**Java Questions:**

1. Write a Java program that uses polymorphism by defining an interface called Shape with methods to calculate the area and perimeter of a shape. Then create classes that implement the Shape interface for different types of shapes, such as circles and triangles.

2. Write a Java program to invoke parent class constructor from a child class. Create Child class object and parent class constructor must be invoked. Demonstrate by writing a program. Also explain key points about Constructor.

3. Write a Java programme that takes an integer from the user and throws an exception if it is negative.Demonstrate Exception handling of same program as solution.

4. Create a Java program that simulates a bank account. The program should allow users to deposit and withdraw money, check their balance.

5. Demonstrate the difference between abstract class and interface by writing programs as well as in keypoints.

6. Write a Java program that uses stream api to perform operations on a large data set, such as sorting or filtering the data.

7. Create a Java program that implements a binary search algorithm. The program should accept user input for the target value and search for it in a sorted array. The program should return the index of the target value if found or a message if not found.

8. Write a Java program that creates two threads. The first thread should print even numbers between 1 and 10, and the second thread should print odd numbers between 1 and 10.

9. Write a Java program that implements a producer-consumer model using multithreading. The program should have a producer thread that generates random numbers and adds them to a queue, and a consumer thread that reads numbers from the queue and calculates their sum. The program should use synchronization to ensure that the queue is accessed by only one thread at a time.

10. Write a Java program that reads a set of integers from the user and stores them in a List. The program should then find the second largest and second smallest elements in the List.

11. Write a Java program that connects to a MySQL database using JDBC. The program should read data from a table and display the results in the console.

12. Write a Java program that uses JDBC to implement a simple CRUD (create, read, update, delete) application. The program should allow users to add, view, update, and delete records in a MySQL database table.

13. Create a Java program that connects to a PostgreSQL database and executes a batch update. The program should read the input data from a file and insert it into the database using JDBC batch updates.

14. Create a Java servlet that reads the name of the user from a form and displays a welcome message on the web page. The servlet should use the GET method to read the input data from the user.

15. Write a Java servlet that reads the data from a MySQL database table and displays it in an HTML table on the web page. The servlet should use JDBC to connect to the database and retrieve the data.

16. Create a Java servlet that uses session management to maintain the state of the user across multiple requests. The servlet should store the user's name in a session object and display it on multiple pages of the web application.

17. Create a web application that lets users create and view blog posts. The web application should use the MVC pattern, with servlets as controllers, JSPs as views, and a database as the model. Users should be able to create new blog posts by filling out a form that includes a title, description, and content. The web application should use a servlet to store the blog post data in the database. Users should also be able to view all the blog posts on a separate page, and the web application should use a servlet to retrieve the blog post data from the database and display it in a formatted way.

18. Create a Java program that uses Hibernate to connect to a MySQL database and retrieve data from a table. The program should use Hibernate to map the table to a Java object and then display the data on the console.

19. Create a Java program that uses Hibernate to insert data into a MySQL database table. The program should use Hibernate to map the table to a Java object and then insert the data into the table. After inserting the data, the program should retrieve it from the database and display it on the console.

20. The program should use Hibernate to map the table to a Java object and then update the data in the table. After updating the data, the program should retrieve it from the database and display it on the console.

21. Create a Spring Boot application that inserts data into a MySQL database table using JPA and Hibernate. The application should use Spring Data JPA to map the table to a Java object and then insert the data into the table.

22. Create a Spring Boot application that uses Spring Data JPA to retrieve data from a database. The application should have entities for users and orders, and should allow for querying orders by user.

23. Create a Spring MVC application that allows users to register and login. The application should have a registration form that accepts user details and a login form that authenticates users.

24. Create a Spring Boot application that uses Spring MVC to create a REST API. The API should accept a JSON request with data and insert it into a MySQL database table using JPA and Hibernate. The application should use Spring Data JPA to map the table to a Java object and then insert the data into the table.

25. Create a Spring Boot application that uses Spring AOP to log method calls. The application should have a service class with methods that perform operations. The application should use Spring AOP to log the method calls with input and output parameters to the console.

26. Create a Spring Boot application that exposes a REST API for managing a list of products. The API should allow for creating, updating, deleting, and retrieving products.

27. Create a Spring Boot application that uses Spring Cloud to register a service with Eureka Server. The application should expose a REST API for retrieving data from a database and the API should be discovered by Eureka Server.

28. Create a Spring Boot application that uses Spring Cloud Config Server to externalize configuration. The application should have a property file that defines properties for database connection and other application settings.

29. Create a Spring Boot application that uses Spring Data JPA to retrieve data from a database and expose it as a REST API. The API should allow for filtering, sorting, and paging.

30. Create a Spring Boot application that uses Spring Cloud Circuit Breaker to handle failures in a REST API. The API should use a circuit breaker pattern to handle timeouts and other errors.