**Spring core Assignment**

**1)**

**Address.java**

package com.asus;  
  
public class Address {  
 private String street;  
 private String city;  
 private String state;  
 private long zip;  
 private String country;  
  
 public Address() {  
 super();  
 }  
  
 public Address(String street, String city, String state, long zip, String country) {  
 this.street = street;  
 this.city = city;  
 this.state = state;  
 this.zip = zip;  
 this.country = 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 long getZip() {  
 return zip;  
 }  
  
 public void setZip(long zip) {  
 this.zip = zip;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 @Override  
 public String toString() {  
 return "Address: [" +  
 "street='" + street + '\'' +  
 ", city='" + city + '\'' +  
 ", state='" + state + '\'' +  
 ", zip=" + zip +  
 ", country='" + country + '\'' +  
 ']';  
 }  
}

**Customer.java**

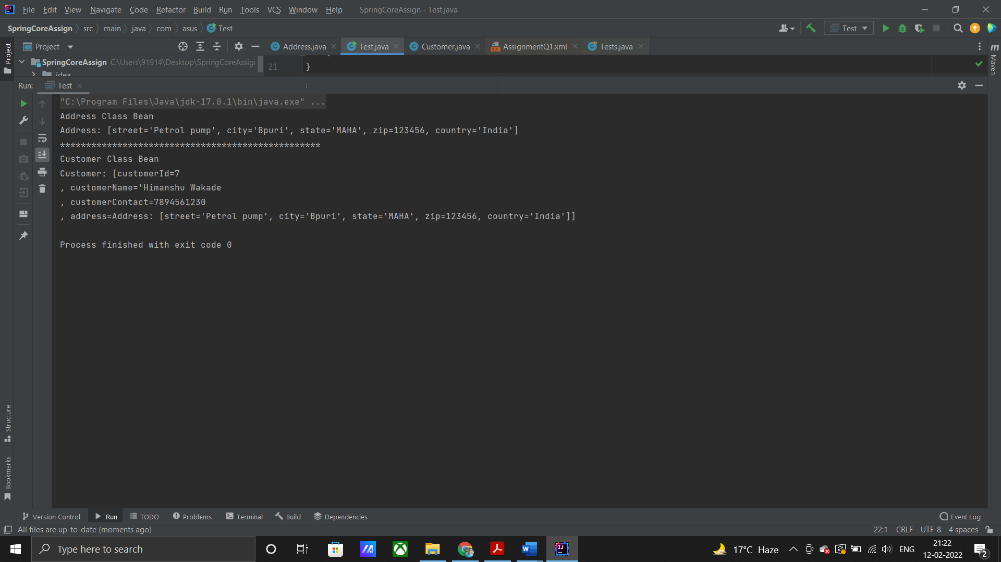
package com.asus;  
  
public class Customer {  
 private int customerId;  
 private String customerName;  
 private long customerContact;  
 private Address address;  
  
 public Customer() {  
 super();  
 }  
  
 public Customer(int customerId, String customerName, long customerContact, Address address) {  
 this.customerId = customerId;  
 this.customerName = customerName;  
 this.customerContact = customerContact;  
 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 long getCustomerContact() {  
 return customerContact;  
 }  
  
 public void setCustomerContact(long customerContact) {  
 this.customerContact = customerContact;  
 }  
  
 public Address getAddress() {  
 return address;  
 }  
  
 public void setAddress(Address address) {  
 this.address = address;  
 }  
  
 @Override  
 public String toString() {  
 return "Customer: [" +  
 "customerId=" + customerId + "\n" +  
 ", customerName='" + customerName + "\n" +  
 ", customerContact=" + customerContact + "\n" +  
 ", address=" + address +  
 ']';  
 }  
}

**Test.java**

package com.asus;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class Test {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("AssignmentQ1.xml");  
  
 Address address1 = context.getBean("address1", Address.class);  
 System.*out*.println("Address Class Bean");  
  
 System.*out*.println(address1);  
  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
  
 Customer customer1 = context.getBean("customer1", Customer.class);  
 System.*out*.println("Customer Class Bean ");  
 System.*out*.println(customer1);  
 }  
}

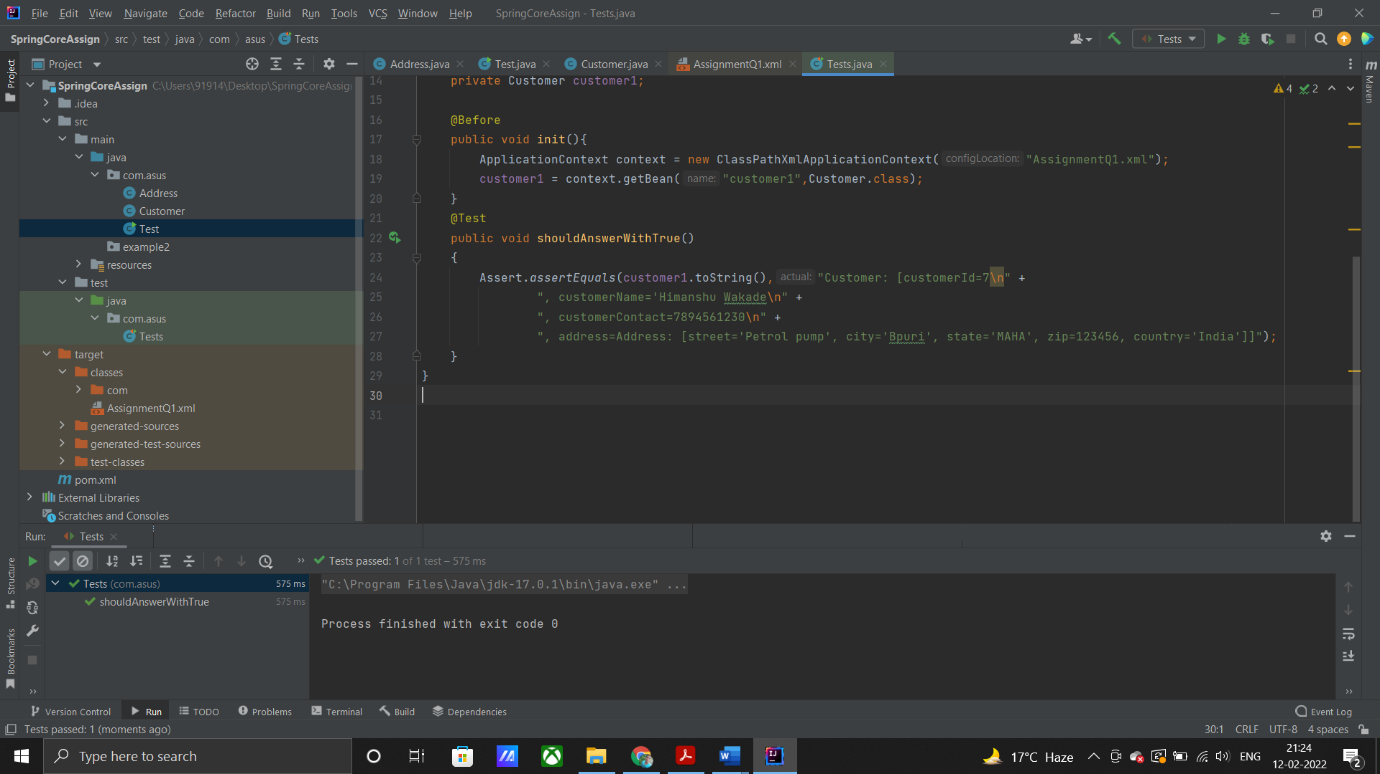
**AssignmnetQ1.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns="http://www.springframework.org/schema/beans"  
 xmlns:util="http://www.springframework.org/schema/util"  
 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/util  
 http://www.springframework.org/schema/util/spring-util.xsd">  
  
 <!-- creating bean for Address class object-->  
 <bean class="com.asus.Address" name="address1">  
 <property name="street">  
 <value>Petrol pump</value>  
 </property>  
 <property name="city">  
 <value>Bpuri</value>  
 </property>  
 <property name="state">  
 <value>MAHA</value>  
 </property>  
 <property name="zip">  
 <value>123456</value>  
 </property>  
 <property name="country">  
 <value>India</value>  
 </property>  
 </bean>  
  
 <!-- creating bean for customer class object-->  
 <bean name="customer1" class="com.asus.Customer">  
 <property name="customerId">  
 <value>07</value>  
 </property>  
 <property name="customerName">  
 <value>Himanshu Wakade</value>  
 </property>  
 <property name="customerContact">  
 <value>7894561230</value>  
 </property>  
 <property name="address">  
 <ref bean="address1"/>  
 </property>  
 </bean>  
  
 <!-- creating beans using constructor injections-->  
 <!-- creating bean for address-->  
 <bean class="com.asus.Address" name="address2">  
 <constructor-arg name="street" value="Kurud"/>  
 <constructor-arg name="city" value="Navegaon"/>  
 <constructor-arg name="state" value="Maha"/>  
 <constructor-arg name="zip" value="203010"/>  
 <constructor-arg name="country" value="India"/>  
 </bean>  
  
 <!-- creating bean for customer-->  
 <bean name="customer2" class="com.asus.Customer">  
 <constructor-arg name="customerId" value="14"/>  
 <constructor-arg name="customerName" value="DJ"/>  
 <constructor-arg name="customerContact" value="9632587410"/>  
 <constructor-arg name="address" ref="address2"/>  
 </bean>  
</beans>

****

**Tests.java**

package com.asus;  
  
import static org.junit.Assert.*assertTrue*;  
  
import com.asus.Customer;  
import org.junit.Assert;  
import org.junit.Before;  
import org.junit.Test;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class Tests  
{  
 private Customer customer1;  
  
 @Before  
 public void init(){  
 ApplicationContext context = new ClassPathXmlApplicationContext("AssignmentQ1.xml");  
 customer1 = context.getBean("customer1",Customer.class);  
 }  
 @Test  
 public void shouldAnswerWithTrue()  
 {  
 Assert.*assertEquals*(customer1.toString(),"Customer: [customerId=7\n" +  
 ", customerName='Himanshu Wakade\n" +  
 ", customerContact=7894561230\n" +  
 ", address=Address: [street='Petrol pump', city='Bpuri', state='MAHA', zip=123456, country='India']]");  
 }  
}

****

**2)**

**Question1.java**

package example2;  
  
import java.util.List;  
  
public class Question1 {  
 private int questionId;  
 private String question;  
 private List<String> answers;  
  
 public Question1() {  
 super();  
 }  
  
 public int getQuestionId() {  
 return questionId;  
 }  
  
 public void setQuestionId(int questionId) {  
 this.questionId = questionId;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public void setQuestion(String question) {  
 this.question = question;  
 }  
  
 public List<String> getAnswers() {  
 return answers;  
 }  
  
 public void setAnswers(List<String> answers) {  
 this.answers = answers;  
 }  
  
 @Override  
 public String toString() {  
 return "Question1: [" +  
 "questionId=" + questionId + '\n' +  
 ", question='" + question + '\n' +  
 ", answers=" + answers + '\n' +  
 ']';  
 }  
}

**.xml**

<bean class="example2.Question1" name="question1">  
 <property name="questionId" value="1"/>  
 <property name="question" value="What is Spring Framework?"/>  
 <property name="answers">  
 <list>  
 <value>Spring is a powerful open-source, loosely coupled, lightweight, java framework meant for reducing the complexity of developing enterprise-level applications.</value>  
 <value>This framework is also called the “framework of frameworks” as spring provides support to various other important frameworks like JSF, Hibernate, Structs, EJB, etc.</value>  
 <value>Spring handles all the infrastructure-related aspects which lets the programmer focus mostly on application development.</value>  
 </list>  
 </property>  
</bean>

**Question2.java**

package example2;  
  
import java.util.Set;  
  
public class Question2 {  
 private int questionId;  
 private String question;  
 private Set<String> answers;  
  
 public Question2() {  
 super();  
 }  
  
 public int getQuestionId() {  
 return questionId;  
 }  
  
 public void setQuestionId(int questionId) {  
 this.questionId = questionId;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public void setQuestion(String question) {  
 this.question = question;  
 }  
  
 public Set<String> getAnswers() {  
 return answers;  
 }  
  
 public void setAnswers(Set<String> answers) {  
 this.answers = answers;  
 }  
  
 @Override  
 public String toString() {  
 return "Question2: [" +  
 "questionId=" + questionId +  
 ", question='" + question + '\'' +  
 ", answers=" + answers +  
 ']';  
 }  
}

**1.xml**

<bean class="example2.Question2" name="question2">  
 <property name="questionId" value="2"/>  
 <property name="question" value="What are the features of Spring Framework?"/>  
 <property name="answers">  
 <set>  
 <value>Spring is a lightweight, java based, loosely coupled framework.</value>  
 <value>Spring provides generic abstraction layer for transaction management that is also very useful for container-less environments.</value>  
 </set>  
 </property>  
</bean>

**Question3.java**

package example2;  
  
import java.util.Map;  
  
public class Question3 {  
 private int questionId;  
 private String questionName;  
 private Map<Integer,String> answers;  
  
 public Question3() {  
 super();  
 }  
  
 public int getQuestionId() {  
 return questionId;  
 }  
  
 public void setQuestionId(int questionId) {  
 this.questionId = questionId;  
 }  
  
 public String getQuestionName() {  
 return questionName;  
 }  
  
 public void setQuestionName(String questionName) {  
 this.questionName = questionName;  
 }  
  
 public Map<Integer, String> getAnswers() {  
 return answers;  
 }  
  
 public void setAnswers(Map<Integer, String> answers) {  
 this.answers = answers;  
 }  
  
 @Override  
 public String toString() {  
 return "Question3: [" +  
 "questionId=" + questionId +'\n' +  
 ", questionName='" + questionName + '\n' +  
 ", answers=" + answers +'\n' +  
 ']';  
 }  
}

**3.xml**

<bean class="example2.Question3" name="question3">  
 <property name="questionId" value="3"/>  
 <property name="questionName" value="Explain the difference between constructor and setter injection?"/>  
 <property name="answers">  
 <map>  
 <entry key="1">  
 <value>In constructor injection, partial injection is not allowed whereas it is allowed in setter injection.</value>  
 </entry>  
 <entry key="2">  
 <value>The constructor injection doesn’t override the setter property whereas the same is not true for setter injection.</value>  
 </entry>  
 <entry key="3">  
 <value>Constructor injection creates a new instance if any modification is done. The creation of a new instance is not possible in setter injection.</value>  
 </entry>  
 <entry key="4">  
 <value>In case the bean has many properties, then constructor injection is preferred. If it has few properties, then setter injection is preferred.</value>  
 </entry>  
  
 </map>  
 </property>  
</bean>

**Test.java**

package example2;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class Test {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext(".xml");  
  
 Question1 question1 = context.getBean("question1",Question1.class);  
 System.*out*.println("1.A program where answers is of type List<String> or String []");  
  
 System.*out*.println(question1);  
  
  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
  
 Question2 question2 = context.getBean("question2",Question2.class);  
 System.*out*.println("A program where answers is of type Set<String>");  
 System.*out*.println(question2);  
  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
  
 Question3 question3 = context.getBean("question3",Question3.class);  
 System.*out*.println("A program where answers is of type Map<Integer, String>");  
 System.*out*.println(question3);  
 }  
}

**Question4)**

**GreetingBot.java**

package exampl3;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.annotation.\*;  
  
@Configuration  
@ComponentScan  
public class Main {  
  
 @Bean  
 GreetingService mockGreetingService() {  
 return new GreetingService() {  
 public String greet() {  
 return "This is my First Annotation program using Springcore";  
 }  
 };  
 }  
  
 public static <GreetingBot> void main(String[] args) {  
 ApplicationContext context = new AnnotationConfigApplicationContext(  
 Main.class);  
 GreetingBot greeter = context.getBean(GreetingBot.class);  
 greeter.print();  
 }  
}

**GreetingService.java**

package exampl3;  
  
public interface GreetingService{  
 String greet();  
}

**Main.java**

package exampl3;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.annotation.\*;  
  
@Configuration  
@ComponentScan  
class Main {  
  
 @Bean  
 GreetingService mockGreetingService() {  
 return new GreetingService() {  
 public String greet() {  
 return "This is my First Annotation program using Springcore";  
 }  
 };  
 }  
  
 public static void main(String[] args) {  
 ApplicationContext context = new AnnotationConfigApplicationContext(  
 Main.class);  
 GreetingBot greeter = context.getBean(GreetingBot.class);  
 greeter.print();  
 }  
}

**Question 8 & 9**

**CustomLifeCycleMethodBean.java**

package exam89;  
  
class CustomLifeCycleMethodBean {  
 private String name;  
  
 public CustomLifeCycleMethodBean()  
 {  
 System.*out*.println("Constructor of bean is called !! ");  
 }  
  
 public void customDestroy() throws Exception {  
  
 System.*out*.println("custom destroy method of bean is called !! ");  
 }  
  
 public void customInit() throws Exception {  
 System.*out*.println("custom Init method of bean is called !! ");  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

**c.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"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">  
  
 <bean id="customLifeCycleMethodBean"  
 class="CustomLifeCycleMethodBean"  
 init-method="customInit"  
 destroy-method="customDestroy">  
 <property name="name" value="custom methods bean" ></property>  
 </bean>  
</beans>

**TestCustomMethodLifeCycleBea.java**

package exam89;  
  
  
  
  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.AbstractApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class TestCustomMethodLifeCycleBea {  
 public static void main(String[] args) {  
  
 ApplicationContext context =  
 new ClassPathXmlApplicationContext("c.xml");  
  
 CustomLifeCycleMethodBean bean = (CustomLifeCycleMethodBean)context.getBean("customLifeCycleMethodBean");  
 ((AbstractApplicationContext) context).registerShutdownHook();  
 }  
}