Short Answer:

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail (Provide examples to each question).

1. Describe the Collections Type Hierarchy. What Are the Main Interfaces, and What Are the Differences Between Them?
2. What are List interface implementations and what are the differences between them and when to use what?
3. What are Queue interface implementations and what are the differences and when to use what?
4. What are Set interface implementations and what are the differences and when to use what?
5. Explain the structure of the Deque implementation of LinkedList and its usage.
6. What are the differences between HashMap, LinkedHashMap and TreeMap?
7. What is the hashCode() and equals() function?
8. How Is HashMap Implemented in Java? How Does Its Implementation Use hashCode and equals Methods of Objects? What Is the Time Complexity of Putting and Getting an Element from Such Structure?
9. What is Comparable and Comparator interface? What are differences between them and how to use them?
10. What is Functional Interface? How do you create your own Functional Interface?
11. What is Lambda Expression? Why does Lambda Expression work so closely with Functional Interface?

Coding Questions:

Write code in Java to solve following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code. (Note: Please send your testing cases with the code)

1. Change Java PriorityQueue to a maximum PriorityQueue.

2. Use the given ListNode structure below to solve below questions.

```
class ListNode {
    int val;
    ListNode next;
    ListNode() {}
    ListNode(int val) { this.val = val; }
    ListNode(int val, ListNode next) { this.val = val;
this.next = next; }
}
```

a. Merge two sorted LinkedList

```
ListNode merge(ListNode list1, ListNode list2)
```

b. Remove ListNode with odd value from a LinkedList

```
ListNode removeOdd(ListNode head)
```

3. Design and implement your own generic HashMap which should support linked list structure (cannot use LinkedList from Java Collections, must implement your own linked list) for each slot in the bucket. Please at least implement put() and get() method.