



CS544 EA  
Spring

Startup: Creation Order

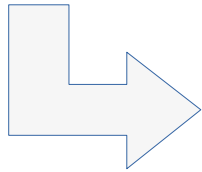
# Creation Order

- Beans are created in the order they are found
  - Except if dependency injection needs them before that
- For the different configuration options:
  - XML: the order of <bean> definitions in the file
  - Annotations: the alphabetical order inside the package
    - the order in which component scanning finds them
  - Java Config: the order of @Bean methods
    - inside the @Configuration class

# XML order Demo 1/3

```
<?xml version="1.0" encoding="UTF-8"?>
<beans ...>

  <bean id="dataSource" class="cs544.spring31.startup.order.DataSource" />
  <bean id="customerDao" class="cs544.spring31.startup.order.CustomerDao" >
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="orderDao" class="cs544.spring31.startup.order.OrderDao" parent="abstractDao" >
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="productDao" class="cs544.spring31.startup.order.ProductDao" parent="abstractDao" >
    <property name="dataSource" ref="dataSource"/>
  </bean>
</beans>
```



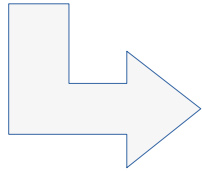
DataSource constructor  
CustomerDao Constructor  
OrderDao constructor  
ProductDao Constructor

Creation order is based on order in XML  
DataSource created once, injected 3 times

# XML order Demo 2/3

```
<?xml version="1.0" encoding="UTF-8"?>
<beans ...>

  <bean id="customerDao" class="cs544.spring31.startup.order.CustomerDao" >
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="orderDao" class="cs544.spring31.startup.order.OrderDao">
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="productDao" class="cs544.spring31.startup.order.ProductDao">
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="dataSource" class="cs544.spring31.startup.order.DataSource" />
</beans>
```



CustomerDao Constructor  
DataSource constructor  
OrderDao constructor  
ProductDao Constructor

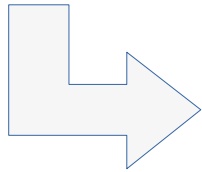
DataSource moved to bottom

DataSource still created 2<sup>nd</sup>  
Because needed for Setter Injection

# XML order Demo 3/3

```
<?xml version="1.0" encoding="UTF-8"?>
<beans ...>

  <bean id="productDao" class="cs544.spring31.startup.order.ProductDao">
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="customerDao" class="cs544.spring31.startup.order.CustomerDao" >
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="orderDao" class="cs544.spring31.startup.order.OrderDao">
    <property name="dataSource" ref="dataSource"/>
  </bean>
  <bean id="dataSource" class="cs544.spring31.startup.order.DataSource" />
</beans>
```



ProductDao Constructor  
DataSource constructor  
CustomerDao Constructor  
OrderDao constructor

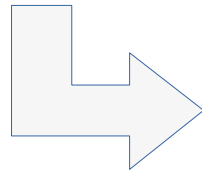
ProductDao moved to Top

ProductDao made first

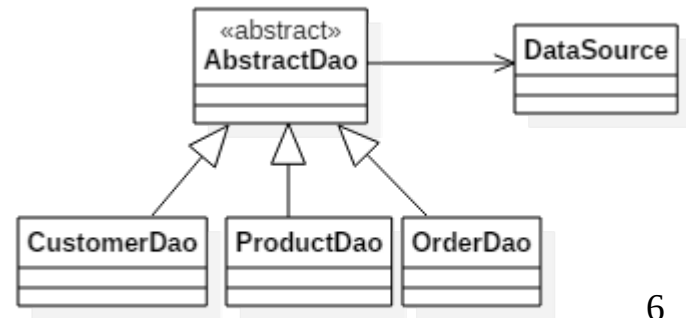
# Inheritance Creation

- Using the same example as before
  - Adding println() to each constructor

```
<bean id="abstractDao" class="cs544.spring30.startup.inherit.AbstractDao" abstract="true">  
  <property name="dataSource" ref="dataSource"/>  
</bean>  
<bean id="customerDao" class="cs544.spring30.startup.inherit.CustomerDao" parent="abstractDao" />  
<bean id="productDao" class="cs544.spring30.startup.inherit.ProductDao" parent="abstractDao" />  
<bean id="orderDao" class="cs544.spring30.startup.inherit.OrderDao" parent="abstractDao" />
```



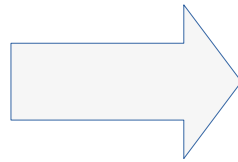
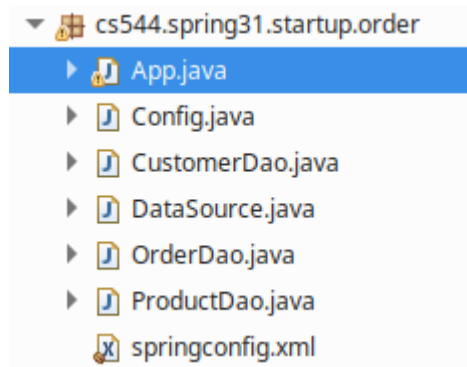
AbstractDao constructor  
CustomerDao Constructor  
AbstractDao constructor  
ProductDao Constructor  
AbstractDao constructor  
OrderDao constructor



Although no object of AbstractDao is ever made, when an object of a subclass is made, it calls **super()**

# Annotation Order Demo

- Order beans are found inside the package



CustomerDao Constructor  
DataSource constructor  
OrderDao constructor  
ProductDao Constructor

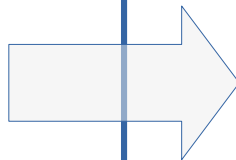
DataSource would be made second  
regardless of order  
because of DI into CustomerDao

# Java Config Order

```
package cs544.spring31.startup.order;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class Config {
    @Bean
    public DataSource dataSource() {
        return new DataSource();
    }
    @Bean
    public ProductDao productDao() {
        return new ProductDao();
    }
    @Bean
    public OrderDao orderDao() {
        return new OrderDao();
    }
    @Bean
    public CustomerDao customerDao() {
        return new CustomerDao();
    }
}
```



DataSource constructor  
ProductDao Constructor  
OrderDao constructor  
CustomerDao Constructor