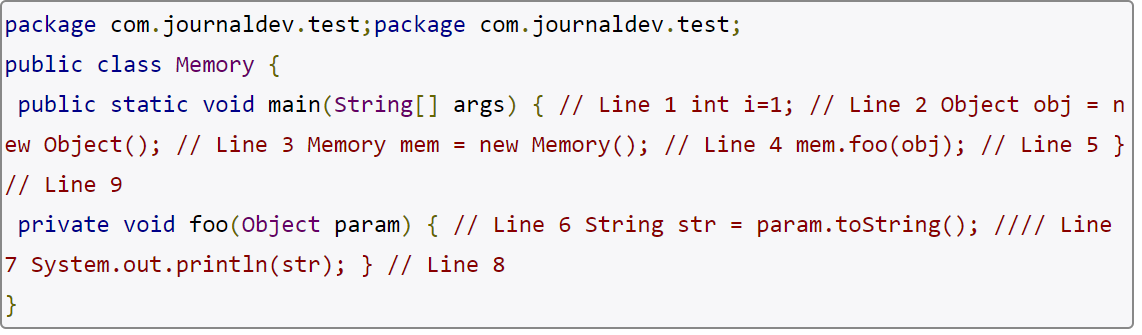
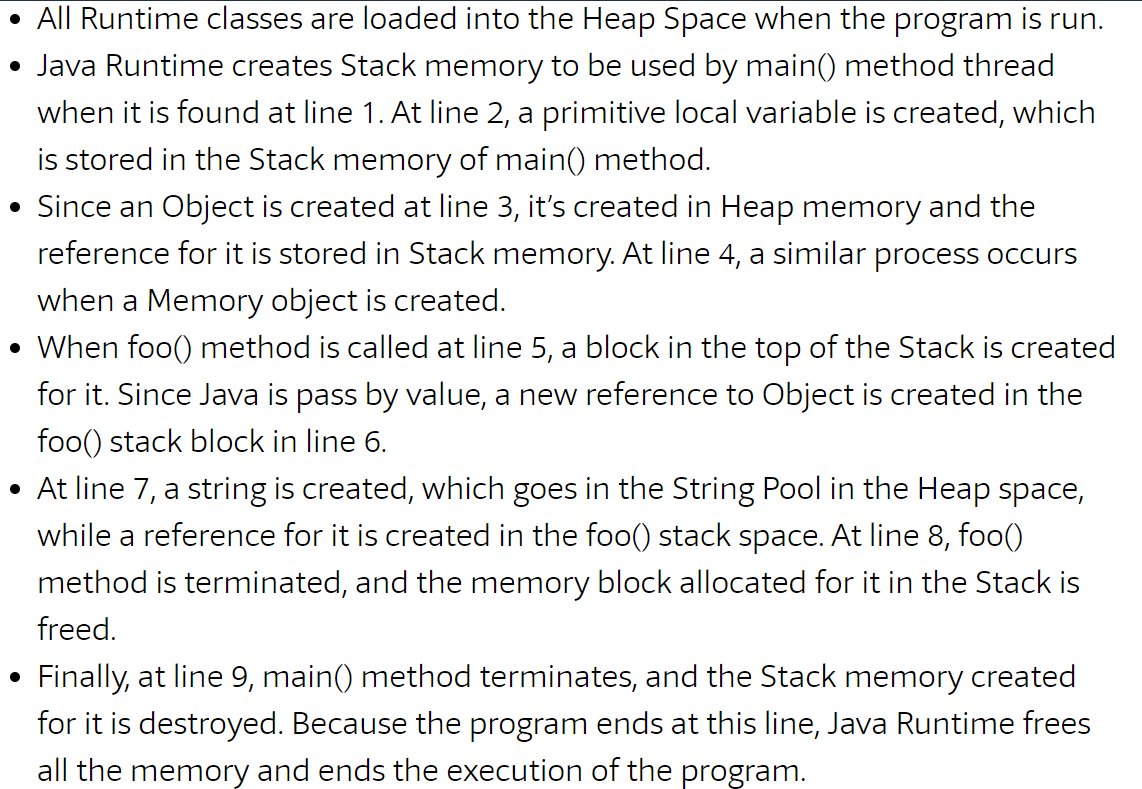
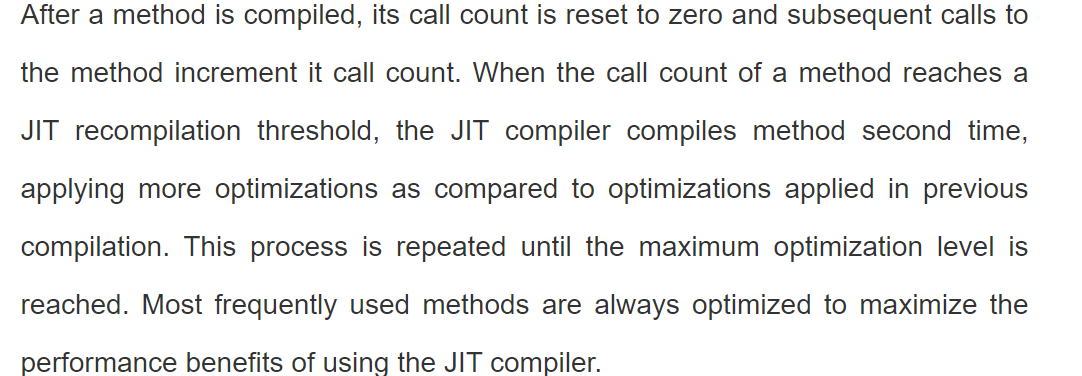
1. Print number sequentially in multiple thread (Concurrency) :  
   (<https://java2blog.com/print-sequence-3-threads-java/>)
2. Avoid If-Else conditions for factory design pattern. (<https://www.baeldung.com/java-replace-if-statements>  
   <https://www.javacodegeeks.com/2014/10/factory-without-if-else.html>)
3. Design pattern  
   (<https://sourcemaking.com/design_patterns/factory_method>  
   <https://refactoring.guru/design-patterns>)
4. Why volatile is needed with singleton in double check mechanism?  
   **Ans :** In case of volatile variable, values being read from main memory rather than cached value. Also, just to avoid CPU re-ordered writes volatile inserts memory barrier which means all write operation happens before any read or modification operation.   
   (<https://www.javacodegeeks.com/2018/03/volatile-java-works-example-volatile-keyword-java.html>)
5. Why there is no atomic float or atomic double in java?  
   **Ans:** Not only java but all languages suffer with issue as its hard to implement compare & swap solution for float/double using assembly languages. This is why many languages provided solution for most commonly used primitive types. Alternatively we can use AtomicReference in java to solve this issue. Also, we can use conversion like floatToIntBits or IntToFloatBits.  
   (<https://stackoverflow.com/questions/5505460/java-is-there-no-atomicfloat-or-atomicdouble>)
6. What is shallow, deep & lazy copy of object? How to manage object cloning for mutable object reference?  
   (<https://www.geeksforgeeks.org/deep-shallow-lazy-copy-java-examples/>  
   <https://howtodoinjava.com/java/cloning/a-guide-to-object-cloning-in-java/>)
7. What is shadowing of static function in java?  
   **Ans :** Basically class type static method will be called not object type.  
   (<https://www.geeksforgeeks.org/g-fact-63/?ref=rp>)
8. Are static local variables allowed in java?  
   **Ans :** No (<https://www.geeksforgeeks.org/g-fact-47/?ref=rp>)
9. How can we assign values to final or static final variables?  
   (<https://www.geeksforgeeks.org/g-fact-55/?ref=rp>)
10. Java memory management  
    (<http://tutorials.jenkov.com/java-concurrency/java-memory-model.html>  
    )
11. Contract between equals() and hashcode() method in java?
12. Prevent breaking a singleton pattern.  
    (<https://dzone.com/articles/prevent-breaking-a-singleton-class-pattern>)
13. More restrictive access in derived class  
    Example : When you open main gate then there is no point of closing other doors whereas when you protect main gate then you can open internal doors. Similarly if base class method is public then derived must be public whereas if base class method is private then derived class method can be anything (even public).  
    (<https://www.geeksforgeeks.org/more-restrictive-access-is-given-to-a-derived-class-method-in-java/?ref=rp>)
14. How can we clone bean or singleton object for another JVM?  
    **Ans :** Serialization
15. How can we do serialization/de-serialization explicitly?  
    **Ans :** By using ObjectOutputStream#writeObject() and ObjectInputStream#readObject() methods.
16. Java Concurrency   
    (<https://howtodoinjava.com/java-concurrency-tutorial/>)
17. How can you detect deadlock in application and how can we resolve?
18. How to analyse thread dump?  
    (<https://dzone.com/articles/how-analyze-java-thread-dumps>)
19. Approach to detect memory leak using MAT  
    (<https://dzone.com/articles/java-thread-retained-memory>)
20. What is MBeans, Jstat, JStack, Jmap, jps in java?
21. Whats is spring profile vs maven profile?
22. How can you create immutable object in java?
23. Type of references in java? Weak, Soft, Phantom, Strong.  
    (<https://www.geeksforgeeks.org/types-references-java/>)
24. Covariance & Contravariance in java?  
    **Contravariance :** ? super T write anything with superclass of type T   
    **Covariance :** ? extends T read anything with superclass of type T   
    (<https://dzone.com/articles/covariance-and-contravariance>  
    )
25. What is covariant return type?  
    The derived class overriding method can return object of sub-class type of base class method return type. Java doesn’t support contravariance for method overloading; it gives compiler error.  
    (<https://www.geeksforgeeks.org/covariant-return-types-java/>)
26. **What are the compatible & incompatible changes of class during serialization/de-serialization process? (must visit this link)**  
    (<https://www.javamadesoeasy.com/2015/02/serialization-top-25-interview.html>)
27. What will happen if one of the member class doesn’t implement serializable interface?  
    **Ans :** NotSerializableException is thrown.
28. What will happen if we used List,Set and Map in class during serialization process.  
    **Ans :** These collection classes implements Serializable interface so it will work as expected. Also, all primitive data types are part of serialization process.
29. How can sub-class avoid serialization if super class implements serializable interface?  
    **Ans :** We need to override writeObject() method in sub-class and throw NotSerializableException() from it.
30. Difference between Externalizable & Serializable :  
    (<https://www.javamadesoeasy.com/2015/07/difference-between-externalizable-and.html>)
31. Types of exceptions   
    Checked or compile time or exception (IO exception, SQL Exception, Broken Barrier)  
    Unchecked or run time or error (null pointer, stack over flow, system failure )
32. Can finally block runs when system.exit() gets called?  
    **Ans :** No.
33. Which class can be used inside try-with-resources block  
    **Ans :** Any class which implements **AutoCloseable** Interface.
34. What will happen when catch and finally block or try and finally block returns value.  
    **Ans :** Ultimately, finally block value will be returned in both the cases.  
    (<https://www.javamadesoeasy.com/2015/05/exceptions-top-60-interview-questions_16.html>)
35. In multi-catch block if sub-class (IOException) mentioned along with super-class (Exception). Is this fine?  
    **Ans :** It throws compile time exception as IOException already been caught by Exception.
36. If method overloaded based on the generic exception and specific exception then which method will be called?  
    **Ans :** Method with specific exception.
37. What is the difference between ClassNotFoundException and NoClassDefFoundException?  
    <https://www.javamadesoeasy.com/2015/12/what-is-difference-between.html>
38. HotSpot JVM Architecture  
    (<https://www.javamadesoeasy.com/2017/03/top-30-jvmjava-virtual-machine.html>)
39. What is functioning of class loader sub-system & runtime data areas of JVM?  
    (<https://www.javamadesoeasy.com/2017/03/top-30-jvmjava-virtual-machine.html>)
40. **Major memory classification of JVM :  
    Ans :** Primitive data type - Stack  
     Address or reference to object - Stack  
     Class - Class area or class loader sub-system  
     Objects - Heap  
     Methods - Method area (neither stack nor heap)  
     Temporary values of method execution - Stack  
     Strings - String pool or pool area in heap  
     Constants - Constant pool in heap  
     Static methods & variables - MetaSpace a new memory area (old permgen inside heap)  
     Threads - Inside heap with its PC (program counter) & stack  
     Heap - Young generation, old generation  
     Program counter registers - Current instruction and next instruction address  
     Method frames or blocks - Stack  
     Native method execution - Native method area  
     Native internal threads - Contains information about native (windows or linux) platform  
     Stack - Last In First Out  
     Below is the overlook :  
      
      
      
    (<https://stackify.com/java-heap-vs-stack/>  **Must Visit)**
41. What is execution engine in JVM?  
    **Ans :** JIT, Garbage Collector & Interpreter   
    (<https://www.javamadesoeasy.com/2017/03/top-30-jvmjava-virtual-machine.html> Q5)
42. How JIT improves performance of Most frequently used methods ?  
      
    
43. Difference between JVM, JDK & JRE?  
    JDK - JVM + JRE (for compilation, development and runtime environment)  
    JRE - RunTime environment + native libraries + other libraries  
    JVM - Platform to run byte code (after conversion of machine code)
44. Different types of Garbage collection algorithm :   
    Ans : Serial GC, Parallel GC, Concurrent Mark Sweep GC, G1 GC (to be set in -XX : \_\_\_ parameter of JVM tuning)
45. Different parameters for JVM tuning  
    (<https://www.javamadesoeasy.com/2017/03/top-30-jvmjava-virtual-machine.html> Q12)
46. How to avoid OutOfMemoryError?  
    **Ans :** By setting appropriate stack space (-Xx:) or heap space (-Xms and -Xmx)
47. How the Java heap memory divided?  
    (<https://stackoverflow.com/questions/1262328/how-is-the-java-memory-pool-divided>)
48. **Coding exercise for comparator & comparable interface**  
    ([https://www.javamadesoeasy.com/search/label/Comparable%20and%20Comparator%20program](https://www.javamadesoeasy.com/search/label/Comparable and Comparator program))
49. How to synchronize arraylist to make it completely thread safe?  
    (<https://www.javamadesoeasy.com/2015/12/how-to-synchronize-arraylist-in-java-to.html>)
50. Can we extend Enum?  
    **Ans :** No
51. Difference between composition, aggregation vs association?
52. Difference between bridge & adapter design pattern?  
    **Ans :** Bridge pattern can only be used before designing or implementation to provide components work independently whereas Adapter provides additional interface to connect incompatible implementation after implementation without any modification in existing functionality.   
    (<https://www.javapedia.net/Design-Patterns/1967> Must Visit)