* 1. What is Java?
  2. What does it mean to be a “pass-by-value” language? How is this different from passing by reference?
  3. What is the difference between the JRE, JDK, and JVM?
  4. What happens during the compilation process?
  5. What are the benefits of using Java?
  6. What is a constructor?
  7. What is the first line of a constructor?
  8. What are the primitive data types?
  9. What is a no args constructor?
  10. What is the default constructor?
  11. Under what circumstances does a class have a default constructor?
  12. What are the scopes of a variable in java?
  13. What are the different access modifiers? What does each one do?
  14. What are the different control statements, and how are they different?
  15. How do you create an Array in java?
  16. What is varargs?
  17. What is the first line of a java application?
  18. What are packages and imports?
  19. What is a static import?
  20. What does the static modifier do?
  21. What are Strings?
  22. What are some string methods?
  23. What is the difference between String, StringBuilder, and StringBuffer?
  24. What is the string pool?
  25. What is the difference between the stack and the heap?
  26. What goes on the stack?
  27. What goes on the heap?
  28. What is an exception?
  29. What is the difference between an exception and an error?
  30. What are the different ways in which we can handle an exception?
  31. What are the differences between a checked exception and an unchecked exception?
  32. How many catch blocks can be used in a try catch? (min and max)?
  33. What does finally do?
  34. Can finally be skipped? How?
  35. How do I create a custom exception?
  36. What is autoboxing?
  37. What is auto-unboxing?
  38. What is a wrapper class?
  39. What are the different wrapper classes?
  40. What is garbage collection?
  41. How do I perform garbage collection?
  42. What is the difference between final, finally, and finalize?
  43. What is the Reflection API?
  44. What is a Lambda expression?
  45. What is a functional interface?
  46. What are the pillars of object oriented programming? Explain them.
  47. What is the difference between an abstract class and an interface?
  48. How many interfaces can a class implement in Java?
  49. How many interfaces can an interface implement in Java?
  50. How many interfaces can interface extend in Java?
  51. Can an abstract class implement an interface in Java?
  52. How many classes can a class extend in Java?
  53. How many abstract classes can a class extend in Java?
  54. What are the different non-access modifiers? What do they do?
  55. What are the differences between FileinputStream, FileReader, and BufferedReader (as well as their counterparts)?
  56. What is the Scanner class?
  57. What are some Scanner methods?
  58. What is serialization and deserialization?
  59. How do I serialize an object?
  60. What is a marker interface? What is an example of a marker interface?
  61. How do I prevent some data from getting serialized?
  62. What is the difference between Collection and Collections?
  63. What are some of the subinterfaces of the Collection interface?
  64. What are the various input/delete/get methods for List, Set, and Queue?
  65. How are the List, Set, and Queue interfaces different from each other?
  66. What is an iterator?
  67. How is an iterator different from a ListIterator?
  68. What are generics? Why use them?
  69. Do generics support primitive data types?
  70. What is the difference between comparator and comparable?
  71. What is the purpose of the Object class?
  72. What are some methods that are inherited from the Object class?
  73. What is the difference between == and .equals?
  74. What is the purpose of hashcode?
  75. What is the Map interface’s relationship to the Collection interface?
  76. Explain the difference between Hashmap and Hashtable.
  77. How do I iterate through a HashMap?
  78. What is multithreading? Why do we use it?
  79. What are the different ways to create a thread? What is different about them?
  80. Why would I choose one method of creating a thread over another?
  81. What are the JVM recognized states of a thread?
  82. What are various methods that threads have?
  83. What is synchronization?
  84. What are the risks of synchronization?
  85. What is deadlock, livelock, and thread starvation?
  86. What is the producer and consumer problem?
  87. What is JUnit?
  88. What are the annotations of JUnit?
  89. What are the different assert methods of JUnit?
  90. How do I create a test case and test suite in JUnit?
  91. What is Maven?
  92. What is the Maven lifecycle?
  93. What is the purpose of the pom.xml?
  94. What is the purpose of using Maven?
  95. What is the difference between a Singleton and Factory?
  96. What is the difference between a Java Bean and a POJO?
  97. What is the difference between implicit and explicit casting?
  98. When can you upcast a variable? Downcast a variable?
  99. What is Git?
  100. What are some of the benefits of using Git?
  101. What is version control?
  102. What are some basic Git commands?
  103. What is the difference between git init and git clone?