**Spring Core Assignment - 1**

Q1**. Create an Address Class with the following attributes:- street, city, state?**

Ans.

Config File->

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean name="Address" class="com.springcore.Address">

<property name="street" value="y"/>

<property name="city" value="d"/>

</bean>

<bean class="com.springcore.Student" name="student1">

<property name="studentId">

<value>22233</value>

</property>

<property name="studentName">

<value>Satyam</value>

</property>

<property name="studentContact">

<value>Nearby</value>

</property>

<property name="studentAddress" ref="Address"/>

</bean>

</beans>

ADRESS FILE->

package com.springcore;

public class Address {

private String street;

private String city;

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public Address(String street, String city) {

super();

this.street = street;

this.city = city;

}

public Address() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Address [street=" + street + ", city=" + city + "]";

}

}

STUDENT->

package com.springcore;

public class Student {

private int studentId;

private String studentName;

private String studentContact;

private Address studentAddress;

public int getStudentId() {

return studentId;

}

public void setStudentId(int studentId) {

this.studentId = studentId;

}

public String getStudentName() {

return studentName;

}

public void setStudentName(String studentName) {

this.studentName = studentName;

}

public String getStudentContact() {

return studentContact;

}

public void setStudentContact(String studentContact) {

this.studentContact = studentContact;

}

public Address getStudentAddress() {

return studentAddress;

}

public void setStudentAddress(Address studentAddress) {

this.studentAddress = studentAddress;

}

public Student(int studentId, String studentName, String studentContact, Address studentAddress) {

super();

this.studentId = studentId;

this.studentName = studentName;

this.studentContact = studentContact;

this.studentAddress = studentAddress;

}

public Student() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Student [studentId=" + studentId + ", studentName=" + studentName + ", studentContact=" + studentContact

+ ", studentaddress=" + studentAddress + "]";

}

}

APP->

package com.springcore;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

/\*\*

\* Hello world!

\*

\*/

public class App

{

public static void main( String[] args )

{

ApplicationContext context=new ClassPathXmlApplicationContext("com/springcore/config.xml");

Student student1 =(Student) context.getBean("student1");

System.out.println(student1);

}

}

Q2**. Example of Injecting collections?**

Ans.

APP->

package com.springcore;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

/\*\*

\* Hello world!

\*

\*/

public class App

{

public static void main( String[] args )

{

ApplicationContext context=new ClassPathXmlApplicationContext("com/springcore/config.xml");

Student student1 =(Student) context.getBean("student1");

System.out.println(student1);

}

}

STUDENT->

package com.springcore;

import java.util.List;

public class Student {

public List<String> questionId;

public List<String> question;

public List<String> answers;

public List<String> getQuestionId() {

return questionId;

}

public void setQuestionId( List<String> questionId) {

this.questionId = questionId;

}

public List<String> getQuestion() {

return question;

}

public void setQuestion( List<String> question) {

this.question = question;

}

public List<String> getAnswers() {

return answers;

}

public void setAnswers( List<String> answers) {

this.answers = answers;

}

public Student( List<String> questionId, List<String> question, List<String> answers) {

super();

this.questionId = questionId;

this.question = question;

this.answers = answers;

}

public Student() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Student [questionId=" + questionId + ", question=" + question + ", answers=" + answers + "]";

}

}

CONFIG->

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean class="com.springcore.Student" name="student1">

<property name="questionId">

<list>

<value>234324

</value

></list>

</property>

<property name="question"><list>

<value>crazy ass

</value

></list>

</property>

<property name="answers">

<list>

<value>pumpkin

</value

></list>

</property>

</bean>

</beans>

FOR SET JUST CHANGE TYPE OF STUDENT ATTRIBUTES TO Set<String>

FOR MAP

CONFIG FILE

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean class="com.springcore.Student" name="student1">

<property name="questionId">

<map >

<entry key="java" value="course"></entry>

</map>

</property>

<property name="question">

<map >

<entry key="python" value="course"></entry>

</map>

</property>

<property name="answers">

<map >

<entry key="C" value="course"></entry>

</map>

</property>

</bean>

</beans>

AND CHANGE STUDENT ATTRIBUTE TYPE TO

Map<String, String>

D. CREATE A TEST CLASS.

Test file->

package com.springcore;

import static org.junit.Assert.\*;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

public void test() {

ApplicationContext Ac= new ClassPathXmlApplicationContext("config.xml");

Student data = (Student) Ac.getBean("student1");

System.out.println(data);

}

}

REST ALL THE FILES ARE SAME

Q5. **Write a program to demonstrate use of @Resource, @inject, @Required Annotations?**

Ans.

package com.springcore;

public class Address {

private String street;

private String city;

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public String getCity() {

return city;

}

@Required

public void setCity(String city) {

this.city = city;

}

public Address(String street, String city) {

super();

this.street = street;

this.city = city;

}

public Address() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Address [street=" + street + ", city=" + city + "]";

}

}

CONFIG.XML

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean name="Address" class="com.springcore.Address">

<property name="street" value="y"/>

<property name="city" value="d"/>

</bean>

<bean class="com.springcore.Student" name="student1">

<property name="studentId">

<value>22233</value>

</property>

<property name="studentName">

<value>Satyam</value>

</property>

<property name="studentContact">

<value>Nearby</value>

</property>

<property name="studentAddress" ref="Address"/>

</bean>

<bean class="org.springframework.beans.factory.annotation.RequiredAnnotationBeanPostProcessor" />

</beans>

In above program required shows error if we dont have value in the bean class it shows error

THE RESOURCE ANNOTATION

CONFIG.xml file

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<!-- To activate the @Required annotation in spring -->

<context:annotation-config />

<bean class="org.springframework.beans.factory.annotation.CommonAnnotationBeanPostProcessor"/>

</beans>

COMPANY.java file

public class Company {

private String name;

private String location;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getLocation() {

return location;

}

public void setLocation(String location) {

this.location = location;

}

@Override

public String toString() {

return "Company [name=" + name + ", location=" + location + "]";

}

}

EMPLOYEE.java

package com.spring.pojo;

import javax.annotation.Resource;

public class Employee {

private String id;

private String name;

@Resource(name="mycompany")

private Company company;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Company getCompany() {

return company;

}

public void setCompany(Company company) {

this.company = company;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", company=" + company.toString() + "]";

}

}

APP file

package com.spring.util;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class AppMain {

@SuppressWarnings("resource")

public static void main(String[] args) {

ApplicationContext ac = new ClassPathXmlApplicationContext("resource-annotation.xml");

Employee emp = ac.getBean("myemployee", Employee.class);

System.out.println(emp.toString());

}

}

@Resource annotation is not available in newer versions .

Q6. Example of @Component, @Value , @PropertySource & Environment?

Create dbConfig.properties file which contains database configuration details like driver class name, dburl, username, password.

b. create a java class in which you have to read all properties and display on a console.

Ans

Teststereo file

**package** com.spring.core.annotations;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Teststereo {

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

ApplicationContext Ac = **new** ClassPathXmlApplicationContext("com/spring/core/annotations/stereoannotation.xml");

Student data= Ac.getBean("student",Student.**class**);

System.***out***.println(data);

}

}

Student file

**package** com.spring.core.annotations;

**import** org.springframework.beans.factory.annotation.Value;

**import** org.springframework.stereotype.Component;

@Component

**public** **class** Student {

@Value("1001")

**private** **int** studentid;

@Value("Johnny cage")

**private** String studentName;

**public** **int** getStudentid() {

**return** studentid;

}

**public** **void** setStudentid(**int** studentid) {

**this**.studentid = studentid;

}

**public** String getStudentName() {

**return** studentName;

}

**public** **void** setStudentName(String studentName) {

**this**.studentName = studentName;

}

@Override

**public** String toString() {

**return** "Student [studentid=" + studentid + ", studentName=" + studentName + "]";

}

}

Stereoannotation.xml file

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

<context:component-scan base-package=*"com.spring.core.annotations"*/>

</beans>

Q7.Java Program to demonstrate Spel?

Ans.

Student.java File//

package com.spring.core.annotations;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.context.annotation.Scope;

import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;

import org.springframework.core.env.Environment;

import org.springframework.stereotype.Component;

@Component

public class Student {

@Value("#{22+1}")

private String studentName;

private List<String> address;

public String getStudentName() {

return studentName;

}

public void setStudentName(String studentName) {

this.studentName = studentName;

}

public List<String> getAddress() {

return address;

}

public void setAddress(List<String> address) {

this.address = address;

}

@Override

public String toString() {

return "Student [studentName=" + studentName + ", address=" + address + "]";

}

}

Teststereo.java file //

**package** com.spring.core.annotations;

**import** org.springframework.expression.Expression;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** org.springframework.stereotype.Component;

**import** org.springframework.expression.spel.standard.SpelExpressionParser;

**public** **class** Teststereo {

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

ApplicationContext context = **new** ClassPathXmlApplicationContext("com/spring/core/annotations/stereoannotation.xml");

Student s1=context.getBean("student", Student.**class**);

System.***out***.println(s1);

}

}

Stereoannotation.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

<context:component-scan base-package=*"com.spring.core.annotations"*/>

</beans>

Q8. Java program to demonstrate initializingBean and DisposableBean?

Ans. Config file

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd"*>

<bean class=*"com.spring.core.annotations.Pepsi"* name=*"p1"*>

<property name=*"price"* value=*"5"*/>

</bean>

</beans>

Pepsi file

**package** com.spring.core.annotations;

**import** org.springframework.beans.factory.DisposableBean;

**import** org.springframework.beans.factory.InitializingBean;

**public** **class** Pepsi **implements** InitializingBean, DisposableBean{

**private** **double** Price;

**public** **double** getPrice() {

**return** Price;

}

**public** **void** setPrice(**double** price) {

Price = price;

}

**public** Pepsi() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Pepsi [Price=" + Price + "]";

}

**public** **void** afterPropertiesSet() **throws** Exception {

// **TODO** Auto-generated method stub

//init

System.***out***.println("taking pepsi: init");

}

**public** **void** destroy() **throws** Exception {

// **TODO** Auto-generated method stub

System.***out***.println("Going to put bottle :destroy");

}

}

Test.java

package com.spring.core.annotations;

import org.springframework.expression.Expression;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Component;

import org.springframework.expression.spel.standard.SpelExpressionParser;

public class Teststereo {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("com/spring/core/annotations/stereoannotation.xml");

Student s1=context.getBean("student", Student.class);

System.out.println(s1);

}

}

Q9. Spring program to demonstrate complete bean life cycle?

Ans.

Teststereo.java

**package** com.spring.core.annotations;

**import** org.springframework.expression.Expression;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** org.springframework.stereotype.Component;

**import** org.springframework.expression.spel.standard.SpelExpressionParser;

**public** **class** Teststereo {

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

ApplicationContext context = **new** ClassPathXmlApplicationContext("com/spring/core/annotations/stereoannotation.xml");

Student s1=context.getBean("student", Student.**class**);

System.***out***.println(s1);

}

}

Stereoannotation.java

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

<context:component-scan base-package=*"com.spring.core.annotations"*/>

<bean class=*"com.spring.core.annotations.Pepsi"* name=*"pepsi"* init-method=*"init"* destroy-method=*"destroy"* >

<property name=*"Price"* value=*"100"* />

</bean>

</beans>

Pepsi.java

**package** com.spring.core.annotations;

**import** org.springframework.beans.factory.DisposableBean;

**import** org.springframework.beans.factory.InitializingBean;

**public** **class** Pepsi{

**private** **double** Price;

**public** **double** getPrice() {

**return** Price;

}

**public** **void** setPrice(**double** price) {

Price = price;

}

**public** Pepsi() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Pepsi [Price=" + Price + "]";

}

**public** **void** init() {

System.***out***.println("initialized the method");

}

**public** **void** destroy() {

System.***out***.println("destroyed the method");

}

}

Q. Program to demonstrate ApplicationContextAware interface?

Ans.

Test.java File

**package** com.spring.core.annotations;

**import** org.springframework.beans.BeansException;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.ApplicationContextAware;

**import** org.springframework.context.support.AbstractApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Test{

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

ApplicationContext context=**new** ClassPathXmlApplicationContext("com/spring/core/annotations/config.xml");

Student triangle = (Student) context.getBean("triangle");

}

}

Student.java file

package com.spring.core.annotations;

import java.util.List;

import org.springframework.beans.BeansException;

import org.springframework.beans.factory.BeanNameAware;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.context.ApplicationContext;

import org.springframework.context.ApplicationContextAware;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.context.annotation.Scope;

import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;

import org.springframework.core.env.Environment;

import org.springframework.stereotype.Component;

public class Student implements ApplicationContextAware, BeanNameAware{

private String pointA;

private String pointB;

private String pointC;

private ApplicationContext context = null;

@Override

public String toString() {

return "Student [pointA=" + pointA + ", pointB=" + pointB + ", pointC=" + pointC + "]";

}

public Student() {

super();

// TODO Auto-generated constructor stub

}

public String getPointA() {

return pointA;

}

public void setPointA(String pointA) {

this.pointA = pointA;

}

public String getPointB() {

return pointB;

}

public void setPointB(String pointB) {

this.pointB = pointB;

}

public String getPointC() {

return pointC;

}

public void setPointC(String pointC) {

this.pointC = pointC;

}

public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {

this.context = context;

}

public void setBeanName(String beanName) {

// TODO Auto-generated method stub

System.out.println("Bean name is: "+beanName);

}

}

Config.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd"*>

<bean id=*"triangle"* class=*"com.spring.core.annotations.Student"* autowire=*"byName"*>

</bean>

</beans>