MPP Final Exam Review Points

The midterm will consist of 40 points. Short Answer questions may require English explanations and discussion, but may also require UML and Java code.

Bring a pencil, eraser, and a well-rested, well-nourished, coherent, alert brain

Lesson 7

1. Java 8 Interfaces enhancements -- default and static methods
2. Usage of Interface default methods to design for polymorphism
3. Java 8 Enum capabilities – basic enum usage plus use as a Singleton
4. Java 8 resolution of the Diamond problem and method clash between interfaces and superclasses
5. Rules for overriding equals and hashCode

Skills:

- Create a class diagram solving a design problem with both superclasses and interfaces
- Pseudocode - show usage of Enums
- Reverse engineer a class diagram from Java code

Lessons 8 & 9

6. Definition of functional interface, functor, and closure.
7. Create your own functional interface
8. Define a lambda using a functional interface
9. Solve a problem using a stream pipeline
10. Given a lambda expression use a method reference in place of it
11. Usage of basic stream operations
    a. How to create a Stream
    b. How to use filters
    c. How we limit and skip and concat streams
    d. How to use map
    e. How to use stateful operations -- sort and distinct
    f. How to use terminal operations – count, collect, and reduce
12. Primitive type streams
13. Generalize your solution to a lambda Library

Skills

- Write code to create a stream, apply filters, map, sort, etc. and collect the results
- Given a lambda expression create a lambda library element
- Code your own functional interface and use it to type a lambda expression

Lesson 10

14. Principles for unit testing lambdas – simple and complex
15. How to create your own annotations
16. Threads and race conditions
17. Parallel Streams considerations

Skills

- Write a JUnit test for a simple and a complex lambda expression
- Replacing a lambda expression with a method reference
- Building a JUnit TestLibraryCompanion with auxiliary methods
- Making a thread safe singleton

Lesson 11

18. Create a generic method to solve a problem (like finding max element of a list, finding second largest element of a list).
19. Be able to create a generic solution (generic classes/ generic methods) to avoid downcasting
20. From a generic interface create;
    a. A generic implementation
    b. A parametrized type implementation
21. The use of ***extends*** and ***super*** keywords to create a bounded type variable, e.g. <T extends Comparable>
22. The use of bounded wildcards to create parametrized types
    a. <? Extends T> any subclass of T
    b. <? Super T> any superclass of T
23. Create a hierarchy class diagram to show the relationship between Lists of generic types using bounded wildcards

Skills

- Given a specific solution modify it to be a generic solution
- Given a set of Lists with bounded wildcards create a UML showing their relationship