**CHAPTER 1**

1. **Spring is what kind of component? (page.1)**

a) Lightweight

b) Heavyweight

c) None of one

**Ans. A**

**IoC is a technique that externalizes the creation and management of (page.2)**

a) Context dependencies

b) Component dependencies

c) Action dependencies

d) None

**Ans. B**

1. **Benefits of DI(Dependency Injection) is (p.3-4)**

a) Reduced glue code

b) Simplified application configuration

c) Ability to manage common dependencies

d) Improved testability

e) Fostering good application design

**Ans. A, B, C, D, E**

1. **Spring Expression Language (SpEL) is a technology to allow an application (page.5)**

a) To manipulate Java objects at runtime

b) To evaluate expressions at runtime

c) To access Java objects and spring beans at runtime

d) Above all

**Ans. D**

1. **Spring’s data access module provides out-of-the-box support for (page.6)**

a) JDBC

b) JDO

c) Hibernate

d) JPA

e) Above all

**Ans. E**

1. **Transforming JavaBeans into XML(page.7)**

a) Marshaling

b) Unmarshaling

**Ans. A**

1. **Transforming XML into Java objects (page.7)**

a) Marshaling

b) Unmarshaling

**Ans. B**

1. **What is spring? (page.1)**
2. a database
3. a framework
4. a component
5. a java class

**Ans. B**

1. **The core of the Spring Framework is based on the principle of(page.2)**
2. DoC
3. JNDI
4. IoC
5. XML

**Ans. C**

1. **What is IoC? (page.2)**
2. A javaBean
3. Depend on DI
4. A framework
5. Externalize the management of component dependencies

**Ans. D**

1. **Spring’s DI implementation is based around two core java concepts: (page.2)**
2. javaBeans
3. interfaces
4. java object
5. method

**Ans. A, B**

1. **There are different ways to configure dependency. Which are? (page.2)**
2. Externally in Xml file
3. Spring java configuration classes
4. Faces-config
5. Java annotations

**Ans. A, B, D**

1. **AOP provides the ability to implement (page.4)**
2. Crosscutting logic
3. MVC logic
4. Constraint logic
5. Bean logic

**Ans. A**

1. **Spring introduce which expression Language? (page.5)**
2. Special Expression language
3. SQL Expression
4. SpEL
5. Java Language

**Ans. C**

1. **What does mean the @NotNull annotation to bean’s property ? (page.5)**
2. Attribute shouldn’t contain a null value
3. Attribute should contain a null value
4. Shouldn’t use for validation

**Ans. A**

1. **By default, Spring will first look for which validator? (page.5)**
2. Spring validator
3. Hibernate validator
4. Bean validator
5. Xml validator

**Ans. B**

1. **Which are true ? (page.7)**
2. Marshaling (transforming JavaBeans to XML)
3. Unmarshaling (transforming JavaBeans to XML)
4. Unmarshaling (transforming XML into Java objects)
5. Marshaling (transforming XML into Java objects)

**Ans. A, C**

1. **Which are the alternatives to Spring Framework? (page.12)**
2. JBoss Seam Framework
3. Google Guice
4. PicoContainer
5. JEE 6 Container
6. Above All

**Ans. E**

1. **AOP stands for(page.4)**
2. Access oriented programming
3. Aspect organized programming
4. Aspect-oriented programming
5. All of the above

**Ans: C**

1. **Which is not Spring own Module JAR File(page.15)**
2. Aop
3. Oxm
4. primeface
5. Asm

**Ans: c**

1. **Spring is described as a (page.1)**
2. Heavyweight framework
3. Loosely type framework
4. Lightweight framework
5. Stand alone framework

**Ans. C**

1. **Spring DI implementation is based around which two core java concepts. (page.2)**

a. JavaBeans.

b. jdbc

c. SpEL

d. Interfaces.

**Ans. A, D**

1. **When Dependency Injections are injected by Spring ? (page.2)**
2. Runtime
3. Coding time
4. Compile time

**Ans. A**

1. **JavaBeans also Known as ? (page.2)**
2. POJOs
3. XML
4. ANNOTATIONS

**Ans. a**

**Chapter -2**

1. **If we use Maven for Spring’s applications, where we can add dependencies? (page.18)**
2. pom.xml
3. Web.xml
4. Context.xml
5. Config.xml

**Ans. A**

1. **Which is a byte code manipulation framework ? (page.15)**
2. Aop
3. asm
4. Jdbc
5. Jsm

**Ans. B**

1. **Which module is needed for every Spring application? (page.16)**
2. Bean
3. Jdbc
4. core
5. Orm

**Ans. C**

1. **If you are using EJB 2.1 or prior versions then you must use –Style of IoC? (page.6)**
2. Lookup style
3. Injection Style
4. None

**Ans. A**

1. **Which one has zero Impact on your component code?**
2. Injection
3. Lookup
4. both

**Ans. A**

1. **Each Bean can be assigned either an -----**

a ) ID or a name or both

b) ID or property

c) ID or method

**Ans. A**

1. **Which Interface reads XML files ?**
2. XmlBeanDefinitionReader
3. Serializable
4. BeanDefinitionRegistry

**Ans. A**

1. **Which version of jdk and spring support java annotation ?**
2. Jdk 4 and spring 1.0
3. Jdk 5 and spring 2.5
4. Jdk 2 and spring 2.0

**Ans. B**

**Chapter 4**

1. **Bean Factory is**

a) An Interface

b) a Object

c) a Class

d) None

**ans. A**

1. **ApplicationContext is**

a) Interface

b) Object

c) Class

d) None

**ans. A**

1. **ApplicationContext is an extension of**

a) ActionServlet

b) Action

c) BeanFactory

d)None

**Ans. C**

1. **There are two ways we can configure the ApplicationContext in Spring. Which are**

a) XML based

b) Annotation based

c) jdbc based

d) SpEL Based

**Ans. A, B**

1. **Some service provided by BeanFactory are:**

a) Inheritance

b) life-cycle

c) autowiring

d) None

**ans. A, b, c**

1. **A component that requires certain dependencies is often referred to as the**

a) ActionServlet

b) Dependent object

c) Independent object

d) None

Ans. B

1. **IoC can be decomposed into two subtypes are**

a) Dependency Injection

b) Dependency Lookup

c) Bean factory

d) None

**ans. A, B**

1. **Dependency Lookup is**

a) Traditional approach

b) Newer approach

c) Both

**ans. A**

1. **Dependency Injection is**

a) Traditional approach

b) Newer approach

c) Both

**Ans. B**

1. **Dependencies are injected into the component by** the

a) Servlet container

b) java container

c) IoC container

**Ans. C**

1. **Dependency Lookup are two types**

a) Dependency Pull

b) Contextualized Dependency Lookup (CDL)

c) Bean factory

d) None

**ans. A, B**

1. **Dependency Injection are two types**

a) Constructor Dependency Injection

b) Setter Dependency Injection

c) Both twos

**Ans. C**

1. **Dependencies are pulled from a registry as required in(page.45)**

a) Dependency Pull

b) Setter Dependency Injection

c) Constructor Dependency Injection

**Ans. A**

1. **Lookup-based solutions are more complex than injection-based ones**

a) True

b) False

**Ans. A**

1. **Passive code is not much simpler to maintain than active code**

a) True

b) False

**Ans. B**

1. **Setter Injection is that it allows dependencies to be declared on an interface(page.59)**

a) True

b) False

**Ans. A**

1. **Configuration parameters are(page.60)**

a) passive

b) Information

c) Components

d) Simple values

**Ans. A, B, D**

1. **Transaction and AOP service, message source for internationalization (i18n), and application event handling are the services of(page.65)**

a) ApplicationContext

b) BeanFactory

c) FactoryBean

d) ActionServlet

**Ans. A**

1. **XML file configuration will override the annotation ones(page.65)**

a) True

b) False

**Ans. A**

1. **Which namespace provides support for configuring Spring’s ApplicationContext(page.66)**

a) context

b) p

c) c

d) ActionServlet

**Ans. A**

1. **Which namespace provides a simpler DI configuration for Setter Injection(page.66)**

a) e

b) p

c) c

Ans. B

1. **Which namespace provides a more simple DI configuration for Constructor Injection(page.66)**

a) context

b) p

c) c

**Ans. C**

1. **Which namespace provides some useful utilities for DI configuration(page.66)**

a) context

b) p

c) c

d) util

**Ans. D**

1. **Multiple base-package can be defined by using**

a) comma

b) Semicolon

c) Space

d) Underscore

**Ans. A, B, C**

1. **By default, all beans in Spring are singletons(page.103)**

a) True

b) False

**Ans. A**

1. **Dependency injection services that Spring offers , including**
2. Setter Injection
3. Constructor Injection
4. Method Injection
5. Getter Injection

**Ans. A, B, C**

1. **Spring’s ApplicationContext interface extends**
2. BeanFactory
3. GenericApplicatinContext
4. GenericApplication
5. None

**Ans. A**

1. **How many ways to configure Spring application context?**
2. Three
3. Four
4. Two
5. Five

**Ans. C**

1. **IoC can be decomposed into two subtypes : which are**
2. Dependency Injection
3. Dependency Lookup
4. Dependency Pull
5. Dependency Search

**Ans. A, B**

1. **How many types of Dependency Lookup?**
2. Three
3. Two
4. Four
5. Five

**Ans. B**

1. **Dependency Lookup comes into two types :**
2. Dependency Pull
3. Dependency Injection
4. Ioc Lookup
5. Contextualized Dependency Lookup

**Ans. A, D**

1. **Dependency Injection also has two common flavors : which are**
2. Constructor Dependency Injection
3. Getter Dependency Lookup
4. Setter Dependency Injection
5. Method Injection

**Ans. A, C**

1. **Dependency Pull is the most familiar type of**
2. IoC
3. DI
4. Contextualized Dependency Lookup
5. None

**Ans. A**

1. public class DepInjection{

private Dependency dependency ;

public void setDependency(Dependency dependency ){

this.dependency = dependency ;

}

}

**What type of the above code is?**

1. Constructor Dependency Injection
2. Setter Dependency Injection
3. Method Dependency Injection
4. Getter Dependency Injection

**Ans. B**

1. **Active code is much simpler and less error prone.**
2. True
3. False

**Ans. B**

1. **IoC Stands for**
2. Inversion of Control
3. Inversion of Class

**Ans. A**

1. **DI is specialized from**
2. CDL
3. IoC

**Ans. B**

1. **BeanFactory is a class**
2. True
3. False

**Ans: b**

1. **Two type of IoC container are**
2. BeanFactory
3. ApplicationContext.
4. Both a and b

**Ans: c**

1. **What is the role IoC container in Spring**
2. Create the instance
3. Configure the instance
4. Assemble the dependencies
5. None

**Ans: a, b , c**

1. **Which is the more traditional approach**
2. Dependency Injection
3. Dependency Lookup

**Ans: b**

1. **Dependency Lookup are**
2. Dependency Pull
3. Contextualized Dependency Lookup
4. None of above

**Ans: a, b**

1. **DI stands for ..**
2. Declaration interface
3. Dependency interceptor
4. Dependency Injection

**Ans: c**

1. **CDL stands for..**
2. Controller Dependency Lookup
3. Contextualized Dependency lookup
4. None

**Ans: b**

1. **BeanDefinitionRegistry is an interface**
2. True
3. False

**Ans: a**

1. **Dependency Injection service that Spring offers**
2. Setter injection.
3. Method injection.
4. Constructor injection
5. Above all

**Ans. D**

1. **IoC can be decomposed in which subtype**
2. Dependency Injection.
3. Dependency certain.
4. Dependency Lookup.
5. Dependency pull

**Ans. A, C**

1. **Which statement are correct**
2. Dependency Lookup is newer.
3. Dependency Injection is newer.
4. Dependency Lookup is more familiar to java programmers.
5. Dependency Lookup is much traditional.

**Ans. B, C, D**

1. **Function of dependency pull are**
2. Establish relation between two dependencies.
3. Make Connection to Database.
4. Dependencies are pulled from a registry as required.

**Ans. C**

1. **Dependency requirement exposed by the**

a. setDependency();

b. getDependency();

c. putDependency ()

d. dependency constructor();

**Ans. A**

1. **Passive code is much simpler to maintain than active code**
2. True.
3. False.

**Ans. A**

1. **The core of Spring’s implementation is based on**
2. Lookup injection
3. Dependency injection
4. Dependency lookup.
5. Constructor injection

**Ans. B**

1. **Bean can also instantiated without any ID and name known as**
2. BeanFactory.
3. Anonymous class.
4. Anonymous bean.
5. XmlBeanDefination.

**Ans. C**

1. **Which statement are true about namespace**
2. c: The context namespace provides support for configuring spring’s application context
3. P: The p namespace provides a simpler DI configuration for setter injection.
4. util: The util namespace provides some useful utilities for DI configuration for constructor Injection.

**Ans. B, C**

1. **Spring is described as ………………. For building java application.**
2. A lightweight framework
3. A standard framework
4. A explain framework
5. None

**Ans: a**

1. **Inversion of control (IoC) represent –**
2. Dependency lookup
3. Dependency injection
4. Traditional approach
5. Both a and b

**Ans: d**

1. **Dependency injection has two common flavors, there are**
2. Method Dependency injection
3. Constructor Dependency injection
4. Getter Dependency injection
5. Setter Dependency injection

**Ans: b, d**

1. **For xml configuration you need to declared the required-**
2. Namespace
3. Logical method
4. Dependency pull

**Ans: a**

1. **CDL means-**
2. Context dependency lookup
3. Control dependency lookup
4. Contextualized dependency lookup
5. None

**Ans: c**

1. **What are the true for configure parameter**
2. Configure parameters are passive
3. Configuration parameters are usually information not other component.
4. Configure parameters are usually simple value.
5. All of above .

**Ans: d**

1. **Advantage of Dependency injection**
2. Makes the code loosely coupled, so easy to maintain
3. Makes the code easy to test
4. Both a and b
5. None

**Ans: c**

1. **Advantage of Spring framework**
2. Powerful abstraction
3. Lightweight
4. Easy to test
5. All of above

**Ans: d**

1. **There are two types of IoC containers. They are**

a) BeanFactory

b) ApplicationContext

c) BeanContext

d) ApplicationFactory

**Ans: a, b**

1. **Which is the Artifact ID for spring aspects Module JAR File.**
2. spring-aspects
3. spring-core
4. spring-asm
5. spring-aop

**Ans: a**

1. **SpEL Means**
2. Spring Expression Language
3. Spring Exception Language
4. Spring Export Language
5. None

**Ans: a**

1. **Which tags are used in Annotation based configuration?**
2. <context: annotation-config>
3. <context: component-scan base-package=”com.exam”>
4. A and B
5. None

**Ans: c**