1.Create a class called Customer that has the below attributes

|  |  |
| --- | --- |
| custId | int |
| custName | String |
| emailId | String |
| contactNo | long |

Create another class  MemberShip that has the below attributes

|  |  |
| --- | --- |
| membershipId | int |
| membershipType | String |
| visitsPerYear | int |
| customer | Customer |

Attributes should be private.

Customer has to set to the membership via constructor injection in the XML file.

Create main class with the main method to test.

2. Create a class called Passport that has the below attributes:

|  |  |
| --- | --- |
| passNum | int |
| dateOfIssue | Date |
| dateOfExpiry | Date |

Create another class called Employee that has the below attributes

|  |  |
| --- | --- |
| empId | int |
| empName | String |
| passObj | Passport |

The Passport object should be autowired above the property via annotations. Create a class called ApplicationConfig that has the required annotations for scanning and registering the bean definitions. Test with main class.

3. Create a class called CurrencyConverter that has the following attribute:

|  |  |
| --- | --- |
| mObj | Map<String,String> |

Map that holds currencyName and the conversion rate. The map needs to be configured in the xml and injected into the CurrencyConverter via setter based injection. Write a method called getTotalCurrencyValue that accepts a String. The string contains the total currency and the currencyName.  For ex: “7Dollar”. The method needs to parse the string and depending on the user input fetch the appropriate currency value from the xml file calculate the total rate and return the result back to the user.

Suppose if the user has entered a value “5Dollar”, then the output should be 325.

The map values in the xml file for various currencies are as follows:

|  |  |
| --- | --- |
| DOLLAR | 65 |
| EURO | 80 |
| DINAR | 218 |

4. 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.

Declare Address as a inner bean to AddressBook.

Address should be injected into AddressBook via Constructor based Injection. Create a main class to test.

5. **Create Vehicle class with the attribute**

|  |  |
| --- | --- |
| source | String |
| destinationMap | Map<String, Double> |

The default value for source must be “Chennai”

The Map should be configured using @Value annotation**.**

|  |  |
| --- | --- |
| **Key -destination (String)** | **Value-**cost**(Double)** |
| Bangalore | 1500 |
| Delhi | 2500 |
| Mumbai | 2000 |
| Hyderabad | 1000 |
| Pune | 2500 |

Create main class. Get destination as input from the user. For the given destination from source, display the cost.