1. Overriding equals and also sorting using Comparators following best practices (i.e., comparisons  
should be consistent with equals) (Lab 8 Prob2.)

You should be ready to override hashCode()  
whenever you override equals() and understand why we need to override them in different  
cases(Lab 7 Prob 1 Parts B-D).

8. The Java 8 features of interfaces (static and default methods) and best practices for using them.

Diamond problem

4. Write code that handles a situation in which one of the lambdas in a stream pipeline needs to  
throw a checked exception but cannot because the functional interface it implements does not  
permit an exception to be thrown (use one of the techniques mentioned in Lesson 11; see  
Problems 5 in Lab 11).

5. Know how to create JUnit test and best practice for unit testing stream pipelines.

7. Create the most general possible method (a “generic method”) to solve some problem (Like  
Problem 4 in Lab10).

9. SCI Principle

6. Use the reduce method on Streams to solve a problem.

2. Solve a problem using a stream pipeline; then generalize your solution to a Lambda Library  
element. Finally, replace lambdas with inner classes. (Like the quiz)

3. Given a lambda expression, find an appropriate type for it, name it with a (typed) variable,  
rewrite it as a method reference, state which type of method reference it is, and finally, rewrite  
the lambda expression as an inner class that implements the functional interface that represents  
the lambda expression. (Like Lab 8, Problem 6)