Practical 03

1. Write a program to demonstrate dependency injection via Constructor for City class.

City.java
Code:

```
package mypack;

public class City {
    private String name;
    private String state;

    public City() {}

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getState() {
        return state;
    }
    public void setState(String state) {
        this.state = state;
    }
    @Override
    public String toString() {
        return "City{name='" + name + "', state='" + state + "'}";
    }
}
```

MainApp.java
Code:

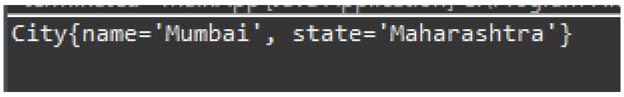
```
package mypack;

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

public class MainApp {
    public static void main(String[] args) {
        try (ConfigurableApplicationContext context =
                new ClassPathXmlApplicationContext("AppConfig.xml")) {

            City city = (City) context.getBean("city");
```

```
                    System.out.println(city);
                }
            }
}
```

AppConfig.xml
Code:

```
<?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="city" class="mypack.City">
          <property name="name" value="Mumbai"/>
          <property name="state" value="Maharashtra"/>
       </bean>

</beans>
```

Output:



```
City{name='Mumbai', state='Maharashtra'}
```

2. **Write a program to demonstrate List dependency injection via Constructor for a Project Class.**

Project.java
Code:

```
package q2;
import java.util.List;

public class Project {
   private String projectName;
   private List<String> teamMembers;

   public Project(String projectName, List<String> teamMembers) {
      this.projectName = projectName;
      this.teamMembers = teamMembers;
   }

   public void displayProjectDetails() {
      System.out.println("Project Name: " + projectName);
```

```
        System.out.println("Team Members:");
        for (String member : teamMembers) {
            System.out.println("- " + member);
        }
    }
}
```

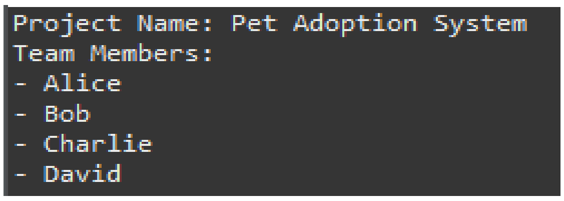
MainApp.java
Code:

```
package q2;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {
    public static void main(String[] args) {
             try (ConfigurableApplicationContext context =
                 new ClassPathXmlApplicationContext("question2.xml")) {

                        Project project = (Project) context.getBean("project");

                        project.displayProjectDetails();
        }
    } }
```

Question2.xml
Code:

```
<?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-3.0.xsd">
    <bean id="project" class="q2.Project">
        <constructor-arg value="Pet Adoption System"/> <!-- Project Name -->
        <constructor-arg>
            <list>
                <value>Alice</value>
                <value>Bob</value>
                <value>Charlie</value>
                <value>David</value>
            </list>
        </constructor-arg>
    </bean>
</beans>
```

Output:



```
Project Name: Pet Adoption System
Team Members:
- Alice
- Bob
- Charlie
- David
```

3. **Demonstrate injection of a Map with bean references as values. The map consists of the key, supplier ID and the value is a list of products.**

Product.java
Code:

```
package q4;

public class Product {
    private String name;

    public Product(String name) {
        this.name = name;
    }
    @Override
    public String toString() {
        return name;
    }
}
```

Supplier.java
code:

```
package q4;

import java.util.List;
import java.util.Map;

public class Supplier {
    private Map<String, List<Product>> supplierProducts;

    public Supplier(Map<String, List<Product>> supplierProducts) {
        this.supplierProducts = supplierProducts;
    }
    public void displaySupplierProducts() {
        for (Map.Entry<String, List<Product>> entry : supplierProducts.entrySet()) {
            System.out.println("Supplier ID: " + entry.getKey());
            System.out.println("Products:");
            for (Product product : entry.getValue()) {
                System.out.println("- " + product);
            }
            System.out.println();
        }
    } }
```

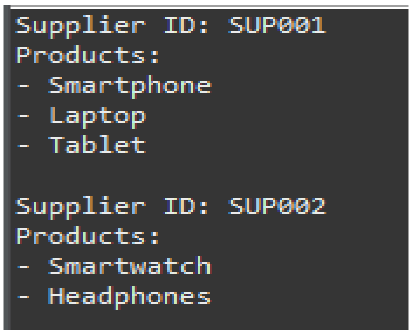
MainApp.java
Code:

```
package q4;

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

public class MainApp {
    public static void main(String[] args) {
        // Load the Spring configuration file
            try (ConfigurableApplicationContext context =
                new ClassPathXmlApplicationContext("question4.xml")) {

                  Supplier supplier = (Supplier) context.getBean("supplier");
                  supplier.displaySupplierProducts();
         }
    }
}
```

Question4.xml
Code:

```
<?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-3.0.xsd">

    <bean id="product1" class="q4.Product">
        <constructor-arg value="Smartphone"/>
    </bean>
    <bean id="product2" class="q4.Product">
        <constructor-arg value="Laptop"/>
    </bean>
    <bean id="product3" class="q4.Product">
        <constructor-arg value="Tablet"/>
    </bean>
    <bean id="product4" class="q4.Product">
        <constructor-arg value="Smartwatch"/>
    </bean>
    <bean id="product5" class="q4.Product">
        <constructor-arg value="Headphones"/>
    </bean>

    <bean id="supplier" class="q4.Supplier">
        <constructor-arg>
            <map>
                <entry key="SUP001">
                    <list>
                        <ref bean="product1"/>
                        <ref bean="product2"/>
                        <ref bean="product3"/>
                    </list>
                </entry>
                <entry key="SUP002">
                    <list>
                        <ref bean="product4"/>
                        <ref bean="product5"/>
                    </list>
                </entry>
            </map>
        </constructor-arg>
    </bean>
</beans>
```

Output:



```
Supplier ID: SUP001
Products:
- Smartphone
- Laptop
- Tablet

Supplier ID: SUP002
Products:
- Smartwatch
- Headphones
```

4. **Demonstrate injection of a List using constructor injection for Product class.**

Product.java
Code:

```
package q3;
import java.util.List;
public class Product {
    private String category;
    private List<String> productList;

    public Product(String category, List<String> productList) {
        this.category = category;
        this.productList = productList;
    }
    public void displayProductDetails() {
        System.out.println("Product Category: " + category);
        System.out.println("Products:");
        for (String product : productList) {
            System.out.println("- " + product);
        }
    }
}
```

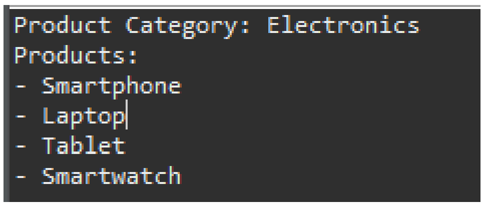
MainApp.java
Code:

```
package q3;

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

public class MainApp {
    public static void main(String[] args) {
            try (ConfigurableApplicationContext context =
                new ClassPathXmlApplicationContext("question3.xml")) {

                        Product product = (Product) context.getBean("product");

                        product.displayProductDetails();
         }
    }
}
```

question3.xml
Code:

```
<?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-3.0.xsd">

    <bean id="product" class="q3.Product">
        <constructor-arg value="Electronics"/>
        <constructor-arg>
            <list>
                <value>Smartphone</value>
                <value>Laptop</value>
                <value>Tablet</value>
                <value>Smartwatch</value>
            </list>
        </constructor-arg>
    </bean>
</beans>
```

Output:



```
Product Category: Electronics
Products:
- Smartphone
- Laptop
- Tablet
- Smartwatch
```

5. **Demonstrate injection of a Map with nested collections. The map consists of the key as the orderID and the value is an Order object.**

Product.java
Code:

```
package q5;

public class Product {
    private String name;

    public Product(String name) {
        this.name = name;
    }
    @Override
    public String toString() {
        return name;
    }
}
```

Order.java
Code:

```
package q5;

import java.util.List;

public class Order {
    private String orderName;
    private List<Product> products;
    public Order(String orderName, List<Product> products) {
        this.orderName = orderName;
        this.products = products;
    }
    @Override
    public String toString() {
        return "Order Name: " + orderName + ", Products: " + products;
    }
}
```

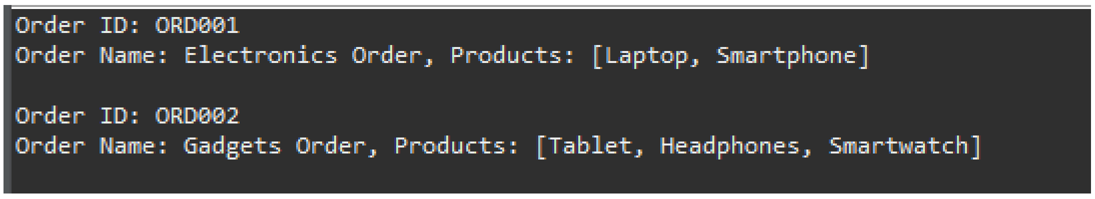
MainApp.java
code:

```
package q5;

import org.springframework.context.ApplicationContext;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import java.util.Map;

public class MainApp {
    public static void main(String[] args) {

            try (ConfigurableApplicationContext context =
                new ClassPathXmlApplicationContext("question5.xml")) {
                    Map<String, Order> orders = (Map<String, Order>)
context.getBean("orderMap");

                for (Map.Entry<String, Order> entry : orders.entrySet()) {
                    System.out.println("Order ID: " + entry.getKey());
                    System.out.println(entry.getValue());
                    System.out.println();
                }
        }
    }
}
```

Question.xml
Code:

```
<?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-3.0.xsd">
    <bean id="product1" class="q5.Product">
        <constructor-arg value="Laptop"/>
    </bean>
    <bean id="product2" class="q5.Product">
        <constructor-arg value="Smartphone"/>
    </bean>
    <bean id="product3" class="q5.Product">
        <constructor-arg value="Tablet"/>
    </bean>
    <bean id="product4" class="q5.Product">
        <constructor-arg value="Headphones"/>
    </bean>
    <bean id="product5" class="q5.Product">
        <constructor-arg value="Smartwatch"/>
    </bean>
    <bean id="order1" class="q5.Order">
        <constructor-arg value="Electronics Order"/>
        <constructor-arg>
            <list>
                <ref bean="product1"/>
                <ref bean="product2"/>
            </list>
        </constructor-arg>
    </bean>
    <bean id="order2" class="q5.Order">
        <constructor-arg value="Gadgets Order"/>
        <constructor-arg>
            <list>
                <ref bean="product3"/>
                <ref bean="product4"/>
                <ref bean="product5"/>
            </list>
        </constructor-arg>
    </bean>
    <bean id="orderMap" class="java.util.HashMap">
        <constructor-arg>
            <map>
                <entry key="ORD001" value-ref="order1"/>
                <entry key="ORD002" value-ref="order2"/>
            </map>
        </constructor-arg>
    </bean>
</beans>
```

Output:



```
Order ID: ORD001
Order Name: Electronics Order, Products: [Laptop, Smartphone]

Order ID: ORD002
Order Name: Gadgets Order, Products: [Tablet, Headphones, Smartwatch]
```