🚀 Java Spring Boot– Well-Structured Notes

🔹 1. What is Spring Boot?

✅ Answer

Spring Boot is a framework built on top of Spring that simplifies application development by providing auto-configuration, embedded servers, and production-ready features.

💻 Example

```java
@SpringBootApplication
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

🔹 2. What are the main features of Spring Boot?

✅ Answer

- Auto Configuration
- Embedded Servers (Tomcat, Jetty)
- Starter Dependencies
- Production Ready Tools (Actuator)
- Minimal Configuration

🔹 3. What is `@SpringBootApplication`?

✅ Answer

It is a combination of three annotations:

- `@Configuration`
- `@EnableAutoConfiguration`
- `@ComponentScan`

💻 Example

```java
@SpringBootApplication
public class App {}
```

🔹 4. What is Auto Configuration?

✅ Answer

Auto configuration automatically configures Spring application based on dependencies present in the classpath.

💻 Example

If MySQL dependency is present, Spring Boot automatically configures datasource.

🔹 5. What is Starter Dependency?

✅ Answer

Starter dependencies provide pre-configured dependencies for specific functionality.

💻 Example

```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

🔹 6. What is Embedded Server?

✅ Answer

Spring Boot comes with built-in servers like Tomcat, so external server deployment is not required.

💻 Example

Run application → Automatically runs on port 8080.

🔹 7. What is `application.properties`?

✅ Answer

It stores configuration settings like database, server port, etc.

💻 Example

```
None
server.port=9090
```

🔹 8. Difference between `application.properties` and `application.yml`?

✅ Answer

- Properties → Key-value format
- YAML → Hierarchical and readable

💻 Example (YAML)

```
None
server:
  port: 9090
```

🔹 9. What is `@RestController`?

✅ Answer

It combines `@Controller` and `@ResponseBody`. It returns JSON or XML data directly.

💻 Example

```
Java
@RestController
public class HelloController {

    @GetMapping("/hello")
    public String hello() {
        return "Hello Spring Boot";
    }
}
```

🔹 10. Difference between `@Controller` and `@RestController`?

✅ Answer

- `@Controller` → Returns View
- `@RestController` → Returns Data (JSON/XML)

🔹 11. What is Dependency Injection?

✅ Answer

It allows objects to get dependencies automatically instead of creating them manually.

💻 Example

```java
@Service
public class MyService {}

@RestController
public class MyController {

    @Autowired
    private MyService service;
}
```

🔹 12. What is `@Autowired`?

✅ Answer

It automatically injects dependency into Spring beans.

🔹 13. What is `@ComponentScan`?

✅ Answer

It scans package and loads Spring beans automatically.

🔹 14. What is `@Bean` Annotation?

✅ Answer

Used to define a bean manually inside configuration class.

💻 Example

```java
@Configuration
public class Config {

    @Bean
    public MyService myService() {
        return new MyService();
    }
}
```

🔹 15. What is Spring Boot Actuator?

✅ Answer

Provides monitoring features like health check, metrics, etc.

💻 Example

```xml
spring-boot-starter-actuator
```

Access:

```
/actuator/health
```

🔹 16. What is `@Value` Annotation?

✅ Answer

Used to read values from properties file.

💻 Example

```java
@Value("${server.port}")
private String port;
```

🔹 17. What is `@ConfigurationProperties`?

✅ Answer

Used to map multiple properties into a class.

💻 Example

```java
@ConfigurationProperties(prefix="app")
public class AppConfig {
    private String name;
}
```

🔹 18. What is `CommandLineRunner`?

✅ Answer

Runs code after application startup.

💻 Example

```java
@Component
public class MyRunner implements CommandLineRunner {

    public void run(String... args) {
        System.out.println("App Started");
    }
}
```

🔹 19. What is Spring Boot DevTools?

✅ Answer

Provides automatic restart and live reload.

🔹 20. How to change default port?

✅ Answer

```
None
server.port=9091
```

🔹 21. What is Spring Data JPA?

✅ Answer

It simplifies database operations using repository interfaces.

💻 Example

```
Java
@Repository
public interface UserRepo extends JpaRepository<User, Long> {}
```

🔹 22. What is `@Entity`?

✅ Answer

Marks class as database table.

💻 Example

```java
@Entity
public class User {
    @Id
    private Long id;
}
```

🔹 23. What is `@Repository`?

✅ Answer

Indicates DAO layer and handles database exceptions.

🔹 24. What is `@Service`?

✅ Answer

Defines business logic layer.

🔹 25. What is `@RequestMapping`?

✅ Answer

Maps HTTP request to controller method.

💻 Example

```java
@RequestMapping("/home")
```

🔹 26. Difference between `@GetMapping` and `@PostMapping`?

✅ Answer

- `@GetMapping` → Fetch data
- `@PostMapping` → Send data

🔹 27. What is Exception Handling in Spring Boot?

✅ Answer

Handled using `@ControllerAdvice` and `@ExceptionHandler`.

💻 Example

```java
@ControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(Exception.class)
    public String handleException() {
        return "Error occurred";
    }
}
```

🔹 28. What is Profiles in Spring Boot?

✅ Answer

Used to run application in different environments like dev, test, prod.

💻 Example

```
None
spring.profiles.active=dev
```

🔹 29. What is Spring Boot Security?

✅ Answer

Provides authentication and authorization.

💻 Example

```
XML
spring-boot-starter-security
```

🔹 30. What is Microservices and how Spring Boot supports it?

✅ Answer

Microservices is architecture where application is divided into small independent services. Spring Boot supports it using REST APIs, Spring Cloud, and Service Discovery.

💻 Example

```
Java
@RestController
public class OrderService {

    @GetMapping("/orders")
    public String getOrders() {
        return "Order List";
    }
}
```
