**1. 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. Staff class should be registered as a bean with the bean id "staffObj" via XML file.**

**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. Department class should be registered as a bean with the bean id "departmentObj" via XML file.**

**Staff has to set to the Department via constructor injection in the XML file.**

**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.**

**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 beans.xml and return the same**
* **public static void main(String[] args)-->  Inside the main method invoke the loadStaffDetails method and obtain the Department object.**

**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.**
* **The XML configuration should be done in the file beans.xml. This XML should be available under the resources folder of the created maven project.**

**Sample Output:**

**Staff Details:**

**Staff Id:1**

**Staff Name:Ragul**

**Contact Number:9445543300**

**Department Name:CSE**

**Department Id:123**

**2. Gold Rate Calculation - Collections**

**A jewellery shop need an application to calculate the gold rate for the specific grams chosen by their customers. Create a spring core application using Maven to perform this task.**

**GoldRateInfo class with the below private attribute is provided as a part of code skeleton**

|  |  |
| --- | --- |
| **rateInfo** | **Map<Integer,Double>** |

**Getter and setter methods for the above attribute is provided as a part of code skeleton. GoldRateInfo class should be registered as a bean with the bean id "rateInfoObj" via XML file.**

**Map that holds goldCarat and the rate per gram. The map needs to be configured in the beans.xml and injected into the GoldRateInfo via setter based injection.**

**A method  public double calculateGoldRate(int goldCarat, double grams) will be provided in the GoldRateInfo class as a part of code skeleton. Depending on the user input fetch the appropriate gold rate from the beans.xml file and calculate the gold rate and return the result back to the user.**

**The map values in the beans.xml file for gold rate per gram for various carat are as follows:**

|  |  |
| --- | --- |
| **Carat (Integer)** | **Rate per gram (Double)** |
| **18** | **3800.0** |
| **22** | **4300.0** |
| **24** | **4500.0** |

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

* **public static GoldRateInfo loadGoldRateDetails()--> This method should fetch the GoldRateInfo object from beans.xml and return the same**
* **public static void main(String[] args)-->  Inside the main method invoke the loadGoldRateDetails method and obtain the GoldRateInfo object.**

**Design Constraints:**

* **GoldRateInfo and Driver class should be present in com.spring.app package**
* **Maintain the same className/Attribute Name/PackageName as specified in the problem statement. Do not create any new packages.**
* **All the configuration should be done in the file named "beans.xml" in the created maven project**

**Sample Input:  
Enter the carat:  
18  
Enter Total Grams:  
13  
Sample Output:  
Total Gold Rate is Rs:49400.0**

**3. Account-Loan InnerBean**

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

|  |  |
| --- | --- |
| **accNumber** | **String** |
| **accHolderName** | **String** |
| **accBalance** | **double** |
| **loanInfo** | **Loan** |

**Getter and setter methods for all the above attributes are provided as a part of the code skeleton.**

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

|  |  |
| --- | --- |
| **loanType** | **String** |
| **loanAmount** | **double** |

**Getter and setter methods for all the above attributes are provided as a part of the code skeleton.**

* **Account should be registered as a bean with the spring container via XML file**
* **Loan is used for only particular property , so Loan should be declared as an inner bean of Account.**
* **Loan should be injected into Account via Constructor based Injection**
* **Input for Account and Loan attributes should be set through beans.xml file**

**Create a class called Driver with the below methods**

**\* loadAccount --> This method should fetch the Account object from beans.xml and return the same**

**\* main method -->  Inside the main method invoke the loadAccountmethod, obtain the Account object and display the Account and Loan details as shown in the sample output**

**Design Constraints:**

* **Account class should be present in com.spring.app package**
* **Loan class should be present in com.spring.app package**
* **Driver class should be present in com.spring.app package**
* **The className/Attribute Name/PackageName should be same as specified in the problem statement. Do not create any new packages.**
* **The XML configuration should be done in the file beans.xml. This XML should be available under the resources folder of the created maven project.**

**Sample Output**

**Account number:335647852**

**Account holder name:Vanitha**

**Balance:250000.0**

**Loan type:HomeLoan**

**Loan amount:150000.0**