Spring Assignment

1. Create an Address class with the following attributes:- street, city, state, zip, country Create an Customer class with the following attributes:- customerld, customer Name, customerContact, customerAddress. object and print details of Customer. Inject the Address bean into Customer bean using setter injection Create a Test class with main() method, get Customer bean from ApplicationContext Also write the JUnit Test cases for above program.

* Modify the above application and inject the bean using constructor injection
* Use XML based Configuraion.

//Address Class

**package** springcore.com;

**public** **class** Address {

**private** String city, street, state, country;

**private** **int** zip;

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

**public** **int** getZip() {

**return** zip;

}

**public** **void** setZip(**int** zip) {

**this**.zip = zip;

}

}

Customer.java

**public** **class** Customer {

**private** **int** customerId;

**private** **int** customercontact;

**private** String customerName;

**private** Address customerAddress;

**public** **int** getCustomerId() {

**return** customerId;

}

**public** **void** setCustomerId(**int** customerId) {

**this**.customerId = customerId;

}

**public** **int** getCustomercontact() {

**return** customercontact;

}

**public** **void** setCustomercontact(**int** customercontact) {

**this**.customercontact = customercontact;

}

**public** Address getCustomerAddress() {

**return** customerAddress;

}

**public** **void** setCustomerAddress(Address customerAddress) {

**this**.customerAddress = customerAddress;

}

**public** String getCustomerName() {

**return** customerName;

}

**public** **void** setCustomerName(String customerName) {

**this**.customerName = customerName;

}

**public** String toString() {

**return** "Customer Id: " + getCustomerId() + ", Customer Contact: " + getCustomercontact()

+ ", Customer Address: {City: " + getCustomerAddress().getCity() + ", Street: "

+ getCustomerAddress().getStreet() + ", State: " + getCustomerAddress().getState() + ", Country: "

+ getCustomerAddress().getCountry() + ", Zip: " + getCustomerAddress().getZip() + "}, Customer Name: "

+ getCustomerName();

}

}

Test //Main

package springcore.com;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("spring.xml");

Customer customer = (Customer) context.getBean("Customer");

System.out.println(customer);

}

}

Spring.xml

<bean id = "Customer" class="springcore.com.Customer">

<property name="customer\_id" value="1"/>

<property name="customer\_name" value="John"/>

<property name="customer\_contact" value="98522"/>

<property name="street" ref="customer\_address"/>

<property name="city" ref="customer\_address"/>

<property name="state" ref="customer\_address"/>

<property name="zip" ref="customer\_address"/>

<property name="country" ref="customer\_address"/>

</bean>

<bean id="customer\_address" class="springcore.com.Address">

<property name="street" value="Baker Street"/>

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

<property name="state" value="Toronto "/>

<property name="zip" value="9854"/>

<property name="country" value="Canada"/>

</bean>

1. Example of Injecting collections (List, Set and Map)

Create a class Question with following attributes: questionid, question, answers. There are 3 cases for above program.

* Write a program where answers is of type List or String[]
* Write a program where answers is of type Set
* Write a program where answers is of type Map<Integer, String>. In case of Map, Integer value represents answer's sequence number.
* Create a Test class with main() method, get Question bean from ApplicationContext object and print question and its answers. e. Also write the JUnit Test cases for above program.

Use XML based configuration.

**Main Class Question2.java**

**package** Spring\_Core;

**import** **static** org.junit.jupiter.api.Assertions.\*;

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

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

**import** org.junit.jupiter.api.Test;

**class** Question2

{

@SuppressWarnings("unused")

**private** **static** ApplicationContext *context*;

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

{

@SuppressWarnings("resource")

ApplicationContext context = **new** ClassPathXmlApplicationContext("Coll.xml");

List list = (List) context.getBean("Collections");

System.***out***.println("List: " + list.getList());

}

}

**Question2.java**

**package** Spring\_Core;

**public** **class** Question

{

**private** **int** questionId;

**private** String question, answers;

**public** **void** setQuestionId(**int** questionId) {

**this**.questionId = questionId;

}

**public** **void** setQuestion(String question) {

**this**.question = question;

}

**public** **void** setAnswers(String answers) {

**this**.answers = answers;

}

**public** String toString()

{

**return** "Question ID: "+questionId+ ", Questions: "+question+ ", Answers: "+answers+"\n";

}

}

**List.java**

package Spring\_Core;

import java.util.ArrayList;

import java.util.Map;

import java.util.Set;

public class List

{

private ArrayList<Object> list;

private Set<Object> set;

private Map<Object, Object> map;

public ArrayList<Object> getList() {

return list;

}

public void setList(ArrayList<Object> list) {

this.list = list;

}

public Set<Object> getSet() {

return set;

}

public void setSet(Set<Object> set) {

this.set = set;

}

public Map<Object, Object> getMap() {

return map;

}

public void setMap(Map<Object, Object> map) {

this.map = map;

}

}

**Coll.xml**

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

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

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

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

xmlns:task=*"http://www.springframework.org/schema/task"* xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"Collections"* class=*"Spring\_Core.List"*>

<property name=*"list"*>

<list>

<ref bean=*"answers"* />

<ref bean=*"answers1"* />

<ref bean=*"answers2"* />

</list>

</property>

</bean>

<bean id=*"answers"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*”Teja"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"123"*/>

</bean>

<bean id=*"answers1"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Ritu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"456"*/>

</bean>

<bean id=*"answers2"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Manu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"789"*/>

</bean>

</beans>

**Main Class Question2.java**

**package** Spring\_Core;

**import** **static** org.junit.jupiter.api.Assertions.\*;

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

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

**import** org.junit.jupiter.api.Test;

**class** Question2

{

@SuppressWarnings("unused")

**private** **static** ApplicationContext *context*;

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

{

@SuppressWarnings("resource")

ApplicationContext context = **new** ClassPathXmlApplicationContext("Coll.xml");

List set = (List) context.getBean("Collections");

System.***out***.println("Set : " + set.getSet());

}

}

**Coll.xml**

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

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

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

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

xmlns:task=*"http://www.springframework.org/schema/task"* xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"Collections"* class=*"Spring\_Core.List"*>

<property name=*"set"*>

<set>

<ref bean=*"answers"* />

<ref bean=*"answers1"* />

<ref bean=*"answers2"* />

</set>

</property>

</bean>

<bean id=*"answers"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Teja"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"123"*/>

</bean>

<bean id=*"answers1"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Ritu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"456"*/>

</bean>

<bean id=*"answers2"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Manu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"789"*/>

</bean>

</beans>

**Main Class Question2.java**

**package** Spring\_Core;

**import** **static** org.junit.jupiter.api.Assertions.\*;

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

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

**import** org.junit.jupiter.api.Test;

**class** Question2

{

@SuppressWarnings("unused")

**private** **static** ApplicationContext *context*;

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

{

@SuppressWarnings("resource")

ApplicationContext context = **new** ClassPathXmlApplicationContext("Coll.xml");

List map = (List) context.getBean("Collections");

System.***out***.println("Map : " + map.getMap());

}

}

**Coll.xml**

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

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

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

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

xmlns:task=*"http://www.springframework.org/schema/task"* xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"Collections"* class=*"Spring\_Core.List"*>

<property name=*"map"*>

<map>

<entry key=*"1"* value-ref=*"answers"* />

<entry key=*"2"* value-ref=*"answers1"* />

<entry key=*"3"* value-ref=*"answers2"* />

</map>

</property>

</bean>

<bean id=*"answers"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Teja"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"123"*/>

</bean>

<bean id=*"answers1"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Ritu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"456"*/>

</bean>

<bean id=*"answers2"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Manu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"789"*/>

</bean>

</beans>

1. Write a program to demonstrate use of @Resource, @inject, @Required annotations

* @Required

Employee.java

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

**public** **class** employee {

**private** String name;

**private** String designation;

**private** String company;

**public** String getName() {

**return** name;

}

@Required

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getDesignation() {

**return** designation;

}

@Required

**public** **void** setDesignation(String designation) {

**this**.designation = designation;

}

**public** String getCompany() {

**return** company;

}

@Required

**public** **void** setCompany(String company) {

**this**.company = company;

}

@Override

**public** String toString() {

**return** "Employee [name=" + name + ", designation=" + designation + ", company=" + company + "]";

}

}

Test.java //Main

**package** springcore;

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

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

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

@SuppressWarnings("resource")

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

ApplicationContext context = **new** ClassPathXmlApplicationContext("spring.xml");

employee emp = context.getBean("myemployee", employee.**class**);

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

}

}

Spring.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"*

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"*>

<!-- To activate the '@Resource' annotation in the spring framework -->

<context:annotation-config />

<bean id=*"myemployee"* class=*"springcore.employee"*>

<property name=*"name"* value=*"John"* />

<property name=*"designation"* value=*"Manager"* />

<property name=*"company"* value=*"CG"* />

</bean>

</beans>

@resource

Company.java

**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

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() + "]";

}

}

ResourceAnnotation.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"*

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"*>

<!-- To activate the '@Resource' annotation in the spring framework -->

<context:annotation-config />

<bean id=*"mycompany"* class=*"springcore.Company"*>

<property name=*"name"* value=*"CG"* />

<property name=*"location"* value=*"India"* />

</bean>

<bean id=*"myemployee"* class=*"springcore.Employe"*>

<property name=*"id"* value=*"4661076"* />

<property name=*"name"* value=*"Johnson"* />

</bean>

</beans>

@inject

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

**private** **int** studentNo;

**private** String studentName;

**public** **int** getStudentNo() {

**return** studentNo;

}

**public** **void** setStudentNo(**int** studentNo) {

**this**.studentNo = studentNo;

}

**public** String getStudentName() {

**return** studentName;

}

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

**this**.studentName = studentName;

}

}

**package** inject;

**import** javax.inject.Inject;

**public** **class** StudentHolder {

/\* Inject annotation wires the property byType by default \*/

@Inject

Student student;

**public** Student getStudent() {

**return** student;

}

**public** **void** setStudent(Student student) {

**this**.student = student;

}

**public** **void** displayStudentDetails() {

System.***out***.println("Student Details");

System.***out***.println("---------------");

System.***out***.println("Student No: " + student.getStudentNo());

System.***out***.println("Student Name: " + student.getStudentName());

}

}

**package** inject;

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

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

**public** **class** InjectMain {

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

ApplicationContext context = **new** ClassPathXmlApplicationContext("inject.xml");

StudentHolder studentHolder = (StudentHolder) context.getBean("studentHolder");

studentHolder.displayStudentDetails();

}

}

<?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/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<bean id=*"stu"* class=*"inject.Student"*>

<property name=*"studentNo"* value=*"47716"* />

<property name=*"studentName"* value=*"John "* />

</bean>

<bean id=*"studentHolder"* class=*"inject.StudentHolder"* />

<context:annotation-config />

</beans>

1. Write a Java program to demonstrate SPEL (Spring Expression language)

import org.springframework.expression.Expression;

import org.springframework.expression.ExpressionParser;

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

public class Test {

public static void main(String[] args) {

ExpressionParser parser = new SpelExpressionParser();

Expression exp = parser.parseExpression("'Hello SPEL'");

String message = (String) exp.getValue();

System.out.println(message);

//OR

//System.out.println(parser.parseExpression("'Hello SPEL'").getValue());

}

}

1. Write a Java program to demonstrate InitializingBean and DisposableBean.

Try Different ways:

* (Use init-method and destroy-method in xml config file)

**package** q8;

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

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

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

**private** String msg;

**public** String getMsg() {

**return** msg;

}

**public** **void** setMsg(String msg) {

**this**.msg = msg;

}

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

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

System.***out***.println("Spring Container is destroy! Customer clean up");

}

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

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

System.***out***.println("Init method after properties are set : " + msg);

}

}

**package** q8;

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

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

**public** **class** TestMain {

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

ApplicationContext context = **new** ClassPathXmlApplicationContext("spring.xml");

CustomerService cs = (CustomerService) context.getBean("customerService");

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

}

}

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

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

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

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

*http://www.springframework.org/schema/beans/spring-beans-2.5.xsd"*>

<bean id=*"customerService"* class=*"q8.CustomerService"*>

<property name=*"msg"* value=*"property message"* />

</bean>

</beans>

* (Use @PostConstruct and @PreDestroy)

**package** qeight;

**import** javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

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

@Component

**public** **class** MyBean {

**public** MyBean() {

System.***out***.println("MyBean instance created");

}

@PostConstruct

**private** **void** init() {

System.***out***.println("Verifying Resources");

}

@PreDestroy

**private** **void** shutdown() {

System.***out***.println("Shutdown All Resources");

}

**public** **void** close() {

System.***out***.println("Closing All Resources");

}

}

**package** qeight;

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

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

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

@Configuration

**public** **class** MyConfiguration {

@Bean

@Scope(value = "singleton")

**public** MyBean myBean() {

**return** **new** MyBean();

}

}

**package** qeight;

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

**public** **class** App {

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

AnnotationConfigApplicationContext ctx = **new** AnnotationConfigApplicationContext();

ctx.register(MyConfiguration.**class**);

ctx.refresh();

MyBean mb1 = ctx.getBean(MyBean.**class**);

System.***out***.println(mb1.hashCode());

MyBean mb2 = ctx.getBean(MyBean.**class**);

System.***out***.println(mb2.hashCode());

ctx.close();

}

}

1. Write a Java program to demonstrate Complete Bean Life cycle.

**package** q9;

**public** **class** Hello {

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

System.***out***.println("Welcome to SpringCore ");

}

// This method executes

// when the spring container

// is closed

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

System.***out***.println("Destroy method created");

}

}

package q9;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

public static void main(String[] args) throws Exception {

// Loading the Spring XML configuration

// file into the spring container and

// it will create the instance of

// the bean as it loads into container

ConfigurableApplicationContext cap = new ClassPathXmlApplicationContext("spring9.xml");

// It will close the spring container

// and as a result invokes the

// destroy() method

cap.close();

}

}

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

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

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

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

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id=*"hw"* class=*"q9.Hello"*

init-method=*"init"* destroy-method=*"destroy"*/>

</beans>