Long Questions

1. What are the fundamental concepts of exception handling in Java?
2. Explain the different types of exceptions in Java with examples.
3. Describe the termination and resumptive models of exception handling in Java.
4. What are uncaught exceptions and how are they handled in Java?
5. How do you use the try and catch blocks for exception handling in Java?
6. Explain the concept of multiple catch clauses in Java with an example.
7. What are nested try statements and how are they useful in exception handling?
8. When do you use the throw and throws keywords in Java?
9. Describe the purpose and usage of the finally block in Java exception handling.
10. Discuss the built-in exceptions provided by Java and their significance.
11. How can you create your own exception subclasses in Java? Provide an example.
12. Illustrate the scenario where a custom exception subclass might be useful.
13. Explain the differences between thread-based multitasking and process-based multitasking.
14. Describe the Java thread model and its components in detail.
15. Could you provide examples of scenarios where synchronization and inter-thread communication are essential for ensuring the proper functioning of a multithreaded Java application?
16. Create a multithreaded Java application that explores various aspects of thread-based multitasking. Discuss the differences between thread-based multitasking and process-based multitasking and explain the Java thread model. Implement thread creation, set thread priorities, synchronize threads to avoid race conditions, and showcase inter-thread communication techniques such as wait(), notify(), and notifyAll().
17. What is the purpose of the Collections Framework in Java?
18. Name three key interfaces in the Collections Framework.
19. Explain the difference between ArrayList and LinkedList.
20. What is the HashSet class in Java Collections?
21. How does TreeMap differ from TreeSet?

22. Explain the concept of PriorityQueue.
23. What is the purpose of ArrayDeque in Java?
24. How can you access elements in a Collection using an Iterator?
25. Explain the significance of the For-Each loop in Java.
26. What is the role of Map interfaces in the Collections Framework?
27. How do Comparators work in Java Collections?
28. What are Collection algorithms, and give an example.
29. Explain the purpose of the Arrays class in Java.
30. What are Legacy Classes and Interfaces in Java Collections?
31. What is the purpose of the Dictionary class in Java?
32. How does the Properties class differ from Hashtable in Java?
33. Explain the significance of the Stack class in Java Collections.
34. What is the role of the Vector class in Java Collections?
35. Name some additional utility classes in Java Collections.
36. What is the purpose of the String Tokenizer class?
37. Explain the functionality of the BitSet class in Java.
38. What is the purpose of the Date class in Java Collections?
39. How does the Calendar class enhance date and time functionality?
40. What is the initial capacity of an ArrayList in Java?
41. How is a HashSet different from a TreeSet in terms of ordering?
42. What is the time complexity of adding an element to a PriorityQueue?
43. Explain the role of the Iterator's remove() method.
44. What is the default capacity of an ArrayDeque in Java?
45. How does the Arrays.copyOfRange() method work in Java?
46. What is the size of a BitSet in Java when it is created with no initial bits set?
47. In a multi-threaded Java application, you need to store and modify a dynamically changing list of unique product IDs. Which collection from the Java Collections Framework would you choose, and why? How would you ensure thread-safe access to this collection to prevent data corruption?

48. You're designing an algorithm to efficiently find the top 10 frequently occurring words in a large text file. Which collection(s) would you likely use, and how would you structure your code to achieve this effectively?
49. What is Swing in GUI programming, and how does it differ from AWT?
50. Explain the Model-View-Controller (MVC) architecture in the context of GUI programming.
51. What are the limitations of AWT (Abstract Window Toolkit) in GUI programming?
52. Name and describe the common layout managers in Swing.
53. What is the Delegation event model in Java?
54. Explain the terms Event sources and Event listeners in the context of GUI programming.
55. What are Event classes in Java, and how are they related to the Delegation event model?
56. How can mouse and keyboard events be handled in Java GUI programming?
57. What is an Adapter class in the context of event handling in Java?
58. How are Inner classes used in event handling in Java?
59. What is an Anonymous Inner class, and how is it used in event handling?
60. How can parameters be passed to applets in Java?
61. What security issues are associated with Java applets?
62. How do Swing Applets differ from Swing Applications?
63. Explain the process of creating a Swing Applet in Java.
64. How is painting handled in Swing, and can you provide an example of painting in Swing?
65. Explore and describe the JLabel and ImageIcon components in Swing.
66. What is the purpose of the JTextField component in Swing?
67. Explain the functionality of Swing Buttons, including JButton, JToggleButton, JCheckBox, and JRadioButton.
68. What is the role of JTabbedPane in Swing, and how is it used?
69. Describe the purpose of JScrollPane in Swing and its use in handling scrollable components.
70. What is the JList component in Swing, and how is it used?

71. Explain the functionality of JComboBox in Swing and how it differs from JList.

72. What are Swing Menus, and how are they created in Java Swing applications?

73. How can dialog boxes be implemented in Java Swing applications?

74. How many layout managers are provided by Swing, and name them?

75. In a BorderLayout, how many regions (areas) can a container be divided into?

76. If a class A has a method named mouseClicked(MouseEvent e), how does it relate to event handling, and what type of event does it handle?

77. What is the default layout manager for a JPanel in Swing?

78. In a JTabbedPane, if you have three tabs, how do you refer to the second tab programmatically?

79. Create a custom Swing component that extends JPanel and displays a chessboard with interactive squares that change color when clicked. Implement event handling to register and respond to mouse clicks on each square.

80. You're building a user interface for a music player application. Design a Swing layout using a combination of layout managers (e.g., BorderLayout, FlowLayout, GridLayout) to display the following elements: