## 1. DBConfig-SetterBasedInjection

Create a class called DBconfig  with the below attributes

|  |  |
| --- | --- |
| driverName | String |
| url | String |
| userName | String |
| password | String |

Include getters and setters for all the above mentioned attributes.

DBConfig class should be registered as a bean with the spring container via XML file

The  attributes inside the DBConfig should be configured in the xml with the below values

|  |  |
| --- | --- |
| Attribute Name | Value |
| driverName | com.mysql.jdbc.Driver |
| url | jdbc:mysql:@localhost:3306/dbName |
| userName | root |
| password | root |

The values for these attributes should be injected via setter based injection.

Create class called EmployeeDAO that has below attribute

|  |  |
| --- | --- |
| dbConfig | DBConfig |

Include getter/setter for the above attribute

EmployeeDAO should be registered as a bean with the spring container via XML file and DBConfig should be injected using setter based injection into the EmployeeDAO class

Create a class called Driver with the main method. Retrieve the EmployeeDAO object and display the details as shown in the sample input/output.

## 2. Create a Order class with the below private attributes

|  |  |
| --- | --- |
| orderNumber | long |
| orderDate | LocalDate |
| orderAmount | double |

Include getter and setter methods for all the above attributes.

Create multiple Bean of Order

Create a Customer class with the below private attributes

|  |  |
| --- | --- |
| customerId | long |
| contactDetails | AddressBook |
| List<Order> | Order |

Include getter and setter methods for all the above attributes.

Create an AddressBook class with the below private attributes

|  |  |
| --- | --- |
| phoneNumber | String |
| tempAddress | Address |

Include getter and setter methods for all the above attributes.

Create an Address class with the private attributes

|  |  |
| --- | --- |
| houseName | String |
| Street | String |
| City | String |
| State | String |

Include getter and setter methods for all the above attributes.

AddressBook should be registered as a bean with the spring container via java based configuration

Address is used for only particular property , so Address should be declared as a inner bean

to AddressBook.

Address should be injected into AddressBook via Constructor based Injection

Create a configuration class called CustomerConfig

Create a class called Driver with the below methods

               \* loadAddressBook --> This method should fetch the AddressBook object from applicationContext.xml and return the same

               \* main method -->  Inside the main method invoke the loadAddressBook and obtain the AddressBook object.

                                                    Get the Temporary Address details and the phone Number from the user, set the values and display the  temporary address and phone number in the main method.