

◆ Case Study 2: Java-Based Configuration

■ Case Study Title: E-Commerce Order Processing

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example.ecommerce</groupId>
  <artifactId>E-Commerce_Order_Processing</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>E-Commerce_Order_Processing</name>
  <description>E-Commerce_Order_Processing (Java-Based Configuration)</description>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.33</version>
    </dependency>
  </dependencies>
</project>
```

Product.java

```
package com.example.ecommerce;
import java.util.ArrayList;
import java.util.List;
public class Product {
    private List<String> productList = new ArrayList<>();
    public void addProduct(String product) {
        productList.add(product);
        System.out.println("Product - " + product + " added successfully.");
    }
    public void listProducts() {
        System.out.println("Products List:");
        productList.forEach(System.out::println);
    }
}
```

Order.java

```
package com.example.ecommerce;
public class Order {
    public void createOrder(String product) {
```

```

        System.out.println("Order for product - " + product + " created successfully.");
    }
    public void cancelOrder(String product) {
        System.out.println("Order cancelled, Product: " + product);
    }
}
Payment.java
package com.example.ecommerce;
public class Payment {
    public void processPayment(String product) {
        System.out.println("Payment processed for the product: " + product + "
successfully");
    }
    public void refundPayment(String product) {
        System.out.println("Payment refunded for the product - " + product);
    }
}

```

```

EcommerceService.java
package com.example.ecommerce;
public class EcommerceService {
    private final Product product;
    private final Order order;
    private final Payment payment;

    public EcommerceService(Product product, Order order, Payment payment) {
        this.product = product;
        this.order = order;
        this.payment = payment;
    }

    public void placeOrder(String productName) {
        product.addProduct(productName);
        order.createOrder(productName);
        payment.processPayment(productName);
    }

    public void cancelOrder(String productName) {
        order.cancelOrder(productName);
        payment.refundPayment(productName);
    }

    public void allProducts() {
        product.listProducts();
    }
}

```

AppConfig.java

```

package com.example.ecommerce;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
@Configuration
public class AppConfig {
    @Bean
    public Product product() {
        return new Product();
    }
    @Bean
    public Order order() {
        return new Order();
    }
    @Bean
    public Payment payment() {
        return new Payment();
    }
    @Bean
    public EcommerceService ecommerceService() {
        return new EcommerceService(product(), order(), payment());
    }
}

```

MainApp.java

```

package com.example.ecommerce;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
public class MainApp {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(AppConfig.class);
        EcommerceService service = context.getBean(EcommerceService.class);
        service.placeOrder("Mobile");
        service.allProducts();
        service.cancelOrder("Laptop");
        context.close();
    }
}

```