Review exam topics

Working with Java data types

- Use primitives and wrapper classes, including, operators, parentheses, type promotion and casting
- Handle text using String and StringBuilder classes
- Use local variable type inference, including as lambda parameters

Controlling Program Flow

- Create and use loops, if/else, and switch statements

Java Object-Oriented Approach

- Declare and instantiate Java objects including nested class objects, and explain objects' lifecycles (including creation, dereferencing by reassignment, and garbage collection)
- Define and use fields and methods, including instance, static and overloaded methods
- Initialize objects and their members using instance and static initialiser statements and constructors
- Understand variable scopes, apply encapsulation and make objects immutable
- Create and use subclasses and superclasses, including abstract classes
- Utilize polymorphism and casting to call methods, differentiate object type versus reference type
- Create and use interfaces, identify functional interfaces, and utilize private, static, and default methods
- Create and use enumerations

Exception Handling

- Handle exceptions using try/catch/finally clauses, try-with-resource, and multi-catch statements
- Create and use custom exceptions

Working with Arrays and Collections

- Use generics, including wildcards
- Use a Java array and List, Set, Map and Deque collections, including convenience methods
- Sort collections and arrays using Comparator and Comparable interfaces

Working with Streams and Lambda expressions

- Implement functional interfaces using lambda expressions, including interfaces from the java.util.function package
- Use Java Streams to filter, transform and process data
- Perform decomposition and reduction, including grouping and partitioning on sequential and parallel streams

Java Platform Module System

- Deploy and execute modular applications, including automatic modules
- Declare, use, and expose modules, including the use of services

Concurrency

- Create worker threads using Runnable and Callable, and manage concurrency using an ExecutorService and java.util.concurrent API
- Develop thread-safe code, using different locking mechanisms and java.util.concurrent API

Java I/O API

- Read and write console and file data using I/O Streams
- Implement serialization and deserialization techniques on Java objects
- Handle file system objects using java.nio.file API

Secure Coding in Java SE Application

- Develop code that mitigates security threats such as denial of service, code injection, input validation and ensure data integrity
- Secure resource access including filesystems, manage policies and execute privileged code

Database Applications with JDBC

- Connect to and perform database SQL operations, process query results using JDBC API

Localization

- Implement Localization using Locale, resource bundles, and Java APIs to parse and format messages, dates, and numbers

Annotations

- Create, apply, and process annotations