

Section 3: Spring Exercises

Module 11: Dependency Injection

Exercise 11.1 – Spring Hello World

The Setup:

The main objective of this exercise is to test Spring by creating a simple Spring application that prints “*Hello World*” to the console.

Create a new Maven Java project called **exercise11_1**. Then open the pom.xml and add the following dependencies:

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>4.0.2.RELEASE</version>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

The Exercise:

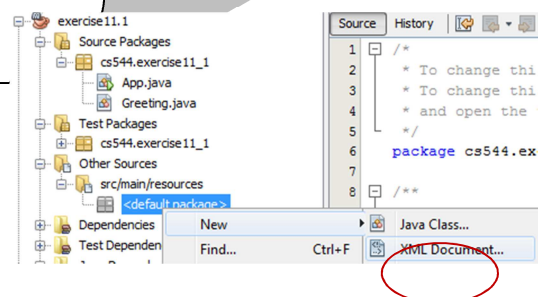
Create a class called Greeting with the following contents:

```
package cs544.exercise11_1;

public class Greeting {
    private String greeting;

    public Greeting() {}
    public Greeting(String greeting) {
        this.greeting=greeting;
    }
    public void sayHello() {
        System.out.println(greeting);
    }
    public void setGreeting(String greeting) {
        this.greeting = greeting;
    }
}
```

Next create an XML file called **springconfig.xml** inside **Other Sources** → **src/main/resources** → **default package**



Your springconfig.xml file should contain the following 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-4.0.xsd">

    <bean id="greetingService" class="cs544.exercisell_1.Greeting">
        <property name="greeting" value="Hello World" / >
    </bean>
</beans>
```

Then update the automatically generated App.java to hold the following code:3

```
package cs544.exercisell_1;

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

public class HelloApp {

    public static void main(String[] args) {
        ConfigurableApplicationContext context = new
            ClassPathXmlApplicationContext("springconfig.xml");
        GreetingService =
            context.getBean("greetingService", Greeting.class);
        greetingService.sayHello();
    }
}
```

You can run App.java, and it should give the following output:

```
log4j:WARN No appenders could be found for logger
(org.springframework.core.env.StandardEnvironment).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Hello World
```

To get rid of the log4j warnings add the following log4j.properties file to **Other Sources** → **src/main/resources** → **default package**

```
### direct log messages to stdout ###
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

### set log levels - for more verbose logging change 'info' to 'debug' ###
log4j.rootLogger=warn, stdout
```