1. What is Spring Boot?

Interview answer:
“Spring Boot is an extension of the Spring framework that simplifies application development by providing auto-configuration, embedded servers, and production-ready features.”

---

2. Why Spring Boot over Spring?

Interview answer:
“Spring Boot reduces boilerplate configuration, eliminates XML, provides embedded servers, and helps create stand-alone applications quickly.”

---

3. What are the main features of Spring Boot?

Interview answer:
“Auto-configuration, starter dependencies, embedded servers, externalized configuration, and production-ready features like Actuator.”

---

4. What is auto-configuration?

Interview answer:
“Auto-configuration automatically configures Spring beans based on classpath dependencies, properties, and existing beans.”

---

5. How does auto-configuration work internally?

Interview answer:
“It works using `@EnableAutoConfiguration`, which loads configuration classes based on conditions like `@ConditionalOnClass` and `@ConditionalOnMissingBean`.”

---

6. What is @SpringBootApplication?

Interview answer:
“@SpringBootApplication is a combination of `@Configuration`, `@EnableAutoConfiguration`, and `@ComponentScan`.”

---

7. What are Spring Boot starters?

Interview answer:
"Starters are predefined dependency bundles that simplify dependency management, such as `spring-boot-starter-web` or `spring-boot-starter-data-jpa`."

---

8. What is an embedded server?

Interview answer:
"An embedded server like Tomcat or Jetty runs inside the application, allowing it to run as a standalone JAR without external deployment."

---

9. Difference between JAR and WAR in Spring Boot?

Interview answer:
"JAR is used for standalone applications with embedded servers, while WAR is used when deploying to an external application server."

---

10. What is application.properties or application.yml?

Interview answer:
"These files are used for externalized configuration such as database settings, server port, and environment-specific properties."

---

11. How does Spring Boot support multiple environments?

Interview answer:
"By using profiles like `application-dev.properties`, `application-test.properties`, and `application-prod.properties`."

---

12. What is Spring Boot Actuator?

Interview answer:
"Actuator provides production-ready endpoints to monitor and manage applications, such as health, metrics, and environment info."

---

13. What is @RestController?

Interview answer:
"@RestController is used to create RESTful web services and returns data directly as JSON or XML."

---

14. How does Spring Boot handle JSON conversion?

Interview answer:
“Spring Boot uses HttpMessageConverters, usually backed by Jackson, to automatically convert Java objects to JSON and vice versa.”

---

15. What is @RequestBody and @ResponseBody?

Interview answer:
“@RequestBody binds JSON request data to Java objects, and @ResponseBody sends Java objects directly as HTTP responses.”

---

16. How does exception handling work in Spring Boot?

Interview answer:
“Using @ExceptionHandler or @ControllerAdvice for global exception handling.”

---

17. What is @ControllerAdvice?

Interview answer:
“@ControllerAdvice is used for centralized exception handling across all controllers.”

---

18. What is Spring Boot DevTools?

Interview answer:
“DevTools provides features like automatic restart and live reload to improve development productivity.”

---

19. How does Spring Boot integrate with databases?

Interview answer:
“Using Spring Data JPA or JDBC with auto-configured DataSource and transaction management.”

---

20. What is Spring Data JPA?

Interview answer:
“Spring Data JPA simplifies data access by providing repository abstractions and reducing boilerplate CRUD code.”

---

21. Difference between Spring Boot and Spring MVC?

Interview answer:
“Spring MVC is a web framework, while Spring Boot is a framework that simplifies configuration and setup of Spring applications.”

---

22. How do you secure a Spring Boot application?

Interview answer:
“By integrating Spring Security for authentication, authorization, and CSRF protection.”

---

23. What is CORS and how is it handled in Spring Boot?

Interview answer:
“CORS allows or restricts cross-origin requests and can be handled using `@CrossOrigin` or global CORS configuration.”

---

24. How do you monitor a Spring Boot application?

Interview answer:
“Using Spring Boot Actuator endpoints and integrating with monitoring tools.”

---

25. How do you package and deploy a Spring Boot application?

Interview answer:
“Spring Boot applications are packaged as executable JARs and can be deployed on any server or cloud platform.”

---

26. Real-time use of Spring Boot in projects?

Interview answer:
“Spring Boot is used to build REST APIs, microservices, backend services, and enterprise applications with minimal configuration.”