Short Questions

1. What is meant by Object-Oriented Thinking?
2. Explain the concept of agents and communities in Object-Oriented Thinking.
3. What are messages and methods in object-oriented programming?
4. Define the term "Responsibilities" in the context of Object-Oriented Thinking.
5. Differentiate between Classes and Instances.
6. Discuss the significance of Class Hierarchies in Object-Oriented Programming.
7. Explain the concept of Inheritance in Java.
8. What is Method Binding in Java?
9. Describe Method Overriding and its significance.
10. What are Exceptions in Java? How are they handled?
11. Summarize the key Object-Oriented concepts.
12. List and explain the Java buzzwords.
13. Provide an overview of Java programming language.
14. Explain Data types and their importance in Java.
15. What are Variables and Arrays? How are they used in Java?
16. Discuss operators and expressions in Java.
17. Explain control statements in Java.
18. Introducing classes: Explain the process and importance.
19. Discuss Methods and Classes in Java.
20. How does Java handle String handling?
21. Define the concept of Inheritance.
22. What are the basics of Inheritance?
23. Explain Member Access in Inheritance.
24. Discuss Constructors in the context of Inheritance.

25. How do you create a Multilevel hierarchy in Java?
26. What is the use of the 'super' keyword in Java?
27. Explain the usage of 'final' with inheritance.
28. Discuss Polymorphism and its types.
29. What is ad hoc polymorphism?
30. Define pure polymorphism.
31. Explain method overriding with an example.
32. What are abstract classes in Java?
33. Describe the Object class in Java.
34. Discuss the different forms of inheritance.
35. Explain the benefits of inheritance.
36. Discuss the costs associated with inheritance.
37. What is a Package in Java?
38. How do you define a Package?
39. Explain the concept of CLASSPATH.
40. Discuss Access protection in Java.
41. What is meant by importing packages?
42. Define Interfaces in Java.
43. How do you define an interface in Java?
44. Explain implementing interfaces in Java.
45. Discuss Nested interfaces in Java.
46. How do you apply interfaces in Java?
47. What are variables in interfaces?
48. Can interfaces be extended in Java?
49. What are Stream classes in Java?
50. Differentiate between Byte streams and Character streams.

51. How do you read console input in Java?
52. Discuss writing console output in Java.
53. What is the purpose of the File class in Java?
54. How do you read and write files in Java?
55. Explain Random access file operations.
56. What is the Console class used for?
57. Discuss Serialization in Java.
58. What are Enumerations in Java?
59. Explain auto boxing in Java.
60. Discuss Generics in Java.
61. What are the fundamentals of exception handling?
62. Explain the different types of exceptions in Java.
63. Differentiate between Termination and Resumptive models of exception handling.
64. What are Uncaught exceptions?
65. How do you use try and catch in Java?
66. Discuss multiple catch clauses in Java.
67. What is the purpose of nested try statements?
68. Explain the use of throw, throws, and finally in Java.
69. What are built-in exceptions in Java?
70. How do you create custom exception subclasses in Java?
71. What are the differences between thread-based multitasking and process-based multitasking?
72. Describe the Java thread model.
73. How do you create threads in Java?
74. Discuss thread priorities in Java.
75. Explain synchronizing threads in Java.
76. What is inter-thread communication?

77. What are the benefits of Object-Oriented Programming?
78. Discuss the importance of encapsulation in Java.
79. How does Java support encapsulation?
80. Explain the concept of abstraction in Java.
81. What is the purpose of constructors in Java?
82. Differentiate between method overloading and method overriding.
83. How does Java support multiple inheritance?
84. What are the access modifiers in Java?
85. Explain the concept of method hiding in Java.
86. Discuss the use of interfaces versus abstract classes in Java.
87. What is the purpose of the 'this' keyword in Java?
88. How does Java handle memory management?
89. What are the different types of variables in Java?
90. Discuss the significance of the 'static' keyword in Java.
91. What is the role of the 'final' keyword in Java?
92. How do you create and use packages in Java?
93. Discuss the importance of exception handling in Java.
94. Explain the difference between checked and unchecked exceptions.
95. How do you handle exceptions in a multi-threaded environment?
96. Discuss the advantages of using interfaces in Java.
97. What is the purpose of the 'transient' keyword in Java?
98. How does Java support method overriding?
99. What is the significance of the 'instanceof' operator in Java?
100. Explain the difference between '==', 'equals()', and 'hashCode()' methods in Java.
101. Discuss the purpose of the 'volatile' keyword in Java.
102. How do you implement synchronization in Java?

103. Explain the concept of deadlock in Java.
104. What is the role of the 'assert' keyword in Java?
105. Discuss the importance of garbage collection in Java.
106. How do you handle file I/O errors in Java?
107. Explain the concept of lambda expressions in Java.
108. What are functional interfaces in Java?
109. Discuss the benefits of using streams in Java.
110. How does Java support parallel programming?
111. Explain the concept of method reference in Java.
112. What are default methods in interfaces?
113. Discuss the benefits of using generics in Java.
114. How do you create and use enumerations in Java?
115. What is the purpose of the 'autoboxing' feature in Java?
116. Explain the concept of varargs in Java.
117. What are the benefits of using annotations in Java?
118. Discuss the purpose of the 'try-with-resources' statement in Java.
119. How do you create and use custom exceptions in Java?
120. Explain the concept of thread safety in Java.
121. What are the different ways to achieve inter-thread communication in Java?
122. Discuss the benefits of using immutable objects in Java.
123. Explain the purpose of the 'native' keyword in Java.
124. How do you handle exceptions in Java streams?
125. Discuss the advantages and disadvantages of multithreading in Java.