Short Questions

1. What is the purpose of a nested try statement?
2. What are built-in exceptions in Java?
3. Can you override a method that throws an exception with a method that doesn't?
4. What is a custom exception?
5. Can a finally block be skipped?
6. What is a stack trace and how is it useful?
7. What is the difference between 'wait()' and 'sleep()' in Java?
8. How do you stop a thread in Java?
9. What is a daemon thread?
10. What is thread join method in Java?
11. Can we call the start() method twice on a thread?
12. What is the purpose of the volatile keyword in Java?
13. What is thread starvation?
14. What is thread deadlock?
15. How can thread deadlock be avoided?
16. What is the difference between 'notify()' and 'notifyAll()' methods?
17. Explain the concept of exception propagation in Java.
18. What is a synchronized block in Java?
19. What are checked exceptions in Java?
20. How can you handle multiple exceptions in a single catch block?
21. What is race condition in multithreading?
22. What is the difference between IS-A and HAS-A in Java?
23. How is the 'throw' keyword used in Java?
24. What are the two ways to create a thread in Java?
25. Can you have a try block without either catch or finally?
26. How does thread synchronization affect thread performance?
27. What is the primary purpose of the Collections Framework in Java?
28. What is an ArrayList in Java?
29. How does LinkedList differ from ArrayList in Java?
30. What is a HashSet in Java?
31. What is the TreeSet in Java used for?
32. How does PriorityQueue function in Java?
33. What is ArrayDeque in Java?
34. What is the role of an Iterator in Java Collections?

35. What is the advantage of a For-Each loop in Java?
36. What is the Map interface in Java?
37. What is the difference between Map and Collection interfaces in Java?
38. What is the significance of the Comparator interface in Java?
39. Name a method used for sorting in the Collections class.
40. What is the purpose of the Arrays class in Java?
41. How is Dictionary class different from Map in Java?
42. What is the use of Hashtable in Java?
43. Explain the Properties class in Java.
44. What is the Stack class in Java?
45. How is Vector different from ArrayList in Java?
46. What is the use of the StringTokenizer class?
47. Explain the BitSet class in Java.
48. What is the purpose of the Date class in Java?
49. How is Calendar different from Date in Java?
50. What is the Random class used for in Java?
51. Describe the Formatter class in Java.
52. What is the functionality of the Scanner class in Java?
53. How does TreeSet ensure that elements are sorted in Java?
54. What is the use of PriorityQueue in real-world applications?
55. What advantage does LinkedHashSet offer over HashSet?
56. How can ArrayDeque be used as a stack in Java?
57. What method in Iterator allows removal of elements during iteration?
58. Why would one use a LinkedList over an ArrayList for a list implementation?
59. What is the difference between HashSet and TreeSet in terms of ordering?
60. How does ArrayDeque differ from LinkedList when used as a queue?
61. What is the purpose of the clone() method in collection classes?
62. Can a null element be added to a TreeSet in Java?
63. How does PriorityQueue handle ordering if no comparator is provided?
64. What is the difference between Iterator and ListIterator?
65. What is the main difference between HashMap and Hashtable?
66. What does the Comparable interface enforce in Java collections?
67. How does Collections.synchronizedList enhance a List in Java?
68. What is the purpose of Collections.unmodifiableCollection?
69. How is EnumSet different from other Set implementations?
70. How can you convert an array to a List in Java?

71. What does the retainAll() method do in Java collections?
72. What is the benefit of using a LinkedHashMap?
73. How does ConcurrentHashMap differ from HashMap?
74. What is the WeakHashMap class used for?
75. What distinguishes IdentityHashMap from HashMap?
76. What is the use of NavigableMap in Java?
77. What is Swing in GUI programming?
78. How does Swing differ from AWT?
79. What is MVC architecture in Swing?
80. Name a limitation of AWT that Swing overcomes.
81. What is a container in Swing?
82. Describe the Flow Layout in Swing.
83. What is the purpose of the Border Layout?
84. Explain the Grid Layout.
85. What is unique about the Card Layout?
86. How does Grid Bag Layout work?
87. What is the Delegation event model in Swing?
88. Define event listeners in Swing.
89. What are event classes in Swing?
90. How are mouse events handled in Swing?
91. What is an Adapter class in Swing?
92. Define inner classes in the context of Swing.
93. What are Anonymous Inner classes in Swing?
94. Describe a simple Swing application structure.
95. What is the relationship between Applets and HTML?
96. What are security issues concerning Applets?
97. How do Applets differ from Applications in Swing?
98. How can parameters be passed to applets?
99. What is a Swing Applet?
100. How is painting done in Swing?
101. What is a JLabel in Swing?
102. How is an ImageIcon used in Swing?
103. Describe the JTextField component in Swing.
104. What is a JButton in Swing?
105. Explain the JToggleButton in Swing.
106. Describe the JCheckBox in Swing.
107. What is a JRadioButton in Swing?
108. Explain the JTabbedPane component.

109. What is the purpose of JScrollPane in Swing?
110. How does JList function in Swing?
111. Describe the JComboBox in Swing.
112. What are Swing Menus used for?
113. Explain the role of Dialogs in Swing.
114. What is the GridBagLayout in Swing?
115. How does the Event Handling mechanism work in Swing?
116. What is the use of Adapter classes in Swing?
117. How are Inner classes used in Swing?
118. What is the significance of Anonymous Inner classes in event handling?
119. How do you create a simple Swing application?
120. What are the differences between Applets and Applications?
121. How is painting managed in Swing?
122. What are the uses of JLabel and ImageIcon in Swing?
123. What are the Swing Buttons and their uses?
124. Describe the JTabbedPane and its functionality.
125. How do JList and JComboBox differ in functionality?