

Q1. Which of the following are valid injection methods in spring?

- ☒ a. Setter Injection
- ☒ b. Constructor Injection
- c. Property Injection
- d. Class Injection

Q2. What is an Advice in AOP?

- a. This is the way to instruct an object to behave in a certain manner.
- b. This is used to inject values in objects.
- ☒ c. This is an actual action to be taken either before or after the method execution.
- d. This is not invoked during program execution by spring AOP framework.

Q3. The HTTP 500X status code indicates\_\_\_\_\_?

- a. Request Processed Successfully
- ☒ b. Server side error
- c. Additional action is required by client like redirect
- d. Client side error

Q4. To initialize the Session Factory in java Hibernate, choose the appropriate function?

Configuration.configuration = new Configuration()

- ☒ a. Build Session Factory
- b. Session Factory
- c. New Session Factory
- d. Create Session Factory

Q5. Which of the following View Resolver maps each view name to an application's directory?

- a. Internal View Resolver
- b. Internal Resource View Resolver
- c. Resource Bundle View Resolver
- d. Internal Resolver

Q6. Which annotation specifies that the class is an entity?

- a. @ID
- b. @Column
- c. @Table
- ☒ d. @Entity

Q7. What is default value of autowire attribute?

- a. By Type
- b. Constructor
- c. By name
- ☒ d. No

Q8. What are the types of autowiring in spring?

- a. ByName, byType, destructor and autodetect
- b. ByName, byMethod, constructor and autodetect
- c. ByName, byType, constructor and autocorect
- d. ByName, byType and constructor.

Q9. Which of these can be annotated in a class to indicate that the class can be used by the spring IoC container as a source of bean definitions?

- a. @Configuration
- ☒ b. @Bean
- c. @Bean source
- d. @Bean configuration

Q10. Which of these is the one described below?

Some of the values returned by this are a String, a View, or a ModelAndView.

- ☒ a. ViewResolvers
- b. HandlerMapping
- c. HandleMethods
- d. Controller

Q11. What is a DispatcherServlet?

- a. DispatcherServlet is used for transaction management
- b. DispatcherServlet is used to achive AOP.
- ☒ c. DispatcherServlet handles all the HTTP requests and response
- d. DispatcherServlet is used to achive Dependency Injection.

Q12. Which of the following modules does not provide?

- a. Web-Servlet module
- b. Spring ORM
- ☒ c. Spring Boot
- d. AOP

Q13. Which of the following best describes Spring Tool Suite?

- a. Not compatible with java runtime
- b. Manual application validation is required
- ☒ c. Eclipse-based development environment.
- d. Only available for windows 32-bit.

Q14. What are the available implementation classes or sub interfaces of Bean Factory?

- ☒ a. ApplicationContext
- ☒ b. XmlBeanFactory
- c. FileSystemBeanFactory
- d. AdvancedBeanFactory

Q15. Which annotation added as an input parameter to the handler method?

- a. @Path
- b. @PathLocale
- ☒ c. @PathVariable
- d. @PathParam

Q16. BeanFactory is an implementation of the \_\_\_\_\_?

- a. Observer Design Pattern
- b. Service Locator Pattern
- ☒ c. Factory Design Pattern
- d. Proxy Pattern

Q17. Which of the following is the default scope in spring?

- a. Prototype
- ☒ b. Singleton
- c. Request
- d. Session

Q18. Spring is a lightweight framework for\_\_?

- a. Database ready application
- b. Enterprise-ready application
- c. View ready applications
- ☒ d. Web application development.

Q19. In AOP \_\_\_\_\_ is the process of linking aspects with other application types or objects to create an advised object?

- a. Binding
- b. Wiring
- ☒ c. Weaving
- d. Data binding

Q20. Define a prototype scope?

- ☒ a. This scopes a single bean definition to have any number of object instances.
- b. This scopes the bean definition to a single instance per HTTP Request.
- c. This scopes the bean definition to a single instance per HTTP Session.
- d. This scopes the bean definition to a single instance per HTTP Application/Global session

Q21. Which of the following components of HTTP request is used to identify the resource on server?

- a. VERB
- b. Request Header
- c. HTTP Version
- ☒ d. URI

Q22. What does Spring Boot Provide?

- ☒ a. Support for Jetty and Undertow as embedded servers
- b. Java code generation
- ☒ c. Auto-configuration of the Spring Framework and third libraries
- d. Support for Websphere and Weblogic as embedded servers
- ☒ e. Support both Java-based and YAML for Spring application context configuration

Q23. How can we hook initialization of clean up code in spring?

- ☒ a. Using custom init() and destroy () methods
- ☒ b. Using interfaces Initializing Bean and DisposableBean
- c. Using <bean> tag
- d. It is not possible

Q24. Which of the following is not an inheritance mapping Strategies?

- a. Table per hierarchy
- b. Table per concrete class
- c. Table per subclass
- ☒ d. Table per class

Q25. What does AOP stand for?

- a. Asset Oriented Programming
- b. Any Object Programming
- c. Aspect Oriented Programming
- d. Aspect Oriented Protocol

Q26. To work with simple JDBC in spring, we use\_\_\_\_\_?

- a. JPATransactionManager
- ☒ b. DataSource TransactionManager
- c. JDOTransactionManager
- d. JTATransactionManager

Q27. Select the hibernate properties that need to be configured in the hibernate file to allow second-level caching?

- ☒ a. hibernate.cache.use\_second\_level\_cache
- ☒ b. hibernate.cache.region.factory\_class
- c. hibernate.enable\_cache
- d. hibernate.cache.cached classes

Q28. Which of the following options are true regarding HQL?

- a. Database independent language.
- b. We cannot sort the result set
- c. Cannot add conditions while fetching results.
- d. Can fetch selected columns from the table.

a. only a

☒ b. a and d

c. c and d

d. b and c

Q29. Which of the following properties should be enabled to all actuator endpoints for monitoring?

- ☒ a. management.endpoints.web.exposure.include=\*

- b. `management.endpoints.web.exposure.all=*`
- c. `management.endpoints.web.exposure.enable=*`
- d. `management.endpoints enable.exposure=*`.

Q30. Which of the following HTTP method is idempotent in nature?

- a. Put & post both
- b. Post
- ☒ c. Put
- d. Patch

Q31. In spring bean configuration \_\_\_\_\_ element helps in writing, which is injecting a list of values, allowing duplicates?

- ☒ a. `<list>`
- b. `<set>`
- c. `<map>`
- d. `<props>`

Q32. Which one of the tags is used to call the setter method in spring bean?

- ☒ a. `<property>`
- b. `<constructor-arg>`
- c. `<Setter>`
- d. `<bean>`

Q33. Which are the IoC containers in spring?

- a. BeanFactory, ApplicationContext, locContextFactory
- ☒ b. BeanFactory, ApplicationContext
- c. BeanFactory, BeanContext, locContextFactory
- d. BeanFactory, ApplicationContext, BeanContext

Q34. Given the below code. (Assume necessary imports are included).

```
@RestController
```

```
@RequestMapping("/friendsfamily")
```

```
public class FriendsFamilyController
```

```
{
```

```
@PostMapping (produces="application/json", consumes="application/json")
```

```
public FriendsFamilyDTO addFriendsFamily (@RequestBody FriendsFamilyDTO
friendsFamilyDTO
```

```
{

//code to add friends family and return the same back

}
```

Identify the things that are optional from the following.

- a. @RequestBody
- b. produces="application/json"
- c. consumes="application/json"
- a. only a
- ☒ b. only b and c
- c. only a and c
- d. a,b,c

Q35. Consider MobileService class with two properties mobileNumber and model as shown below.

```
package com.test;

public class MobileService {
    private long mobileNumber;
    private String model;
    // parameterized constructor
    public MobileService (long mobileNumber, String model) {
        this.mobileNumber = mobileNumber;
        this.model = model;
    }
}
```

config.xml -Configuration file as below:



```
<bean id="mobileService" class="com.test. MobileService"><constructor-arg
value="Nokia"/>
<constructor-arg value="1234567892"/>
</bean>
```

The main method of the application is shown below:

```
ApplicationContext ac=new ClassPathXmlApplicationContext("config.xml"); MobileService
m = (MobileService)ac.getBean ("mobileService");
```

What is the output of the above code after execution?

- ☒ a. Throws Exception as the constructor arguments are not passed in appropriate order in the bean definition
- b. mobileService bean of MobileService class type is created with values as "1234567892" for mobile Number and "Nokia" for model
- c. mobileService bean of MobileService class type is created with values as "0" for mobile Number and "1234567892" for model
- d. mobileService bean of MobileService class type is created with values as "0" for mobileNumber and "null" for model.

Q36. A Spring application includes the following class definitions:

```
package com.test.service;
```

```
public class EmployeeService {
    private int empno;
    private String ename;
    @Autowired
    private AddressService office;
    //getter and setter methods for all properties
}
```

```
package com.test.service;
```

```
public class AddressService {
    String city;
    String street;
    //getter and setter methods for all properties }
```

config.xml:



```
<context:annotation-config />
```

```
<bean id="emp" class="com.test.service.EmployeeService" >
```

```
<property name="empno" value="1001"/>
```

```
<property name="ename" value="John"/>
```

```
</bean>
```

Main method:

```
ApplicationContext context = new Class PathXml ApplicationContext("config.xml");
```

```
Employee Service emp = (Employee Service) context.getBean ("emp");
```

What will happen if the main method of this application is executed?

- ☒ a. Employee Service bean will be created with a null Address Service reference.
- b. Employee Service bean will be created which references an Address Service object with the value of city and street as null.
- c. A Runtime Exception will be thrown.
- d. Compile Time Error as @Autowired cannot be applied on Address Service attribute.

Q37. Given the below code:

```
@RestController
```

```
@RequestMapping("/friendsfamily")
```

```
public class FriendsFamilyController
```

```
{
```

```
    @GetMapping("/{phoneNumber}")
```

```
    public FriendsFamilyDTOgetFriendsFamily(){
```

```
{
```

```
//code to search for friends family based on the phone number and returns the same
```

```
}
```

}

Choose the correct line of code that occupy the parameter section of the request method?

- a. `@PathParam("phoneNumber") long phoneNumber`
- b. `@RequestParam("phoneNumber") long phone Number`
- ☒ c. `@PathVariable("phoneNumber") long phoneNumber`
- d. `@RequestBody("phoneNumber") long phoneNumber`

Q38. Which of the following can be used to specify the scope of a bean?

1. In a XML based configuration file:

```
<bean id="a1" class="Constructor Injection. TaxiService" scope="prototype"/>
```

2. In a Java based configuration file:

```
@Bean (name="a3")  
@Scope("prototype")  
public TaxiService getTaxi() { }
```

3. In the POJO class:

```
@Component("a2")  
@Scope ("prototype")  
public class TaxiService {}
```

Options:

- a. Only 1
- b. 1 & 2
- ☒ c. 1,2,3
- d. 2,3

Q39. Consider the following classes defined in a Spring application:

```
package com.demo.company;
```

```
public class Address {
```

```
private String city;
private String pincode;
public Address () { }
//getter and setter methods)

package com.demo.company;

public class Employee {

private Address address;

//getter and setter methods }
```

Config.xml:

```
<bean id="address" class="com.demo.company.Address">

<property name="city" value="Los Angeles" />

<property name="pincode" value="91167 " />

</bean>

<bean id="employee" class="com.demo.company.Employee" autowire="byName"/>
```

Main class:

```
ApplicationContext ac= new ClassPathXmlApplicationContext("config.xml");

Employee el = (Employee) ac.getBean ("employee");

System.out.println("City: "+el.getAddress().getCity () +" Street:
"tel.getAddress().getPincode());
```

What is the output of the above code execution?

Options:

- a. Throws NullPointerException
- b. Display city as "Los Angeles" and pincode as null
- c. Display city and pincode as null
- d. No bean named 'address' available
- ☒ e. Display city as "Los Angeles" and pincode as "91167".

