**Display Staff Details**

Description

**Objective:**

To work with a Spring Core application using annotations and bean configuration concepts.

**Concept Explanation:**

1. **Annotations** provide a concise way to configure Spring beans and dependencies, reducing XML configuration overhead.
2. **Bean configuration** involves defining and configuring application components (beans) and their dependencies through XML-based configuration or annotation.
3. **Constructor injection:** Dependencies are provided to a class through its constructor.

**Concept Implementation:**

1. **Annotations** mark the **Staff** and **Department** classes as Spring beans, allowing them to be managed by the Spring IoC container.
2. **Bean configurations** are defined within the **ApplicationConfig** class using annotations like **@Bean**.
3. **Constructor injection** is implemented by creating parameterized constructors in the **Staff** and **Department** classes, allowing dependencies to be injected during object instantiation in the Spring configuration class (**ApplicationConfig**).

**Display Staff Details - Constructor Injection**

**Staff**class with the below **private attributes**is provided as a part of code skeleton

|  |  |
| --- | --- |
| staffId | int |
| staffName | String |
| departmentName | String |
| contactNo | long |

**Getter and setter** methods for all the above attributes are provided as a part of code skeleton. Write a four argument constructor which accepts staffId, staffName, departmentName and contactNo as the parameters. Annotate the**Staff** class to be recognized as a Spring bean.

**Department**class with the below **private attributes**is provided as a part of code skeleton

|  |  |
| --- | --- |
| departmentId | int |
| staffs | List<Staff> |

**Getter and setter** methods for all the above attributes are provided as a part of code skeleton. Write a two argument constructor which accepts departmentId and list of staffs as the parameters. Annotate the**Staff** class to be recognized as a Spring bean.

Staff has to set to the Department via **department and staff** methods in the **ApplicationConfig** class.

A method **public void displayStaffDetails()**will be provided in the **Department**class as a part of code skeleton. This method is used to display the Staff details as shown in the sample output.

**ApplicationConfig** will be used as configuration class.

Define beans for Staff and Department objects within this class using the  annotation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method name** | **Input Parameters** | **Output Parameters** | **Logic** |
| staff | nil | Staff | This method will create and return Staff object |
| department | nil | Department | This method will create and return Department object |

**Driver**class with the below methods are provided as a part of code skeleton

* **public static Department loadStaffDetails()**--> This method should fetch the **Department**object from **ApplicatinConfig class and**return the same.
* **public static void main(String[] args)**-->  Inside the main method invoke the **loadStaffDetails**method and obtain the **Department**object to output the staff details.

**Design Constraints**

* **Staff**class and the **Department**class should be present in**com.spring.app** package.
* Write  appropriate constructors
* The class Name/Attribute Name/PackageName should be same as specified in the problem statement. Do not create any new packages.

**Sample Output:**

Staff Details:

Staff Id:1

Staff Name:Ragul

Contact Number:9445543300

Department Name:CSE

Department Id:123