1. What is the role for a classloader in Java?
2. What are Wrapper classes?
3. Why do we need Wrapper classes in Java?
4. What are the different ways of creating Wrapper class instances?
5. What are differences in the two ways of creating Wrapper classes?
6. What is auto boxing?
7. What are the advantages of auto boxing?
8. What is casting?
9. What is implicit casting?
10. What is explicit casting?
11. re all String’s immutable?
12. here are String values stored in memory?
13. hy should you be careful about String concatenation(+) operator in loops?
14. ow do you solve above problem?
15. hat are differences between String and StringBuffer?
16. hat are differences between StringBuilder and StringBuffer?
17. an you give examples of different utility methods in String class?
18. hat are the important things to consider when implementing equals method?
19. hat is the Hashcode method used for in Java?
20. xplain inheritance with examples .
21. hat is method overloading?
22. hat is method overriding?
23. an super class reference variable can hold an object of sub class?
24. s multiple inheritance allowed in Java?
25. hat is an interface?
26. ow do you define an interface?
27. ow do you implement an interface?
28. an you explain a few tricky things about interfaces?
29. an you extend an interface?
30. an a class extend multiple interfaces?
31. hat is an abstract class?
32. hen do you use an abstract class?
33. ow do you define an abstract method?
34. ompare abstract class vs interface?
35. hat is a constructor?
36. hat is a default constructor?
37. ill this code compile?
38. ow do you call a super class constructor from a constructor?
39. an a constructor be called directly from a method?
40. s a super class constructor called even when there is no explicit call from a sub class constructor?
41. hat is polymorphism?
42. hat is the use of instanceof operator in Java?
43. hat is coupling?
44. hat is cohesion?
45. hat is encapsulation?
46. hat is an inner class?
47. hat is a static inner class?
48. an you create an inner class inside a method?
49. hat is an anonymous class?
50. hat happens when a variable is marked as volatile?
51. hat is a static variable?
52. an a switch statement be used around a String
53. hat design pattern is used to implement exception handling features in most languages?
54. n what scenarios is code in finally not executed?
55. an you explain the hierarchy of exception handling classes?
56. hat is the difference between error and exception?
57. What is the difference between checked exceptions and unchecked exceptions?
58. How do you throw an exception from a method?
59. What happens when you throw a checked exception from a method?
60. What are the options you have to eliminate compilation errors when handling checked exceptions?
61. How do you create a custom exception?
62. How do you handle multiple exception types with same exception handling block?
63. Can you explain about try with resources?
64. How does try with resources work?
65. Can you explain a few exception handling best practices?
66. What are the default values in an array?
67. How do you loop around an array using enhanced for loop?
68. How do you print the content of an array?
69. How do you compare two arrays?
70. What is an enum?
71. Can you use a switch statement around an enum?
72. What are variable arguments or varargs?
73. What are asserts used for?
74. When should asserts be used?
75. What is garbage collection?
76. Can you explain garbage collection with an example?
77. When is garbage collection run?
78. What are best practices on garbage collection?
79. What are initialization blocks?
80. What is a static initializer?
81. What is an instance initializer block?
82. What is tokenizing?
83. Can you give an example of tokenizing?
84. What is serialization?
85. How do you serialize an object using serializable interface?
86. How do you de-serialize in Java?
87. What do you do if only parts of the object have to be serialized?
88. How do you serialize a hierarchy of objects?
89. Are the constructors in an object invoked when it is de-serialized?
90. Are the values of static variables stored when an object is serialized?134 . Why do we need collections in Java?
91. What are the important interfaces in the collection hierarchy?
92. What are the important methods that are declared in the collection interface?
93. Can you explain briefly about the List interface?
94. Explain about ArrayList with an example?
95. Can an ArrayList have duplicate elements?
96. How do you iterate around an ArrayList using iterator?
97. How do you sort an ArrayList?
98. How do you sort elements in an ArrayList using comparable interface?
99. How do you sort elements in an ArrayList using comparator interface?
100. What is vector class? How is it different from an ArrayList?
101. What is linkedList? What interfaces does it implement? How is it different from an ArrayList?
102. Can you briefly explain about the Set interface?
103. What are the important interfaces related to the Set interface?
104. What is the difference between Set and sortedSet interfaces?
105. Can you give examples of classes that implement the Set interface?
106. What is a HashSet?
107. What is a linkedHashSet? How is different from a HashSet?
108. What is a TreeSet? How is different from a HashSet?
109. Can you give examples of implementations of navigableSet?
110. Explain briefly about Queue interface?
111. What are the important interfaces related to the Queue interface?
112. Explain about the Deque interface?
113. Explain the BlockingQueue interface?
114. What is a priorityQueue?
115. Can you give example implementations of the BlockingQueue interface?
116. Can you briefly explain about the Map interface?
117. What is difference between Map and sortedMap?
118. What is a HashMap?
119. What are the different methods in a Hash Map?
120. What is a TreeMap? How is different from a HashMap?
121. Can you give an example of implementation of navigableMap interface?
122. What are the static methods present in the collections class?
123. What is the difference between synchronized and concurrent collections in Java?
124. Explain about the new concurrent collections in Java?
125. Explain about copyonwrite concurrent collections approach?
126. What is compareandswap approach?
127. What is a lock? How is it different from using synchronized approach?
128. What is initial capacity of a Java collection?
129. What is load factor?
130. When does a Java collection throw UnsupportedOperationException?
131. What is difference between fail-safe and fail-fast iterators?
132. What are atomic operations in Java?
133. What is BlockingQueue in Java?
134. What are Generics?
135. Why do we need Generics? Can you give an example of how Generics make a program more flexible?
136. How do you declare a generic class?
137. What are the restrictions in using generic type that is declared in a class declaration?
138. How can we restrict Generics to a subclass of particular class?
139. How can we restrict Generics to a super class of particular class?
140. Can you give an example of a generic method?
141. What is the need for threads in Java?
142. How do you create a thread?
143. How do you create a thread by extending thread class?
144. How do you create a thread by implementing runnable interface?
145. How do you run a thread in Java?
146. What are the different states of a thread?
147. What is priority of a thread? How do you change the priority of a thread?
148. What is executorservice?
149. Can you give an example for executorservice?
150. Explain different ways of creating executor services .
151. How do you check whether an executionservice task executed successfully?
152. What is callable? How do you execute a callable from executionservice?
153. What is synchronization of threads?
154. Can you give an example of a synchronized block?
155. Can a static method be synchronized?
156. What is the use of join method in threads?
157. Describe a few other important methods in threads?
158. What is a deadlock?
159. What are the important methods in Java for inter-thread communication?
160. What is the use of wait method?
161. What is the use of notify method?
162. What is the use of notifyall method?
163. Can you write a synchronized program with wait and notify methods?
164. What is functional programming?
165. Can you give an example of functional programming?
166. What is a stream?
167. Explain about streams with an example? what are intermediate operations in streams?
168. What are terminal operations in streams?
169. What are method references?
170. What are lambda expressions?
171. Can you give an example of lambda expression?
172. Can you explain the relationship between lambda expression and functional interfaces?
173. What is a predicate?
174. What is the functional interface - function?
175. What is a consumer?
176. Can you give examples of functional interfaces with multiple arguments?