**Answers to Questions and Exercises:**

**Questions**

1. Question: At the beginning of this lesson, you learned that the core collection interfaces are organized into two distinct inheritance trees. One interface in particular is not considered to be a true Collection, and therefore sits at the top of its own tree. What is the name of this interface?   
   Answer: Map
2. Question: Each interface in the collections framework is declared with the <E> syntax, which tells you that it is generic. When you declare a Collection instance, what is the advantage of specifying the type of objects that it will contain?   
   Answer: Specifying the type allows the compiler to verify (at compile time) that the type of object you put into the collection is correct, thus reducing errors at runtime.
3. Question: What interface represents a collection that does not allow duplicate elements?   
   Answer: Set
4. Question: What interface forms the root of the collections hierarchy?   
   Answer: Collection
5. Question: What interface represents an ordered collection that may contain duplicate elements?   
   Answer: List
6. Question: What interface represents a collection that holds elements prior to processing?   
   Answer: Queue
7. Question: What interface repesents a type that maps keys to values?   
   Answer: Map
8. Question: What interface represents a double-ended queue?   
   Answer: Deque
9. Question: Name three different ways to iterate over the elements of a List.   
   Answer: You can iterate over a List using streams, the enhanced for statement, or iterators.
10. Question: True or False: Aggregate operations are mutative operations that modify the underlying collection.   
    Answer: False. Aggregate operations do not mutate the underlying collection. In fact, you must be careful to never mutate a collection while invoking its aggregate operations. Doing so could lead to concurrency problems should the stream be changed to a parallel stream at some point in the future.

**Exercises**

1. Exercise: Write a program that prints its arguments in random order. Do not make a copy of the argument array. Demonstrate how to print out the elements using both streams and the traditional enhanced for statement.   
   Answer:
2. import java.util.\*;
3. public class Ran {
4. public static void main(String[] args) {
6. // Get and shuffle the list of arguments
7. List<String> argList = Arrays.asList(args);
8. Collections.shuffle(argList);
9. // Print out the elements using JDK 8 Streams
10. argList.stream()
11. .forEach(e->System.out.format("%s ",e));
12. // Print out the elements using for-each
13. for (String arg: argList) {
14. System.out.format("%s ", arg);
15. }
16. System.out.println();
17. }
18. }
19. Exercise: Take the [FindDups](https://docs.oracle.com/javase/tutorial/collections/interfaces/examples/FindDups.java" \t "_blank)example and modify it to use a SortedSet instead of a Set. Specify a Comparator so that case is ignored when sorting and identifying set elements.   
    Answer:
20. import java.util.\*;
21. public class FindDups {
22. public static void main(String[] args) {
23. Set<String> s = new HashSet<String>();
24. for (String a : args)
25. s.add(a);
26. System.out.println(s.size() + " distinct words: " + s);
27. }
28. }
29. Exercise: Write a method that takes a List<String> and applies [String.trim](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html" \l "trim--" \t "_blank) to each element.   
    Answer:   
    The enhanced for statement does not allow you to modify the List. Using an instance of the Iterator class allows you to delete elements, but not replace an existing element or add a new one. That leaves ListIterator:
30. import java.util.\*;
31. public class ListTrim {
32. static void listTrim(List<String> strings) {
33. for (ListIterator<String> lit = strings.listIterator(); lit.hasNext(); ) {
34. lit.set(lit.next().trim());
35. }
36. }
37. public static void main(String[] args) {
38. List<String> l = Arrays.asList(" red ", " white ", " blue ");
39. listTrim(l);
40. for (String s : l) {
41. System.out.format("\"%s\"%n", s);
42. }
43. }
44. }
45. Exercise: Consider the four core interfaces, Set, List, Queue, and Map. For each of the following four assignments, specify which of the four core interfaces is best-suited, and explain how to use it to implement the assignment.   
    Answers:
    * Whimsical Toys Inc (WTI) needs to record the names of all its employees. Every month, an employee will be chosen at random from these records to receive a free toy.  
      Use a List. Choose a random employee by picking a number between 0 and size()-1.
    * WTI has decided that each new product will be named after an employee — but only first names will be used, and each name will be used only once. Prepare a list of unique first names.  
      Use a Set. Collections that implement this interface don't allow the same element to be entered more than once.
    * WTI decides that it only wants to use the most popular names for its toys. Count up the number of employees who have each first name.  
      Use a Map, where the keys are first names, and each value is a count of the number of employees with that first name.
    * WTI acquires season tickets for the local lacrosse team, to be shared by employees. Create a waiting list for this popular sport.  
      Use a Queue. Invoke add() to add employees to the waiting list, and remove() to remove them.