSet Data structure.

**//19. Show with a comparator and Comparable example for a TreeSet?**

**//20. How do we sort the data structure arraylist? show with an example?**

**Solution: using the method Collection.sort(ArrayList), for an example please refer to LargestInteger.java**

//21. How do we sort the arraylist in a reverse order?

Solution: Using Collections.sort(list, Collesctions.reverseorder());

//22. How do you synchronize HashMap and ArrayList?

Solution: for HashMap we use Collections.synchronizedmap(map);

for ArrayList we use Collections.synchronizedlist(list);

**//23. How do you copy items from one arraylist to the other?**

**Solution: List<String>b = new ArrayList<String>(a);**

//24. How does HashMap internally works and what is the default size of hashmap and how does it grow in size?

Solution: HashMap uses array and linked list internally, default size of the HashMap is 16 and it grows by 2 times every time it is 75% full.

//25. What is the difference between a HashMap vs HashTable ?

Solution: HashMap is non synchronized and not thread safe, while HashTable is synchronized and thread safe. HashMap allows one null key and any number of null values, while HashTable do not allow null keys and null values in the HashTable object. HashMap object values are iterated by using iterator. HashTable is the only class other than vector which uses enumerator to iterate the values of HashTable object.

**//26. Demonstrate the usage of list-iterator in Arraylist?**

**Solution: Please check the program** CollectionSetDemo.java

**//27. Demonstrate how HashMap prevents duplicate Keys to come-in to the data structure with an example- create a custom class and have the class object as the key, show how do we iterate and print key-value pairs as well?**

**Solution: Please refer to** HashMapDemo.java class in org.jnit.collections.maps package

**28.Demonstrate with an example of having a list data structure as value in a hashmap ?**

//29. Which performs better HashMap, HashTable and Concurrent HashMap?

Solution: HashMap allows one null key and any number of null values, HashTable does not allow null keys or null values and it is synchronized. In a multiple threaded environment an HashMap is better and faster than HashTable and Concurrent HashMap and HashTable because only one thread can access the code at the same time.

Synchronized HashMap is better than the HashTable because lock is at the bucket level unlike HashTable where lock is at object level. And in case of synchronized HashMap and Concurrent HashMap, Concurrent is better because lock is only on a certain portion of the map, hence multiple threads can access different non-synchronized portion of the map.

//30. What do you mean by Hash-collision in HashMap and what happens in the scenario for insertions and retrievals?

Solution: In an HashMap, if the keys are same according to the hashcode() method of the comparable interface but different according to equals method (both are overridden methods.), then it is said to be Hash-collision, and when hash-collision occurs based on a process called linear probing, values with same keys(hashcodes) are placed in the same entry of the hash map in the form of single linked list and while retrieving equals method is called to distinguish the different keys with same hashcodes.

//31. What is the difference between a stack and queue?

Solution: Stack is a Last-In-First-Out whereas Queue is First-In-Last-Out

//32. What are the implementations of a queue?

Solution: 1) LinkedList 2) Priority Queue

//33. Which methods we use to add, remove elements in a stack and queue data structures?

Solution: In Stack we use PUSH and POP to add and remove the elements, In case of Queue we use ADD, OFFER to add and REMOVE and POLL for removing the elements from the Queue data structure.

//34. What is a Deque?

Solution: Deque is a Double ended Queue, which means the elements can be added or removed from both the ends. It can be implemented using two ways

**//35. Demonstrate an example for TreeMap to show TreeMap sort entries by keys?**

**//36. Demonstrate with an example of how we create a generic class?**

**//37. Demonstrate with an example of how can we create a generic method which can take any argument?**

38. What is the difference between a Bounded vs unbounded wildcard?

//39. final List<String> someList = new ArrayList<> ();

The list is declared final, can we add elements to this list or make any modifications?

Solution: Yes, you can Add, remove and update elements.

//40. What is the difference between a Queue and Deque?

Solution: In Deque elements can be added and removed from both ends but in case of Queue elements can be added and removed from only one end.

//41. What do you mean by priority queue?

Solution: Priority Queue is a normal queue with priorities attached to each element, element with the high priority are inserted before elements with low priority.

**NOTE: For jdbc assignments please refer to jdbc2016 project**