Long Questions

UNIT - III

1. Write a Java program to establish a database connection using JDBC and execute a simple SQL query.
2. Write a Java program to retrieve data from a database using JDBC and display the results.
3. Write a Java program to insert data into a database table using JDBC prepared statements.
4. Write a Java program to update records in a database table using JDBC statements.
5. Write a Java program to delete records from a database table using JDBC statements.
6. Write a Java program to implement a simple TCP client-server application for exchanging text messages.
7. Write a Java program to implement a simple UDP client-server application for sending and receiving datagrams.
8. Write a Java program to create a simple Java Bean with properties, getters, and setters.
9. Write a Java program to demonstrate the usage of RMI for invoking remote methods between a client and server.
10. Discuss the security considerations involved in JDBC database connections and how they can be addressed.
11. Explain the concept of connection pooling in JDBC. How does connection pooling improve database performance?
12. Describe the error handling mechanisms available in JDBC. How can exceptions be handled effectively in JDBC applications?
13. Discuss the transaction management features provided by JDBC. How are transactions initiated, committed, and rolled back?
14. Explain the significance of networking in distributed systems. How does Java support network communication between distributed applications?
15. Discuss the advantages and disadvantages of using Java Beans in software development projects. How can Java Beans enhance code reusability and maintainability?

UNIT - IV

16. Explain the life cycle of a Java applet and discuss each phase in detail.
17. Write a Java applet code demonstrating the life cycle of an applet.
18. How can images be added to a Java applet? Discuss various techniques and considerations.
19. Write a Java applet code that adds an image to the applet and displays it on the screen.
20. Discuss event handling in Java applets. Explain the concept of event listeners and their implementation.
21. Write a Java applet code demonstrating event handling, such as mouse clicks or keyboard inputs.
22. Discuss the fundamentals of AWT (Abstract Window Toolkit) in Java. Explain its role in creating graphical user interfaces.
23. Write a Java code using AWT to draw shapes and text on a window.
24. Explain how to work with Windows Graphics and Text in Java AWT. Provide examples illustrating their usage.
25. Discuss the usage of AWT controls in Java applications. Explain how to create and manipulate various controls.
26. Write a Java code using AWT controls to create a simple user interface with buttons and text fields.
27. Describe the concept of layout managers in Java AWT. Discuss different layout managers and their characteristics.
28. Write a Java code using AWT layout managers to organize components in a structured layout.
29. Discuss the implementation of menus in Java AWT applications. Explain how to create and manage menus using AWT.
30. Write a Java code using AWT menus to create a menu bar with various options and submenus.
31. Explain the life cycle of a servlet in Java. Discuss each phase in detail.
32. Write a Java servlet code demonstrating the life cycle of a servlet and handling HTTP requests.

33. Describe the Servlet API in Java. Discuss its core components and functionalities.

34. Write a Java servlet code that retrieves data from an HTTP request and generates an appropriate response.

35. Explain how HTTP requests and responses are handled in servlets. Provide examples demonstrating request and response processing.

36. Discuss the usage of cookies in servlets. Explain how cookies are created, sent, and received in servlet-based applications.

37. Write a Java servlet code that utilizes cookies to store and retrieve user-specific data.

38. Discuss various techniques for session tracking in servlets. Discuss the advantages and limitations of each technique.

39. Write a Java servlet code that implements session tracking to maintain user sessions across multiple requests.

40. Introduce the concept of JavaServer Pages (JSP). Discuss their role in web development.

41. Write a simple JavaServer Pages (JSP) code demonstrating the usage of JSP directives and scriptlets.

42. Write a Java applet code that integrates sound and plays it when the applet is loaded.

43. Explain the process of adding sound to a Java applet. Provide examples demonstrating sound integration.

44. Explain the method of passing parameters to a Java applet. Discuss the different approaches and their advantages.

45. Write a Java applet code that accepts parameters and displays them on the applet window.

UNIT - V

46. What is XML (eXtensible Markup Language), and how does it differ from HTML? Explain its significance in web development.

47. Describe the structure of an XML document. What are the key components, and how are they defined?

48. Discuss the purpose and usage of XML namespaces. How do they help in avoiding naming conflicts in XML documents?

49. Explain the role of XML Schema Definition (XSD) in XML validation. How are XML schemas created and applied?

50. Describe the process of transforming XML documents using XSL (eXtensible Stylesheet Language). How does XSLT (XSL Transformations) facilitate XML document transformation?

51. Write an XSLT stylesheet to transform an XML document into a different format, such as HTML or plain text.

52. What are web services, and how do they enable interoperability between different applications over the internet?

53. Discuss the components of a web service architecture, including UDDI (Universal Description, Discovery, and Integration) and WSDL (Web Services Description Language).

54. Explain the purpose of UDDI in web services. How does it facilitate service discovery and registration?

55. Describe the structure and content of a WSDL document. How does WSDL define the interface of a web service?

56. Write a WSDL document defining the interface of a simple web service, including operations, input/output parameters, and message formats.

57. Discuss the various types of web services, including SOAP (Simple Object Access Protocol), RESTful services, and JSON (JavaScript Object Notation) services. How do they differ in terms of architecture and communication protocols?

58. Explain the concept of Java web services. How are web services implemented and consumed using Java technologies?

59. Discuss the steps involved in creating a Java web service using JAX-WS (Java API for XML Web Services). Provide a detailed explanation of each step.

60. Write a Java class representing a simple web service endpoint, including methods annotated with JAX-WS annotations for exposing as web service operations.

61. Describe the process of consuming a web service in Java. How are web service clients generated and invoked?

62. Discuss the advantages and disadvantages of using XML-based web services compared to other alternatives, such as RESTful services.

63. Explain the concept of resource-oriented architecture (ROA) in web services. How does it differ from traditional service-oriented architectures (SOA)?

64. Describe the role of HTTP methods (GET, POST, PUT, DELETE) in RESTful web services. How are they used to perform CRUD operations on resources?

65. Discuss the use of XML as a data format in RESTful web services. How are XML representations of resources exchanged between clients and servers?

66. Write a Java servlet to handle HTTP requests and implement RESTful web service endpoints for performing CRUD operations on a resource.

67. Explain the importance of data validation and error handling in web services. How are errors communicated between clients and servers?

68. Discuss the security considerations in web services, including authentication, authorization, and encryption. How are security measures implemented in XML-based services?

69. Describe the process of securing a Java web service using SOAP message-level security. What are the key components of a secure web service?

70. Discuss the role of JSON (JavaScript Object Notation) in web services. How does JSON facilitate data interchange between clients and servers?

71. Explain the concept of web resources in RESTful architecture. What are the characteristics of a resource, and how are they identified and accessed?

72. Describe the HATEOAS (Hypermedia as the Engine of Application State) principle in RESTful web services. How does it enhance the discoverability and navigability of resources?

73. Discuss the advantages of using XML over other data formats in web services, such as JSON or plain text. In what scenarios is XML preferred?

74. Explain the concept of content negotiation in RESTful web services. How are different representations of a resource negotiated between clients and servers?

75. Write a Java class to represent a resource in a RESTful web service, including attributes and methods for manipulating the resource's state.