

## Q1 of 15



Given the following bean definition, what is the default name with which EmployeeRepository bean gets registered in IoC container?

```
public class EmployeeRepository{  
    // rest of the code  
}  
  
//bean configuration  
@Configuration  
public class SpringConfig {  
    @Bean  
    public EmployeeRepository employeeRepo() {  
        EmployeeRepository employeeRepository = new EmployeeRepository();  
        return employeeRepository;  
    }  
}
```

employerepository

employeeRepository

EmployeeRepository

employeeRepo

```
}

@Configuration
public class SpringConfig {
    @Bean
    public CubicleRepositoryImpl getCubicleRepository() {
        return new CubicleRepositoryImpl();
    }
    @Bean
    public CubicleServiceImpl getCubicleService() {
        return new CubicleServiceImpl();
    }
}
```

Assuming that necessary imports are done, what is the output when following code is executed?

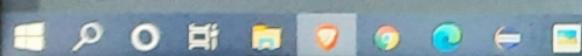
```
public class Tester {
    public static void main(String[] args) {
        ApplicationContext ctx = new AnnotationConfigApplicationContext(SpringConfig.class);
    }
}
```

CubicleRepositoryImpl constructor called CubicleServiceImpl constructor called

CubicleServiceImpl constructor called CubicleRepositoryImpl constructor called

The screenshot shows a learning management system interface. At the top right, it displays "Infosys Foundation P..." under "Learning Path | Intermediate" with a duration of "252h 44m". Below this is a search bar labeled "Queue" with icons for search, refresh, and info. A list of courses is shown, each with a checkmark indicating completion:

- java 11 Essentials (28h 35m)
- java 11 Beyond Basics (32h 53m)
- Service Layer using Spring (8h 59m)
  - Introduction to Service L... (34m 14s)
  - Spring IoC Containers (55m)
  - Dependency Injection using ... (55m)
  - Introduction to Spring Boot (1h 25m)
  - Logging in Spring Boot (50m)
  - AOP with Spring Boot (1h 10m)



```
@ComponentScan(basePackages="com.infy.service com.infy.repository com.infy.controller")
public class SpringConfig {
```

```
}
```

Assuming that necessary imports are done, what is the output when following code is executed?

```
package com.infy;

public class UserInterface{
    public static void main(String[] args){
        ApplicationContext applicationContext = new AnnotationConfigApplicationContext(SpringConfig.class);
        CustomerService service = (CustomerService) applicationContext.getBean("customerService");
    }
}
```

- CustomerService constructor called CustomerController constructor called CustomerRepository constructor called Class
- CustomerService constructor called CustomerRepository constructor called CustomerController constructor called
- CustomerController constructor called CustomerRepository constructor called CustomerService constructor called
- Nothing will be displayed in console



Infosys Foundation P...
Learning Path | Intermediate  
252h 44m

Queue



▼	Java 11 Essentials	28h 35m	✓
▼	Java 11 Beyond Basics	32h 53m	✓
▲	Service Layer using Spring	8h 59m	✓
▼	Introduction to Service L... 34m 14s	34m 14s	✓
▼	Spring IoC Containers	55m	✓
▼	Dependency Injection using ... 55m	55m	✓
▼	Introduction to Spring Boot 1h 25m	1h 25m	✓
▼	Logging in Spring Boot	50m	✓
▼	AOP with Spring Boot	1h 10m	✓

Q4 of 15

Which of the following statement(s) are true about Spring Boot? [Choose any two options]

- Spring Boot by default does not configures logging.
- Spring Boot by default configures logging via Log4j.
- Spring Boot by default configures logging via Apache Commons Logging with Logback implementations
- @SpringBootApplication annotation is combination of @EnableAutoConfiguration, @ComponentScan and @Configuration annotation.

Q5 of 15

Consider the following bean and configuration class:

```
package com.infy.service;  
  
@Component(value = "mobileService")  
public class MobileServiceImpl {
```



Infosys Foundation P...  
Learning Path |  
Intermediate  
252h 44m

Queue

▼	Java 11 Essentials	28h 35m	✓
▼	Java 11 Beyond Basics	32h 53m	✓
▲	Service Layer using Spring	8h 59m	✓
▼	Introduction to Service L...	34m 14s	✓
▼	Spring IoC Containers	55m	✓
▼	Dependency Injection using ...	55m	✓
▼	Introduction to Spring Boot	1h 25m	✓
▼	Logging in Spring Boot	50m	✓
▼	AOP with Spring Boot	1h 10m	✓

}

```
package com.infy.configuration;  
@Configuration  
@ComponentScan(basePackages = "com.infy.service")  
public class SpringConfig {  
}
```

Which of the following is/are correct way(s) of getting object of MobileServiceImpl? [Choose 2 options]

- ApplicationContext context = new AnnotationConfigApplicationContext(SpringConfig.class);  
MobileServiceImpl ms = context.getBean(MobileServiceImpl.class);
- ApplicationContext context = new AnnotationConfigApplicationContext(SpringConfig.class);  
MobileServiceImpl ms = context.getBean("mobileService");
- ApplicationContext context = new AnnotationConfigApplicationContext(SpringConfig.class);  
MobileServiceImpl ms = (MobileServiceImpl) context.getBean("mobileService");
- ApplicationContext context = new AnnotationConfigApplicationContext(SpringConfig.class);  
MobileServiceImpl ms = (MobileServiceImpl) context.getBean("MobileService");



Infosys Foundation P...  
Learning Path |  
Intermediate  
252h 44m

Queue



- ▼  Java 11 Essentials 28h 35m ✓
- ▼  Java 11 Beyond Basics 32h 53m ✓
- ▲  Service Layer using Spring 8h 59m ✓
  - ▼  Introduction to Service L... 34m 14s ✓
  - ▼  Spring IoC Containers 55m ✓
  - ▼  Dependency Injection using ... 55m ✓
  - ▼  Introduction to Spring Boot 1h 25m ✓
  - ▼  Logging in Spring Boot 50m ✓
- ▼  AOP with Spring Boot 1h 10m ✓
  - ^  ENG 04:54

QUESTION 1

```
}
```

```
@Configuration
```

```
public class SpringConfig{
```

```
    @Bean
```

```
    public StudentRepositoryImpl studentRepository() {
```

```
        return new StudentRepositoryImpl();
```

```
    }
```

```
}
```

Assuming that necessary imports are done, what is the output when following code is executed?

```
public class UserInterface {
```

```
    public static void main(String[] args) {
```

```
        ApplicationContext ctx = new AnnotationConfigApplicationContext(SpringConfig.class);
```

```
        StudentRepositoryImpl w1 = ctx.getBean(StudentRepositoryImpl.class);
```

```
        StudentRepositoryImpl w2 = ctx.getBean(StudentRepositoryImpl.class);
```

```
        System.out.print(w1==w2);
```

```
        System.out.println(" " + w1.equals(w2));
```

```
    }
```

```
}
```

true true

true false



```
@Configuration  
public class SpringConfig {  
    @Bean  
    public BankRepositoryImpl bankRepository() {  
        return new BankRepositoryImpl();  
    }  
}
```

Assuming that necessary imports are done, what is the output when the following code is executed?

```
public class UserInterface {  
    public static void main(String[] args){  
        ApplicationContext context = new AnnotationConfigApplicationContext(SpringConfig.class);  
        BankServiceImpl bankService = context.getBean(BankServiceImpl.class);  
    }  
}
```

- NoSuchElementException exception will be thrown.
- BankServiceImpl bean will be created with a null BankRepositoryImpl reference.
- BankServiceImpl bean will be created having a reference of BankRepositoryImpl.
- None of the above.

None of the above.

Q8 of 15



Choose the correct option(s) for which the following pointcut expression is applicable:

`execution(Integer com.ischool.Student.add*(...))`



Only addProduct method of com.ischool.Student class which has no arguments

All methods of com.ischool.Student class whose name is add\*

All the methods of com.ischool.Student class whose name starts with "add", takes any number of arguments and returns Integer

Only addProduct method of com.ischool.Student class which takes atleast one argument



Q9 of 15



Which of the following option(s) are used to declare advice in Spring AOP? [Choose 3 options]

Only addProduct method of com.lschool.Student class which takes atleast one argument

Q9 of 15

Which of the following option(s) are used to declare advice in Spring AOP? [Choose 3 options]

@Before

@Pointcut

@AfterReturning

@After

Q10 of 15

Given the following unit test, what is/are missing from the class definition that prevents the test from being executed correctly?

```
public class CustomerServiceTest {
```

```
    CustomerService customerService = new CustomerService();
```

```
public class CustomerServiceTest {  
  
    CustomerService customerService = new CustomerService();  
  
    CustomerRepository customerRepository;  
  
    @Test  
    public void findById() {  
        Mockito.when(customerRepository.findById(1L)).thenReturn(new Customer());  
        Customer customer = customerService.findById(1L);  
        Assertions.assertNotNull(customer);  
    }  
}
```

- The class should be annotated with `@SpringBootTest` annotation
- Nothing, the test will be executed correctly and it will pass
- The `customerRepository` field is missing `@Mock` annotation
- The `customerService` field is missing `@InjectMocks` annotation

## Q11 of 15

What is the name of following TraineeServiceImpl Spring bean?

```
@Component(value="traineeBean")
public class TraineeServiceImpl{
}
```

traineeBean

traineebean

TraineeServiceImpl

traineeServiceImpl

## Q12 of 15

Consider the following Spring bean and configuration class:

```
package com.infy.service;
public class RegistrationServiceImpl {
    public RegistrationServiceImpl() {

```



Infosys Foundation P...  
Learning Path |  
Intermediate  
252h 44m

Queue



- Java 11 Essentials 28h 35m ✓
- Java 11 Beyond Basics 32h 53m ✓
- Service Layer using Spring 8h 59m ✓
  - Introduction to Service L... 34m 14s ✓
  - Spring IoC Containers 55m ✓
  - Dependency Injection using ... 55m ✓
  - Introduction to Spring Boot 1h 25m ✓
  - Logging in Spring Boot 50m ✓
  - AOP with Spring Boot 1h 10m ✓

}

Assuming that necessary imports are done, what is the output when the following code is executed?

```
package com.infy;
public class Tester {
    public static void main(String[] args) {
        ApplicationContext ctx = new AnnotationConfigApplicationContext(SpringConfig.class);
        RegistrationServiceImpl regService = (RegistrationServiceImpl) ctx.getBean("myregistration");
        regService.showStatus();
    }
}
```



- Registration in progress... Registered
- Registered
- Registered Registration in progress...
- The code will throw exception at runtime.

Registered

Registered Registration in progress...

The code will throw exception at runtime.

**Q13 of 15**

Fill in the blanks.

The `@Controller` annotation is a stereotype for the annotated class, indicating it as a web component and is a specialization of `@Component` annotation.

**Q14 of 15**

Consider the following properties files present in classpath of the Spring Boot application:

`application.properties`

`msg = Hello Spring`

`messages.properties`

`welcome = Welcome to Spring`



Infosys Foundation P...  
Learning Path | Intermediate  
252h 44m

Queue



- Java 11 Essentials 28h 35m ✓
- Java 11 Beyond Basics 32h 53m ✓
- Service Layer using Spring 8h 59m ✓
- Introduction to Service L... 34m 14s ✓
- Spring IoC Containers 55m ✓
- Dependency Injection using ... 55m ✓
- Introduction to Spring Boot 1h 25m ✓
- Logging in Spring Boot 50m ✓
- AOP with Spring Boot 1h 10m ✓

## messages.properties

welcome = Welcome to Spring

Assuming that necessary imports are done, what is the output when following code is executed?

```
@SpringBootApplication  
{@PropertySource("classpath:messages.properties")  
public class DemoApplication implements CommandLineRunner {  
    @Autowired  
    Environment environment;  
  
    public static void main(String[] args) {  
        SpringApplication.run(DemoApplication.class, args);  
    }  
  
    public void run(String... args) throws Exception {  
        System.out.println(environment.getProperty("welcome")+" ");  
        System.out.println(environment.getProperty("msg")+" ");  
    }  
}
```



Welcome to Spring Hello Spring

Hello Spring Welcome to Spring

null Welcome to Spring

null Hello Spring

Q15 of 15



Which of the following AOP advice can access and modify arguments of a joinpoint?

1. @AfterReturning

2. @Around



Only 1

Only 2

Both 1 and 2

None of the given advices.

# ResultPASS

12

Correct

3

Incorrect

0

Unanswered

80

Percentage