Explore More

Subcription : Premium CDAC NOTES & MATERIAL @99

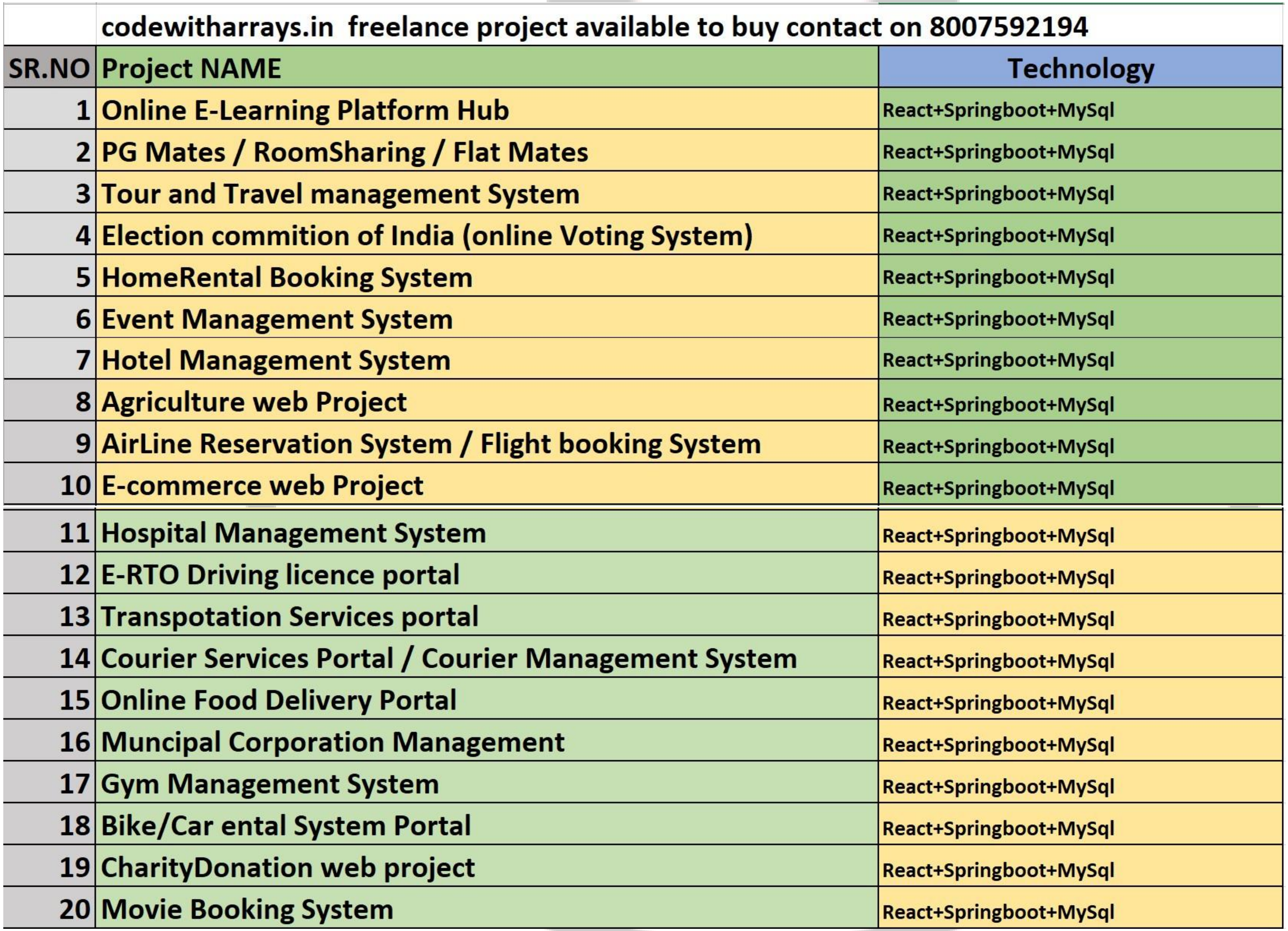
Contact to Join Premium Group

Click to Join Telegram Group

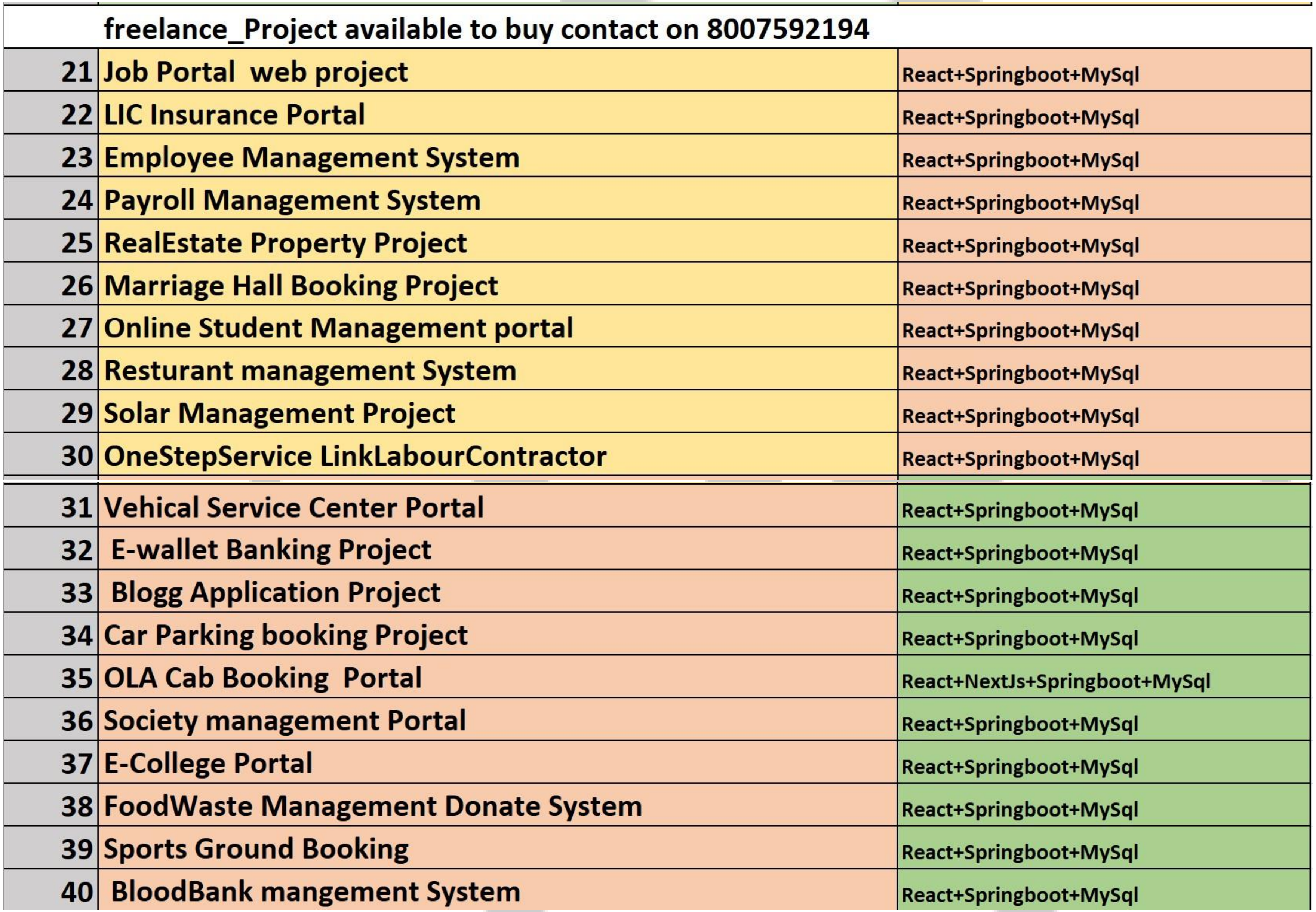
For More E-Notes

Join Our Community to stay Updated

**TAP ON THE ICONS TO JOIN!**



|  | codewitharrays.in  freelance project available to buy contact on 8007592194 |  |
|---|---|---|
| SR.NO | Project NAME | Technology |
| 1 | Online E-Learning Platform Hub | React+Springboot+MySql |
| 2 | PG Mates / RoomSharing / Flat Mates | React+Springboot+MySql |
| 3 | Tour and Travel management System | React+Springboot+MySql |
| 4 | Election commition of India (online Voting System) | React+Springboot+MySql |
| 5 | HomeRental Booking System | React+Springboot+MySql |
| 6 | Event Management System | React+Springboot+MySql |
| 7 | Hotel Management System | React+Springboot+MySql |
| 8 | Agriculture web Project | React+Springboot+MySql |
| 9 | AirLine Reservation System / Flight booking System | React+Springboot+MySql |
| 10 | E-commerce web Project | React+Springboot+MySql |
| 11 | Hospital Management System | React+Springboot+MySql |
| 12 | E-RTO Driving licence portal | React+Springboot+MySql |
| 13 | Transpotation Services portal | React+Springboot+MySql |
| 14 | Courier Services Portal / Courier Management System | React+Springboot+MySql |
| 15 | Online Food Delivery Portal | React+Springboot+MySql |
| 16 | Muncipal Corporation Management | React+Springboot+MySql |
| 17 | Gym Management System | React+Springboot+MySql |
| 18 | Bike/Car ental System Portal | React+Springboot+MySql |
| 19 | CharityDonation web project | React+Springboot+MySql |
| 20 | Movie Booking System | React+Springboot+MySql |



| | freelance_Project available to buy contact on 8007592194 | |
|---|---|---|
| 21 | Job Portal web project | React+Springboot+MySql |
| 22 | LIC Insurance Portal | React+Springboot+MySql |
| 23 | Employee Management System | React+Springboot+MySql |
| 24 | Payroll Management System | React+Springboot+MySql |
| 25 | RealEstate Property Project | React+Springboot+MySql |
| 26 | Marriage Hall Booking Project | React+Springboot+MySql |
| 27 | Online Student Management portal | React+Springboot+MySql |
| 28 | Resturant management System | React+Springboot+MySql |
| 29 | Solar Management Project | React+Springboot+MySql |
| 30 | OneStepService LinkLabourContractor | React+Springboot+MySql |
| 31 | Vehical Service Center Portal | React+Springboot+MySql |
| 32 | E-wallet Banking Project | React+Springboot+MySql |
| 33 | Blogg Application Project | React+Springboot+MySql |
| 34 | Car Parking booking Project | React+Springboot+MySql |
| 35 | OLA Cab Booking Portal | React+NextJs+Springboot+MySql |
| 36 | Society management Portal | React+Springboot+MySql |
| 37 | E-College Portal | React+Springboot+MySql |
| 38 | FoodWaste Management Donate System | React+Springboot+MySql |
| 39 | Sports Ground Booking | React+Springboot+MySql |
| 40 | BloodBank mangement System | React+Springboot+MySql |

| | | |
|---|---|---|
| 41 | Bus Tickit Booking Project | React+Springboot+MySql |
| 42 | Fruite Delivery Project | React+Springboot+MySql |
| 43 | Woodworks Bed Shop | React+Springboot+MySql |
| 44 | Online Dairy Product sell Project | React+Springboot+MySql |
| 45 | Online E-Pharma medicine sell Project | React+Springboot+MySql |
| 46 | FarmerMarketplace Web Project | React+Springboot+MySql |
| 47 | Online Cloth Store Project | React+Springboot+MySql |
| 48 | Train Ticket Booking Project | React+Springboot+MySql |
| 49 | Quizz Application Project | JSP+Springboot+MySql |
| 50 | Hotel Room Booking Project | React+Springboot+MySql |

| | | |
|---|---|---|
| 51 | Online Crime Reporting Portal Project | React+Springboot+MySql |
| 52 | Online Child Adoption Portal Project | React+Springboot+MySql |
| 53 | online Pizza Delivery System Project | React+Springboot+MySql |
| 54 | Online Social Complaint Portal Project | React+Springboot+MySql |
| 55 | Electric Vehical management system Project | React+Springboot+MySql |
| 56 | Online mess / Tiffin management System Project | React+Springboot+MySql |
| 57 | | React+Springboot+MySql |
| 58 | | React+Springboot+MySql |
| 59 | | React+Springboot+MySql |
| 60 | | React+Springboot+MySql |

# Spring Boot + React JS + MySQL Project List

| Sr.No | Project Name | YouTube Link |
|---|---|---|
| 1 | Online E-Learning Hub Platform Project | https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW |
| 2 | PG Mate / Room sharing/Flat sharing | https://youtu.be/4P9cIHg3wvk?si=4uEsi0962CG6Xodp |
| 3 | Tour and Travel System Project Version 1.0 | https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12 |
| 4 | Marriage Hall  Booking | https://youtu.be/VXz0kZQi5to?si=llOS-QG3TpAFP5k7 |
| 5 | Ecommerce Shopping project | https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq |
| 6 | Bike Rental System Project | https://youtu.be/FIzsAmIBCbk?si=7ujQTJqEgkQ8ju2H |
| 7 | Multi-Restaurant management system | https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB |
| 8 | Hospital management system Project | https://youtu.be/IynIouBZvY4?si=CXzQs3BsRkjKhZCw |
| 9 | Municipal Corporation system Project | https://youtu.be/cVMx9NVyI4I?si=qX0oQt-GT-LR_5jF |
| 10 | Tour and Travel System Project version 2.0 | https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ |

| Sr.No | Project Name | YouTube Link |
|---|---|---|
| 11 | Tour and Travel System Project version 3.0 | https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug |
| 12 | Gym Management system Project | https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX |
| 13 | Online Driving License system Project | https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn |
| 14 | Online Flight Booking system Project | https://youtu.be/m755rOwdk8U?si=HURvAY2VnizIyJlh |
| 15 | Employee management system project | https://youtu.be/lD1iE3W_GRw?si=Y_jv1xV_BljhrD0H |
| 16 | Online student school or college portal | https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD |
| 17 | Online movie booking system project | https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm |
| 18 | Online Pizza Delivery system project | https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM |
| 19 | Online Crime Reporting system Project | https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO |
| 20 | Online Children Adoption Project | https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N |

# Spring MCQ with Answers Explained

### 1. What is the primary purpose of the Spring Core framework?

A) To handle web requests and responses.
B) To configure and managing beans (objects) within the IoC container.
C) To manage database connections with Spring project.
D) To replace Core Java application with Spring framework.

**Answer: B)** To configure and managing beans (objects) within the IoC container.
Explanation: The Core container provides the fundamental functionality of the Spring Framework, including configuring and managing beans (objects) within the IoC.

### 2. What is the role of the ApplicationContext in Spring?

A) To provide advanced features to manage and configure beans.
B) To allow beans to be configured with various data sources such as XML, Java annotations, and Java code.
C) To automatically discover and wire beans using component scanning.
D) All of the above.

**Answer: D)** All of the above.
Explanation: ApplicationContext is responsible for all activities provided in all three options.

### 3. Which of the following is used to inject dependencies in Spring?

A) @Autowired
B) @Inject
C) @Resource
D) All of the above

**Answer: D)** All of the above
Explanation: @Autowired, @Inject (JSR-330), and @Resource (JSR-250) are all used for dependency injection in Spring.

### 4. What will be the result of the following code snippet when doWork() is called?

```
@Component
public class MyService {
   @Autowired
   private MyRepository repository;


   public void doWork() {
      System.out.println("Repository: " + repository);
   }
}


@Repository
public class MyRepository {
}
```

Assuming the Spring configuration is properly set up to scan the package:

<context:component-scan base-package="com.example"/>

A) Repository: null
B) Repository: com.example.MyRepository@someHashCode
C) Compilation error
D) Runtime error

**Answer: B)** Repository: com.example.MyRepository@someHashCode
Explanation: Spring will automatically scan and instantiate the MyRepository bean and inject it into MyService due to the @Autowired annotation.

### 5. What does the following Spring configuration achieve?

```
@Configuration
public class AppConfig {

    @Bean
    public MyBean myBean() {
        return new MyBean();
    }
}
```

A) Declares a MyBean instance as a Spring bean
B) Declares MyBean as a singleton bean
C) Declares MyBean as a prototype bean
D) Both A and B

**Answer: D)** Both A and B
Explanation: The @Bean annotation in a @Configuration class declares a Spring bean, and by default, it is a singleton.

### 6. What will be the output when the method printProperty() is called in the following code?

```
@Component
public class MyComponent {

   @Value("${my.property}")
   private String myProperty;


   public void printProperty() {
      System.out.println("Property: " + myProperty);
   }
}
```

Assuming the property is set in an application.properties file:

```
my.property=HelloSpring
```

A) Property: null
B) Property: HelloSpring
C) Property: ${my.property}
D) Runtime error

**Answer: B)** Property: HelloSpring
Explanation: The @Value annotation injects the property value from application.properties into myProperty.

### 7. Which annotation is used to define a bean’s scope as prototype in Spring?

A) @Component (“prototype”)
B) @Scope(“prototype”)
C) @Prototype
D) @PrototypeScope
**Answer: B)** @Scope(“prototype”)
Explanation: The @Scope(“prototype”) annotation is used to define a bean’s scope as prototype, meaning a new instance will be created each time the bean is requested.

### 8. What is Dependency Injection in Spring?

A) The process of manually creating objects
B) A technique to achieve loose coupling in applications
C) A mechanism to handle transactions
D) A method for data access operations
Answer: B) A technique to achieve loose coupling in applications

Explanation: Dependency Injection is a design pattern that helps achieve loose coupling by injecting dependencies from outside the class, rather than creating them within the class.

### 9. What will happen if you run the following Spring configuration?

```
<bean id="myService" class="com.example.MyService" init-method="init" destroy-meth
<property name="dependency" ref="myDependency"/>
</bean>


<bean id="myDependency" class="com.example.MyDependency"/>
```

Assuming MyService class is defined as:

```
public class MyService {
   private MyDependency dependency;


   public void setDependency(MyDependency dependency) {
      this.dependency = dependency;
   }


   public void init() {
      System.out.println("MyService initialized");
   }


   public void cleanup() {
      System.out.println("MyService cleaned up");
   }
}
```

A) MyService initialized
B) MyService cleaned up
C) Both A and B
D) None of the above

**Answer: C)** Both A and B
Explanation: The init-method and destroy-method attributes specify methods to be called after bean initialization and before destruction.

### 10. What does the @Qualifier annotation do in Spring?

A) It sets the default scope of a bean
B) It helps in resolving ambiguity when multiple beans of the same type exist
C) It initializes the bean after properties are set
D) It defines a bean

**Answer: B)** It helps in resolving ambiguity when multiple beans of the same type exist
Explanation: @Qualifier annotation is used along with @Autowired to specify which bean should be injected when there are multiple beans of the same type.

### 11. What is the difference between a bean definition and a bean instance in Spring?

A) They are the same thing.
B) A bean definition is the blueprint, and a bean instance is the actual object created from it.
C) Bean definitions are for singletons, and bean instances are for prototypes.
D) Spring only uses bean definitions internally.

**Answer: B)** A bean definition is the blueprint, and a bean instance is the actual object created from it.
Explanation: : A bean definition specifies how to create a bean (class, properties, dependencies), while a bean instance is the actual object created based on that definition.

### 12. Which statement is true about Spring profiles?

A) We can define custom profiles for different environments.
B) We can define custom profiles for Development, Testing, and Production only.
C) There is only one default Spring profile.
D) Spring profiles are related to bean scopes.

**Answer: A)** We can define custom profiles for different environments.
Explanation: Spring profiles allow us to activate specific bean configurations for the different environments (e.g., dev, test, prod) using the @Profile annotation. For complete details with examples, kindly visit a separate article on Spring Profile.

### 13. Which Spring annotation is used to mark a class that provides business logic functionalities?

A) @Repository
B) @Component
C) @Service
D) @Controller

**Answer: C)** @Service
Explanation: The @Service annotation identifies a class that implements business logic functionalities within the application.

### 14. What are the advantages and disadvantages of using XML configuration for Spring beans compared to annotation-based configuration?

A) XML offers better maintainability, while annotations provide more concise code.
B) Annotations are easier to understand, while XML is more flexible for complex configurations.
C) Spring only supports XML configuration for defining beans.
D) There's no significant difference in functionality; it's a matter of developer preference.

**Answer: A)** XML offers better maintainability, while annotations provide more concise code.
Explanation: Annotations offer concise code but can be less readable for complex configurations. XML provides more flexibility but can become bulky for larger projects.

### 15. What are the components involved in Spring AOP?

A) Beans, properties, and profiles.
B) Aspects, advices, joinpoints, and target objects.
C) Controllers, services, and repositories.
D) Scopes, annotations, and configuration files.

**Answer: B)** Aspects, advices, joinpoints, and target objects.
Explanation: Spring AOP uses aspects (containing advices) that apply to specific joinpoints (method calls, field access) within target objects.

### 16. What is the purpose of the @Transactional annotation in Spring?

A) To define a custom transaction scope for a bean.
B) To define transactional behavior for database operations.
C) To manage caching of data.
D) To implement security checks on method calls.

**Answer: B)** To define transactional behavior for database operations.
Explanation: The @Transactional annotation marks methods that participate in a database transaction, ensuring data consistency and rollback capabilities.

### 17. How can you configure a custom exception handler in Spring MVC?

A) By creating a custom bean that handles exceptions.
B) Spring doesn't support custom exception handling.

C) By defining error pages in the web application.
D) By using the @ControllerAdvice annotation on a class with exception handler methods.

**Answer: D)** By using the @ControllerAdvice annotation on a class with exception handler methods.
Explanation: The @ControllerAdvice annotation marks a class that contains methods for handling specific exceptions thrown by controllers or other beans within the application.

### 18. What is the purpose of the @ControllerAdvice annotation in Spring?

A) To define custom controllers for handling web requests.
B) To define global exception handling for controllers.
C) To manage user authentication and authorization.
D) To implement security checks on method calls.

**Answer: B)** To define global exception handling for controllers.
Explanation: The @ControllerAdvice annotation marks a class that provides centralized exception handling for controllers throughout the application.

### 19. What is the primary purpose of Spring MVC?

A) To manage database connections and persistence.
B) To provide a foundation for building enterprise Java applications.
C) To handle web requests and responses for building web applications.
D) To customize MVC framework in Spring.

**Answer: C)** To handle web requests and responses for building web applications.
Explanation: Spring MVC offers a Model-View-Controller (MVC) framework for building web applications. It handles web requests, responses, and managing data flow between the model, view, and controller layers.

### 20. How can we map an incoming request URL to a specific controller method in Spring MVC?

A) By manually parsing the request URL in the controller.
B) Spring automatically maps URLs to controllers based on naming conventions.
C) We need to configure URL mappings in an XML configuration file.
D) By using the @RequestMapping annotation on the controller method.

**Answer: D)** By using the @RequestMapping annotation on the controller method.
Explanation: The @RequestMapping annotation on a controller method specifies the URL path that maps to that method for handling requests.

### 21. What is the role of the Model object in Spring MVC?

A) To handle web request and response data.
B) To hold data that needs to be passed from the controller to the view layer.
C) To manage the lifecycle of Spring MVC beans.
D) To define validation rules for user input.

**Answer: B)** To hold data that needs to be passed from the controller to the view layer.
Explanation: The Model object acts as a container for data that the controller wants to expose to the view layer for rendering.

**22. Which statement is correct about @Controller and @RestController annotations?**

A) They serve the same purpose, there is no difference.
B) @Controller handles both request and response data, while @RestController assumes JSON or serialized data for responses.
C) @RestController is for handling web forms, and @Controller is for REST APIs.
D) Spring MVC only uses @Controller.

**Answer: B)** @Controller handles both request and response data, while @RestController assumes JSON or serialized data for responses.
Explanation: Both annotations mark classes as Spring MVC controllers. However, @RestController assumes a response body with serialized data (e.g., JSON) and removes the need for explicit @ResponseBody annotations on methods.

**23. How can Spring Security be integrated with Spring MVC applications?**

A) Requires manual configuration for user authentication and role checks.
B) Spring Security automatically handles security for all Spring MVC applications.
C) Spring Security is a separate framework and cannot be used with Spring MVC.
D) Spring provides annotations like @PreAuthorize to integrate with Spring Security for authorization.

**Answer: D)** Spring MVC provides annotations like @PreAuthorize to integrate with Spring Security for authorization.
Explanation: Using @PreAuthorize annotation, we can integrate with Spring Security for authorization.

**24. What is the purpose of the Spring IoC container?**

A) To perform Aspect-Oriented Programming
B) To provide JDBC operations
C) To manage the lifecycle of beans
D) To handle MVC operations

**Answer: C)** To manage the lifecycle of beans
Explanation: The Spring IoC container is responsible for instantiating, configuring, and managing the lifecycle of Spring beans.

**25. What is the purpose of the @Configuration annotation in Spring?**

A) To mark a class as a Spring configuration class
B) To specify a bean's scope
C) To declare an aspect
D) To enable component scanning

**Answer: A)** To mark a class as a Spring configuration class
Explanation: The @Configuration annotation indicates that a class can be used by the Spring IoC container as a Spring configuration class for bean definitions.

**26. What is the role of the @ComponentScan annotation in Spring?**

A) To define a Spring bean
B) To configure a Spring application context
C) To enable component scanning in specified packages
D) To specify a bean's scope
**Answer: C)** To enable component scanning in specified packages
Explanation: The @ComponentScan annotation is used to enable component scanning so that Spring can find and register beans within the specified packages.

**27. Which of the following is true about Spring Framework?**

A) Spring supports only XML-based configuration
B) Spring does not support annotation-based configuration
C) Spring supports both XML and annotation-based configuration
D) Spring supports only annotation-based configuration

**Answer: C)** Spring supports both XML and annotation-based configuration
Explanation: Spring Framework supports both XML-based and annotation-based configurations for defining beans and wiring dependencies.

**28. Which of the following can be used for dependency management in Spring Boot project?**

A) Maven
B) Gradle
C) Ant
D) Both A and B

**Answer: D)** Both A and B
Explanation: Maven and Gradle are the two primary tools used for dependency management in Spring Boot projects. Ant is an older build tool that does not natively support dependency management as efficiently as Maven and Gradle.

**29. Which of the following statements is true about dependency management in Spring Boot?**

A) Spring Boot uses a proprietary dependency management system.
B) Spring Boot relies on third-party tools like Maven and Gradle for dependency management.
C) Spring Boot does not support dependency management.
D) Spring Boot uses XML-based configuration for dependency management.

**Answer: B)** Spring Boot relies on third-party tools like Maven and Gradle for dependency management.
Explanation: Spring Boot relies on widely-used third-party tools like Maven and Gradle for

dependency management. These tools handle the dependencies and build configuration for Spring Boot applications.

### 30. Which annotation in Spring Boot can be used to automatically manage dependencies?

A) @Component
B) @SpringBootApplication
C) @EnableAutoConfiguration
D) @Autowired

**Answer: C)** @EnableAutoConfiguration
Explanation: The @EnableAutoConfiguration annotation is part of the @SpringBootApplication meta-annotation and is used to automatically configure Spring application context. Although it does not directly manage dependencies, it enables the Spring Boot application to include necessary dependencies based on the classpath contents, hence promoting dependency management.

### 31. Spring dependency injection is useful because it makes your code:

A) Faster
B) More testable
C) Less readable
D) Platform-dependent

**Answer: B)** More testable
Explanation: Dependency injection makes our code more testable by allowing dependencies to be easily swapped out with mock implementations during testing. This leads to more modular and

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/ccee2025notes

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project