# Question 1 - Hands on 2: Spring Core – Load SimpleDateFormat from Spring Configuration XML

Code 1: date-format.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  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="dateFormat" class="java.text.SimpleDateFormat">  
 <constructor-arg value="dd/MM/yyyy" />  
 </bean>  
  
</beans>

Code 2: SpringLearnApplication.java

package com.cognizant.springlearn;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
 public static void main(String[] args) throws ParseException {  
 displayDate();  
 }  
  
 public static void displayDate() throws ParseException {  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
 Date date = format.parse("31/12/2018");  
 System.out.println("Parsed Date: " + date);  
 }  
}

Output:  
Parsed Date: Mon Dec 31 00:00:00 IST 2018

# Question 2 - Hands on 4: Spring Core – Load Country from Spring Configuration XML

Code 1: country.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  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
  
</beans>

Code 2: Country.java

package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);  
  
 private String code;  
 private String name;  
  
 public Country() {  
 LOGGER.debug("Inside Country Constructor.");  
 }  
  
 public String getCode() {  
 LOGGER.debug("Inside getCode()");  
 return code;  
 }  
  
 public void setCode(String code) {  
 LOGGER.debug("Inside setCode()");  
 this.code = code;  
 }  
  
 public String getName() {  
 LOGGER.debug("Inside getName()");  
 return name;  
 }  
  
 public void setName(String name) {  
 LOGGER.debug("Inside setName()");  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

Code 3: SpringLearnApplication.java (Display Country)

package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 displayCountry();  
 }  
  
 public static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 LOGGER.debug("Country : {}", country.toString());  
 }  
}

Output:  
DEBUG com.cognizant.springlearn.Country - Inside Country Constructor.  
DEBUG com.cognizant.springlearn.Country - Inside setCode()  
DEBUG com.cognizant.springlearn.Country - Inside setName()  
DEBUG com.cognizant.springlearn.Country - Inside toString()  
DEBUG com.cognizant.springlearn.SpringLearnApplication - Country : Country [code=IN, name=India]