Java Full Stack Review Questions

Java

Basics

1. What is Java? / Explain some features of Java
2. What is JRE / JDK / JVM?
3. What is the difference between an object and a class?
4. What is the root class from which every class extends?
5. What are the primitive data types in Java?
6. Where are Strings stored?
7. Explain stack vs heap
   1. Are variable references stored on the stack or heap? What about the objects they refer to?
   2. What is a stack frame? When are these created?
8. What are annotations?
9. What is a POJO vs a bean?
10. Can you force garbage collection in Java? When is an object eligible for GC?
11. Why are strings immutable in java? How would you make your own objects immutable?
12. What is the difference between String, StringBuilder, and StringBuffer?
13. What are the different variable scopes in Java?
14. What are the access modifiers in Java? Explain them.
15. What are the non-access modifiers in Java?
16. ‘What is the difference between static and final variables?
17. What are the default values for all data types in Java?
18. What is a wrapper class? List them.
19. What is autoboxing / unboxing?
20. Is Java pass-by-value or pass-by-reference?
21. What makes a class immutable?
22. If two objects are equal, do they have the same hashcode? If not equal?
23. What data types are supported in switch statements?
24. List all non-access modifiers
25. How to pass multiple values with a single parameter into a method?
26. What is a static block?
27. What are static imports?
28. What methods are available in the Object class?
29. What is the difference between == and .equals()?
30. What is an enhanced for loop and what is a forEach loop?
31. What are the 3 usages of “super” keyword?
32. What is the first line of any constructor?
33. How would you perform constructor chaining?
34. What happens if you don’t define a constructor for a class? Can you still instantiate it?

OOP

1. What are the 4 pillars of OOP? Explain each and give examples of you implement them in Java code
2. What is the difference between an abstract class and an interface?
3. Can abstract methods have concrete methods? Can concrete (non-abstract) classes have abstract methods?
4. Can static methods access instance variables? Can non-static methods access static variables?
5. What are the implicit modifiers for interface variables? methods?
6. What is the difference between method overloading and overriding? What are the rules for changing the method signature of overloaded methods?
7. Can you overload / override a main method? static method? a private method? a default method? a protected method?
8. What are covariant return types? What rules apply to return types for overridden methods?
9. When do you use extends or implements keywords?
10. What are enumerations (enums)?
11. What are the implicit modifiers for interface variables / methods?

Collections / Generics

1. What are collections in Java?
2. What are the interfaces in the Collections API?
3. What is the difference between a Set and a List?
4. What is the difference between an Array and an ArrayList?
5. What is the difference between ArrayList and Vector?
6. What is the difference between TreeSet and HashSet?
7. What is the difference between HashTable and HashMap?
8. Are Maps in the Collections API? What makes Map different from other interfaces?
9. List several ways to iterate over a Collection. How would you iterate over a Map?
10. What is the purpose of the Iterable interface? What about Iterator?
11. What is the difference between the Comparable and Comparator interfaces?
12. What are generics? What is the diamond operator (<>)?
13. Create and instantiate a generic class. Create and use a generic method.

Threads

1. What is multi-threading?
2. In what ways can you create a thread?
3. List the methods in the Thread class and Runnable interface
4. Explain the lifecycle of a thread
5. What is deadlock?
6. What is the synchronized keyword?

IO / Serialization

1. How do you serialize / deserialize an object in Java?
2. What is a Marker interface? What does Serializable interface do?
3. What are transient variables?
4. Difference between FileReader and BufferedReader?
5. Explain the try-with-resources syntax
6. List some methods in the Scanner class

Exceptions

1. What is the difference between final, .finalize(), and finally?
2. Explain throw vs throws vs Throwable
3. Do you need a catch block? Can you have more than 1? Is there an order to follow?
4. What is base class of all exceptions? What interface do they all implement?
5. List some checked and unchecked exceptions?
6. Multi-catch block - can you catch more than one exception in a single catch block?

Design patterns

1. What are Singleton / Factory design patterns?
2. How would you create a Singleton?
3. Explain the DAO design pattern

JUnit

1. What is JUnit?
2. What is TDD?
3. What are the annotations in JUnit? Order of execution?
4. Give an example of a test case
5. How would you prevent a test from being run without commenting it out?

Log4j

1. What are the advantages to using a logging library?
2. What is log4j?
3. What are the logging levels of log4j?

Maven

1. What is Maven?
2. What is the default Maven build lifecycle?
3. Where / when does Maven retrieve dependencies from? Where are they stored locally?
4. What is the POM and what is the pom.xml?
5. What defines Maven project coordinates?

Advanced

1. What are functional interfaces? List some that come with the JRE for Java 8
2. What are lambdas?
3. What is try-with-resources? What interface must the resource implement to use this feature?
4. Which collections cannot hold null values?
5. If 2 interfaces have default methods and you implement both, what happens?
6. If 2 interfaces have the same variable names and you implement both, what happens?