Final Exam Review Points

During the final exam, you will not have access to notes, internet, lecture slides, labs, but you will have two cheat sheets (included in this review directory).

In addition to an SCI question at the end, the exam will consist of 6 programming questions selected from the following areas:

1. Overriding equals and also sorting using Comparators (must use comparing and thenComparing), following best practices (i.e., comparisons should be consistent with equals) . You should be ready to override hashCode() whenever you override equals()
2. Solve problems using stream pipelines; then generalize your solutions to Lambda Library elements. 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)
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 10; see Problems 3 and 4 in Lab 10).
5. Use the reduce method on Streams to solve a problem.
6. Create the most general possible method (a “generic method”) to solve some problem (like finding max element of a list, finding second largest element of a list).
7. The Java 8 features of interfaces (static and default methods) and best practices for using them.

Stream API operations that you should know:

of
iterate
limit
skip
concat
filter
map
flatMap
reduce
distinct
sorted (should also know Comparator.comparing and Comparator's thenComparing)
reversed
count
max (need to know how the Optional class works)
findFirst

findAny
ofNullable (be able to use ofNullable together with orElse/orElseGet)
joining, joining(", ")
collect(Collectors.toList())
mapToInt (you should know how to work with IntStream, and to be able to convert to Stream with boxed)

Functional Interfaces you should know well (not available on the Cheat Sheet)

Function
Predicate
Consumer
Comparator