**Basic Spring Boot Concepts**

1. **What is Spring Boot?**  
   Spring Boot is a framework that simplifies Spring application development by removing boilerplate code and configuration. It auto-configures everything you need to start a project quickly.
2. **How is Spring Boot different from Spring?**  
   Spring is a core framework; Spring Boot builds on top of it to reduce manual configuration. Spring Boot = Spring + Auto Configuration + Embedded Server + Starter Dependencies.
3. **What are Spring Boot starters?**  
   Starters are pre-configured Maven/Gradle dependencies. Example: spring-boot-starter-web sets up Spring MVC with Tomcat and Jackson by default.
4. **What is auto-configuration in Spring Boot?**  
   It automatically configures beans based on the dependencies present in your classpath. You don’t need to define beans manually unless you want to override something.
5. **What is the purpose of application.properties or application.yml?**  
   It's used to define external configuration like server port, database connection, etc.

**✅ Spring Boot Annotations**

1. **What does @SpringBootApplication do?**  
   It's a combination of @Configuration, @EnableAutoConfiguration, and @ComponentScan. It marks the main class of a Spring Boot app.
2. **What does @RestController mean?**  
   It's a combination of @Controller and @ResponseBody. It tells Spring to return the response directly as JSON/XML.
3. **What is @RequestMapping used for?**  
   It maps HTTP requests to handler methods. You can also use @GetMapping, @PostMapping, etc. as shortcuts.
4. **What is @Component, @Service, and @Repository?**  
   All are stereotype annotations used to create Spring beans.
   * @Component: generic
   * @Service: for business logic
   * @Repository: for persistence logic
5. **What does @Autowired do?**  
   It automatically injects dependencies into Spring-managed beans.

**✅ Embedded Server, Dev Tools, Profiles**

1. **Which embedded servers are supported by Spring Boot?**  
   By default, it uses Tomcat. Others like Jetty and Undertow are also supported.
2. **How do you change the default port in Spring Boot?**  
   Add server.port=8081 (or your port) in application.properties.
3. **What is Spring Boot DevTools?**  
   A module that enables hot reloading, automatic restarts, and live reload of static resources during development.
4. **What are Spring profiles?**  
   Profiles allow you to define different configurations for different environments (dev, test, prod).  
   Use @Profile("dev") to mark a bean for a specific profile.

**✅ JPA, Database, REST**

1. **How do you connect Spring Boot to a database?**  
   Add DB dependencies (spring-boot-starter-data-jpa), configure DB details in application.properties, and define entities/repositories.
2. **What is Spring Data JPA?**  
   It's a part of Spring that reduces boilerplate JPA code. You just create an interface extending JpaRepository and Spring auto-implements the logic.
3. **What is the use of @Entity and @Id?**  
   @Entity marks a class as a JPA entity. @Id marks the primary key of the entity.
4. **What is the difference between CrudRepository and JpaRepository?**  
   JpaRepository extends CrudRepository and provides more features like pagination and sorting.

**✅ Advanced Concepts**

1. **How do you handle exceptions in Spring Boot REST APIs?**  
   Use @ControllerAdvice with @ExceptionHandler to catch and customize exceptions globally.
2. **What are Actuators in Spring Boot?**  
   Spring Boot Actuator provides production-ready features like health check, metrics, and monitoring via endpoints like /actuator/health.