Assignment - SpringCore

1. Problem Statement:

a). Develop an application using Spring in which you create an employee bean with fields as emp_name,emp_id,salary,dob,doj,address and then create an address bean with fields as house_no,street and city.

Now try to achieve loose coupling using IOC/DI using constructor injection,setter injection and autowiring ?

b) Reuse the above application and try to achieve constructor injection using collection(i.e list,set and map)?

Problem Statement 2

2) Develop a **RedBus Reservation Booking** application with following functional requirements

i) **Passenger** bean with attributes like name, pan_card_number, contact_number, passport_number.

ii) **Bus_Details** bean which has attributes like bus_id, date_of_arrival, arrival_Time and departure_time,From, To, and rate_of_ticket.

ii) **Ticket** bean which has details like Bus_details,No_of_seats, Collection of passengers and total_amount. Ticket class should also have functionality which will select appropriate flight, arrival_time,date and calculate total amount of ticket. Display Ticket information.

Ticket Details :

FROM :

To :

Departure Date :

Departure Time:

****

****

List Of Passengers:

****
****

Total Amount :

——---

3. Problem Statement:

Develop an application using spring core with the below given functionalities

i) Invoice class that a hardware store might use to represent an invoice for
an item sold at the store.

ii) An Invoice should include four pieces of information as instance variables - a
part_number(type String),a part_description(type String),a quantity of the item being
purchased (type int) and a price per item (double).

 iii) The class should have a constructor that initializes the four instance variables. Provide a
set and a get method for each instance variable. In addition, provide a method named
getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price
per item), then returns the amount as a double value. If the quantity is not positive, it should be
set to 0.If the price per item is not positive, it should be set to 0.0.

iv) Write a test application named InvoiceTest that demonstrates class Invoice's capabilities.