**Assignments on Spring Core**

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

Modify the above application and inject the bean using constructor injection

Use XML based Configuraion.

**Class Address**

**package** spring\_example;

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

**private** String street;

**private** String city;

**private** String state;

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

**private** String country;

**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** String getState() {

**return** state;

}

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

**this**.state = state;

}

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

**return** zip;

}

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

**this**.zip = zip;

}

**public** String getCountry() {

**return** country;

}

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

**this**.country = country;

}

}

**Class Customer**

**package** spring\_example;

**import** java.util.List;

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

**private** **int** customerid;

**private** String customername;

**private** String customercontact;

**private** List<Address> address;

**public** List<Address> getAddress() {

**return** address;

}

**public** **void** setAddress(List<Address> address) {

**this**.address = address;

}

**public** **int** getCustomerid() {

**return** customerid;

}

**public** **void** setCustomerid(**int** customerid) {

**this**.customerid = customerid;

}

**public** String getCustomername() {

**return** customername;

}

**public** **void** setCustomername(String customername) {

**this**.customername = customername;

}

**public** String getCustomercontact() {

**return** customercontact;

}

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

**this**.customercontact = customercontact;

}

**public** **void** cust()

{

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

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customer Details\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

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

System.***out***.println("Id: "+**this**.customerid);

System.***out***.println("Name: "+**this**.customername);

System.***out***.println("Contact no: "+**this**.customercontact);

**for**(Address ad : address)

{

System.***out***.println("Address : Street:"+ad.getStreet()+", City:"+ad.getCity()+", State:"+ad.getState()+", Zipcode:"+ad.getZip()+", Country:"+ad.getCountry());

}

}

}

**Beans.xml File**

<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.xsd"*>

<bean id=*"Customer"* class=*"spring\_example.Customer"* >

<property name=*"customerid"* value=*"632"*/>

<property name=*"customername"* value=*"AJAY PRADHAN"*/>

<property name=*"customercontact"* value=*"9096660872"*/>

<property name=*"address"*>

<list>

<ref bean=*"AddressDetails"*/>

</list>

</property>

</bean>

<bean id=*"AddressDetails"* class=*"spring\_example.Address"* >

<property name=*"street"* value=*"Hingna Road"*/>

<property name=*"city"* value=*"Nagpur"*/>

<property name=*"state"* value=*"Maharashtra"*/>

<property name=*"zip"* value=*"440016"*/>

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

</bean>

</beans>

**Main Test CustTest Class**

**package** spring\_example;

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

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

**public** **class** CustTest {

@SuppressWarnings("unused")

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

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

{

@SuppressWarnings("resource")

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

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

ct.cust();

}

}

2) 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.

a. Write a program where answers is of type List<String> or String[]

b. Write a program where answers is of type Set<String>

c. Write a program where answers is of type Map<Integer, String> In case of Map, Integer value represents answer's sequence number.

d. 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.

**package** assignmentQ2;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**public** **class** Question {

**private** **int** QuestionID;

**private** String Question;

**private** List<String> Answer;

**private** Set<String> Answer1;

**private** Map<Integer,String> Answer2;

**public** Question(**int** questionID, String question, List<String> answer, Set<String> answer1, Map<Integer,String> answer2) {

**super**();

QuestionID = questionID;

Question = question;

Answer = answer;

Answer1=answer1;

Answer2=answer2;

}

**public** **void** showAnswer() {

System.***out***.println(QuestionID+ " "+ Question);

System.***out***.println("Answers from(List,Set, Map)");

System.***out***.println("Answers from(List)");

Iterator<String> i=Answer.iterator();

**while**(i.hasNext()) {

System.***out***.println(i.next());

}

System.***out***.println("Answers from(Set)");

Iterator<String> iSet=Answer1.iterator();

**while**(iSet.hasNext()) {

System.***out***.println(iSet.next());

}

System.***out***.println("Answers from(Map)");

**for**(Map.Entry<Integer, String> entry: Answer2.entrySet()) {

System.***out***.println(entry.getKey()+ " "+ entry.getValue());

}

}

}

Maintest class

**package** assignmentQ2;

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

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

**public** **class** mainTest {

@SuppressWarnings("unused")

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

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

@SuppressWarnings("resource")

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

Question q=(Question) context.getBean("Question");

q.showAnswer();

}

}

XML file

<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.xsd"*>

<bean id=*"Question"* class=*"assignmentQ2.Question"*>

<constructor-arg value=*"20"*></constructor-arg>

<constructor-arg value=*"Why Spring is Popular..?"* />

<constructor-arg>

<list>

<value>1. It Support All Framework</value>

<value>2. Spring is used for Enterprise Level Application</value>

</list>

</constructor-arg>

<constructor-arg>

<set>

<value>1. All the frameworks are supported by spring like Structs,Hibernate and Jsf.</value>

<value>2. simple learning curves</value>

</set>

</constructor-arg>

<constructor-arg>

<map>

<entry key=*"1"* value=*"Spring Framework"* />

<entry key=*"2"* value=*"Spring MVC Framework"* />

</map>

</constructor-arg>

</bean>

</beans>

7.write a java program to demonstrate SPEL(Spring Expression language).

**package** assignmentQ2;

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

**import** org.springframework.expression.ExpressionParser;

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

**public** **class** dg {

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

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

ExpressionParser parser = **new** SpelExpressionParser();

Expression exp = parser.parseExpression("'Ajay Pradhan'");

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

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

//OR

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

}

}

10.Write a program to demonstrate ApplicationContextAware Interface

**package** spring\_example;

**import** java.util.List;

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

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

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

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

**public** **class** Customer **implements** ApplicationContextAware, BeanNameAware {

**private** **int** customerid;

**private** String customername;

**private** String customercontact;

**private** ApplicationContext context=**null**;

**private** List<Address> address;

**public** List<Address> getAddress() {

**return** address;

}

**public** **void** setAddress(List<Address> address) {

**this**.address = address;

}

**public** **int** getCustomerid() {

**return** customerid;

}

**public** **void** setCustomerid(**int** customerid) {

**this**.customerid = customerid;

}

**public** String getCustomername() {

**return** customername;

}

**public** **void** setCustomername(String customername) {

**this**.customername = customername;

}

**public** String getCustomercontact() {

**return** customercontact;

}

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

**this**.customercontact = customercontact;

}

**public** **void** cust()

{

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

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customer Details\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

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

System.***out***.println("Id: "+**this**.customerid);

System.***out***.println("Name: "+**this**.customername);

System.***out***.println("Contact no: "+**this**.customercontact);

**for**(Address ad : address)

{

System.***out***.println("Address : Street:"+ad.getStreet()+", City:"+ad.getCity()+", State:"+ad.getState()+", Zipcode:"+ad.getZip()+", Country:"+ad.getCountry());

}

}

**public** **void** setApplicationContext(ApplicationContext Context) **throws** BeansException {

**this**.context=context;

}

**public** **void** setBeanName(String beanName) {

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

}

}

Main class

package spring\_example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

@SuppressWarnings("unused")

private static ApplicationContext context;

public static void main(String a[])

{

@SuppressWarnings("resource")

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

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

ct.cust();

}

}

5. write a program to demonstrate @resource @inject @required annotation in spring core java

**package** assignmentQ2;

**import** java.annotation.Resource;

**public** **class** Employee {

**private** String id;

**private** String name;

**private** String address;

@Required

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

**this**.name = name;

}

**public** String getName() {

**return** name;

}

@Inject

**private** Address address;

@Resource(name="mycompany")

**private** Company company;

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** Company getCompany() {

**return** company;

}

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

**this**.company = company;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

@Override

**public** String toString() {

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

}

}

Mainclass

**package** assignmentQ2;

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

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

**public** **class** ghrtf {

@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());

}

}

XML file

<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.xsd"*>

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

<bean id=*"mycompany"* class=*"com.spring.pojo.Company"*>

<property name=*"name"* value=*"Test Pvt. Ltd."* />

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

</bean>

<bean id=*"myemployee"* class=*"com.spring.pojo.Employee"*>

<property name=*"id"* value=*"123456"* />

<property name=*"name"* value=*"Charlotte O' Neil"* />

</bean>

<bean

class=*"org.springframework.beans.factory.*

*annotation.AutowiredAnnotationBeanPostProcessor"*>

</bean>

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

</bean>

</beans>

9.Write a program to demonstrate Complete life cycle

Triangle

**package** spring\_example;

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

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

**public** **class** triangle **implements** InitializingBean, DisposableBean

{

**private** Point pointA;

**private** Point pointB;

**private** Point pointC;

**public** **void** setPointA(Point pointA) {

**this**.pointA = pointA;

}

**public** **void** setPointB(Point pointB) {

**this**.pointB = pointB;

}

**public** **void** setPointC(Point pointC) {

**this**.pointC = pointC;

}

**public** **void** draw()

{

System.***out***.println("PointA is ("+pointA.getX()+", "+pointA.getY()+")");

System.***out***.println("PointB is ("+pointB.getX()+", "+pointB.getY()+")");

System.***out***.println("PointC is ("+pointC.getX()+", "+pointC.getY()+")");

}

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

{

System.***out***.println("InitializingBean init method is called for Triangle");

}

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

{

System.***out***.println("DisposableBean destroy method is called for Triangle");

}

**public** **void** myInit()

{

System.***out***.println("My init method is called for Triangle");

}

**public** **void** cleanUp()

{

System.***out***.println("cleanUp method is called for Triangle");

}

}

Point

**package** spring\_example;

**public** **class** Point

{

**private** **int** x;

**private** **int** y;

**public** **int** getX()

{

**return** x;

}

**public** **void** setX(**int** x)

{

**this**.x = x;

}

**public** **int** getY()

{

**return** y;

}

**public** **void** setY(**int** y)

{

**this**.y = y;

}

}

DrawingApp

package spring\_example;

import org.springframework.context.support.AbstractApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class DrawingApp

{

public static void main(String[] args)

{

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

context.registerShutdownHook();

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

triangle.draw();

}

}

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"*

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

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

<bean autowire=*"byName"* class=*"com.bean2.Triangle"* destroy-method=*"cleanUp"* id=*"triangle"* init-method=*"myInit"*>

</bean>

<bean class=*"com.bean2.Point"* id=*"pointA"*>

<property name=*"x"* value=*"0"*></property>

<property name=*"y"* value=*"0"*></property>

</bean>

<bean class=*"com.bean2.Point"* id=*"pointB"*>

<property name=*"x"* value=*"-20"*></property>

<property name=*"y"* value=*"0"*></property>

</bean>

<bean class=*"com.bean2.Point"* id=*"pointC"*>

<property name=*"x"* value=*"20"*></property>

<property name=*"y"* value=*"0"*></property>

</bean>

</beans>

8) Write a Java Program to demonstrate initializingBean and DispasableBean

Part A>

package spring\_example;

import org.springframework.beans.factory.DisposableBean;

import org.springframework.beans.factory.InitializingBean;

public class Employee implements InitializingBean,DisposableBean{

private String name;

private int age;

public Employee() {

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public void afterPropertiesSet() throws Exception{

System.out.println("Initializing ");

}

public void destroy() throws Exception{

System.out.println("Destroying Employee Bean");

}

public void display() {

System.out.println("Name of Employee is : "+this.getName());

System.out.println("Age is : "+this.getAge());

}

}

**package** spring\_example;

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

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

**public** **class** Point {

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

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

context.registerShutdownHook();

Employee emp=(Employee) context.getBean("employee");

emp.display();

}

}

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"*

xsi:schemaLocation=*"*

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

<bean id=*"employee"* class=*"spring\_example.Employee"*>

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

<property name=*"age"* value=*"27"*/>

</bean>

</beans>

Part B>

**package** spring\_example;

//import javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

**public** **class** Employee2 {

**private** String name;

**private** **int** age;

**public** Employee2() {

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@PostConstruct

**public** **void** initializing() {

System.***out***.println("Initializing Employee Been");

}

//@PreDestroy

**public** **void** destroying() {

System.***out***.println("Destroying Employee Bean");

}

}

Main class Point

**package** spring\_example;

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

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

**public** **class** Point {

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

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

context.registerShutdownHook();

Employee2 emp=(Employee2) context.getBean("employee");

System.***out***.println("Employee name: "+emp.getName());

System.***out***.println("Age is: "+emp.getAge());

}

}

XML file

<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:annotation-config/>

<bean id=*"employee"* class=*"spring\_example.Employee2"*>

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

<property name=*"age"* value=*"27"*/>

</bean>

</beans>