**1) What is Spring?**

It is a lightweight, loosely coupled and integrated framework for developing enterprise applications in java.

**2) What are the advantages of spring framework?**

1. Predefined Templates
2. Loose Coupling
3. Easy to test
4. Lightweight
5. Fast Development
6. Powerful Abstraction
7. Declarative support

**3) What are the modules of spring framework?**

1. Test
2. Spring Core Container
3. AOP, Aspects and Instrumentation
4. Data Access/Integration
5. Web

**4) What is IOC and DI?**

IOC (Inversion of Control) and DI (Dependency Injection) is a design pattern to provide loose coupling. It removes the dependency from the program.

Let's write a code without following IOC and DI.

public class Employee{

Address address;

Employee(){

address=new Address();//creating instance

}

}

Now, there is dependency between Employee and Address because Employee is forced to use the same address instance.

Let's write the IOC or DI code.

public class Employee{

Address address;

Employee(Address address){

this.address=address;//not creating instance

}

}

Now, there is no dependency between Employee and Address because Employee is not forced to use the same address instance. It can use any address instance.

**5) What is the role of IOC container in spring?**

IOC container is responsible to:

* create the instance
* configure the instance, and
* assemble the dependencies

**6) What are the types of IOC container in spring?**

There are two types of IOC containers in spring framework.

1. BeanFactory
2. ApplicationContext

**7) What is the difference between BeanFactory and ApplicationContext?**

BeanFactory is the **basic container** whereas ApplicationContext is the **advanced container**. ApplicationContext extends the BeanFactory interface. ApplicationContext provides more facilities than BeanFactory such as integration with spring AOP, message resource handling for i18n etc.

**8) What is the difference between constructor injection and setter injection?**

|  |  |  |
| --- | --- | --- |
| **No.** | **Constructor Injection** | **Setter Injection** |
| 1) | No Partial Injection | Partial Injection |
| 2) | Desn't override the setter property | Overrides the constructor property if both are defined. |
| 3) | Creates new instance if any modification occurs | Doesn't create new instance if you change the property value |
| 4) | Better for too many properties | Better for few properties. |

**9) What is autowiring in spring? What are the autowiring modes?**

Autowiring enables the programmer to inject the bean automatically. We don't need to write explicit injection logic. Let's see the code to inject bean using dependency injection.

<bean id="emp" class="com.javatpoint.Employee" autowire="byName" />

The autowiring modes are given below:

|  |  |  |
| --- | --- | --- |
| **No.** | **Mode** | **Description** |
| 1) | no | this is the default mode, it means autowiring is not enabled. |
| 2) | byName | injects the bean based on the property name. It uses setter method. |
| 3) | byType | injects the bean based on the property type. It uses setter method. |
| 4) | constructor | It injects the bean using constructor |

**10) What are the different bean scopes in spring?**

There are 5 bean scopes in spring framework.

|  |  |  |
| --- | --- | --- |
| **No.** | **Scope** | **Description** |
| 1) | singleton | The bean instance will be only once and same instance will be returned by the IOC container. It is the default scope. |
| 2) | prototype | The bean instance will be created each time when requested. |
| 3) | request | The bean instance will be created per HTTP request. |
| 4) | session | The bean instance will be created per HTTP session. |
| 5) | globalsession | The bean instance will be created per HTTP global session. It can be used in portlet context only. |

**11) In which scenario, you will use singleton and prototype scope?**

Singleton scope should be used with EJB **stateless session bean** and prototype scope with EJB **stateful session bean**.

**12) What are the transaction management supports provided by spring?**

Spring framework provides two type of transaction management supports:

1. **Programmatic Transaction Management**: should be used for few transaction operations.
2. **Declarative Transaction Management**: should be used for many transaction operations.

**13) What are the advantages of JdbcTemplate in spring?**

**Less code**: By using the JdbcTemplate class, you don't need to create connection,statement,start transaction,commit transaction and close connection to execute different queries. You can execute the query directly

**14) What are classes for spring JDBC API?**

1. JdbcTemplate
2. SimpleJdbcTemplate
3. NamedParameterJdbcTemplate
4. SimpleJdbcInsert
5. SimpleJdbcCall

**15) How can you fetch records by spring JdbcTemplate?**

You can fetch records from the database by the **query method of JdbcTemplate**. There are two interfaces to do this:

1. [ResultSetExtractor](https://www.javatpoint.com/ResultSetExtractor-example)
2. [RowMapper](https://www.javatpoint.com/RowMapper-example)

**16) What is the advantage of NamedParameterJdbcTemplate?**

NamedParameterJdbcTemplate class is used to pass value to the named parameter. A named parameter is better than ? (question mark of PreparedStatement).

It is **better to remember**.

1**7) What is the advantage of SimpleJdbcTemplate?**

The **SimpleJdbcTemplate** supports the feature of var-args and autoboxing.

**18) What is AOP?**

AOP is an acronym for Aspect Oriented Programming. It is a methodology that divides the program logic into pieces or parts or concerns.

It increases the modularity and the key unit is Aspect.

**19) What are the advantages of spring AOP?**

AOP enables you to dynamically add or remove concern before or after the business logic. It is **pluggable** and **easy to maintain**.

**20) What are the AOP terminology?**

AOP terminologies or concepts are as follows:

* JoinPoint
* Advice
* Pointcut
* Aspect
* Introduction
* Target Object
* Interceptor
* AOP Proxy
* Weaving

**21) What is JoinPoint?**

JoinPoint is any point in your program such as field access, method execution, exception handling etc.

**22) Does spring framework support all JoinPoints?**

No, spring framework supports method execution joinpoint only.

**23) What is Advice?**

Advice represents action taken by aspect.

**24) What are the types of advice in AOP?**

There are 5 types of advices in spring AOP.

1. Before Advice
2. After Advice
3. After Returning Advice
4. Throws Advice
5. Around Advice

**25) What is Pointcut?**

Pointcut is expression language of Spring AOP.

**26) What is Aspect?**

Aspect is a class in spring AOP that contains advices and joinpoints.

**27) What is Introduction?**

Introduction represents introduction of new fields and methods for a type.

**28) What is target object?**

Target Object is a proxy object that is advised by one or more aspects.

**29) What is interceptor?**

Interceptor is a class like aspect that contains one advice only.

**30) What is weaving?**

Weaving is a process of linking aspect with other application.

**31) Does spring perform weaving at compile time?**

No, spring framework performs weaving at runtime.

**32) What are the AOP implementation?**

There are 3 AOP implementation.

1. Spring AOP
2. Apache AspectJ
3. JBoss AOP

**33) What is the front controller class of Spring MVC?**

The **DispatcherServlet** class works as the front controller in Spring MVC.

**34) What does @Controller annotation?**

The **@Controller** annotation marks the class as controller class. It is applied on the class.

**35) What does @RequestMapping annotation?**

The **@RequestMapping** annotation maps the request with the method. It is applied on the method.

**36) What does the ViewResolver class?**

The **View Resolver** class resolves the view component to be invoked for the request. It defines prefix and suffix properties to resolve the view component.

**37) Which ViewResolver class is widely used?**

The **org.springframework.web.servlet.view.InternalResourceViewResolver** class is widely used.

**38) Does spring MVC provide validation support?**

Yes.