**Hands on 5 Spring Core – Demonstration of Singleton Scope and Prototype Scope**

**Main.java:**

**package com.cognizant.spring\_learn;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

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

**public class Main {**

**private static final Logger *LOGGER* = LoggerFactory.*getLogger*(Main.class);**

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

***LOGGER*.info("START");**

**try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("spring.xml")) {**

**Country singleton1 = context.getBean("country", Country.class);**

**Country singleton2 = context.getBean("country", Country.class);**

**System.*out*.println("Singleton 1: " + singleton1);**

**System.*out*.println("Singleton 2: " + singleton2);**

**System.*out*.println("Are singleton objects same? " + (singleton1 == singleton2));**

**Country prototype1 = context.getBean("countryPrototype", Country.class);**

**Country prototype2 = context.getBean("countryPrototype", Country.class);**

**System.*out*.println("Prototype 1: " + prototype1);**

**System.*out*.println("Prototype 2: " + prototype2);**

**System.*out*.println("Are prototype objects same? " + (prototype1 == prototype2));**

**}**

***LOGGER*.info("END");**

**}**

**}**

**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"**

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

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

**<bean id="country" class="com.cognizant.spring\_learn.Country" scope="singleton">**

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

**<property name="code" value="IN"/>**

**</bean>**

**<bean id="countryPrototype" class="com.cognizant.spring\_learn.Country" scope="prototype">**

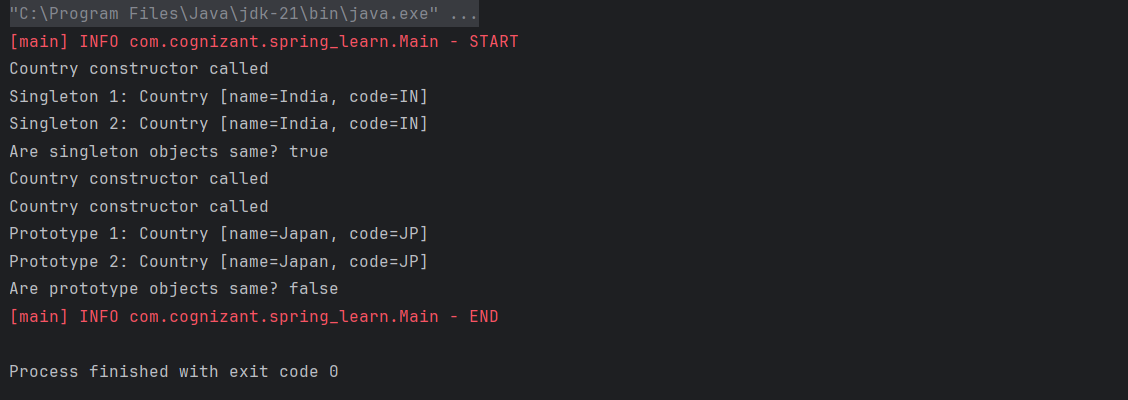
**<property name="name" value="Japan"/>**

**<property name="code" value="JP"/>**

**</bean>**

**</beans>**

**Output:**

****