Spring REST Hands-On Answers

# Hands-on 1: Create a Spring Web Project using Maven

1. Go to https://start.spring.io/.  
2. Set Group to "com.cognizant" and Artifact Id to "spring-learn".  
3. Select dependencies: Spring Boot DevTools and Spring Web.  
4. Download and extract the ZIP to Eclipse Workspace.  
5. Run build command with proxy setup:  
 mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456  
6. Import the Maven project into Eclipse.  
7. Add logs to verify execution of main() method in SpringLearnApplication.  
8. Run SpringLearnApplication.java.  
9. Explain project structure (src/main/java, src/main/resources, etc.).  
10. Use @SpringBootApplication and explain it combines @Configuration, @EnableAutoConfiguration, and @ComponentScan.  
11. Use Dependency Hierarchy to inspect dependencies in pom.xml.

# Hands-on 2: Load SimpleDateFormat from Spring Configuration XML

1. Create date-format.xml in src/main/resources.  
2. Define the bean for SimpleDateFormat with value "dd/MM/yyyy".  
3. In displayDate() method in SpringLearnApplication.java:  
 - Load ApplicationContext using ClassPathXmlApplicationContext.  
 - Retrieve the bean using getBean().  
 - Parse date string and log result.  
4. Run the application and verify date output.

# Hands-on 3: Incorporate Logging

1. Create application.properties file (if not already created).  
2. Add logging configuration:  
 - logging.level.org.springframework=info  
 - logging.level.com.cognizant.springlearn=debug  
 - logging.pattern.console=<pattern>  
3. In SpringLearnApplication.java:  
 - Add static LOGGER using LoggerFactory.  
 - Use LOGGER.info and LOGGER.debug in displayDate().  
4. Replace System.out.println() with logging.

# Hands-on 4: Load Country Bean from XML

1. Define bean for country in country.xml with properties: code="IN", name="India".  
2. Create Country class with fields, constructor (log message), getters/setters (log message), and toString().  
3. Create method displayCountry() in SpringLearnApplication:  
 - Load context and get "country" bean.  
 - Log the country details.  
4. Call displayCountry() in main method and observe logs.

# Hands-on 5: Singleton vs Prototype Scope

1. Load the same "country" bean twice in displayCountry().  
2. Observe constructor call once: Singleton scope.  
3. Change scope to prototype in country.xml:  
 - scope="prototype"  
4. Run again, observe constructor called twice: Prototype scope.

# Hands-on 6: Load List of Countries from XML

1. Define 4 beans (in, us, de, jp) in country.xml.  
2. Create a list bean using ArrayList with <list> and <ref> to the country beans.  
3. In displayCountries() method:  
 - Load "countryList" bean.  
 - Use LOGGER.debug to print all countries.  
4. Ensure logging for start and end of method.