

Spring Core Hands On - Patient Management

Important Instructions:

- Please read the document thoroughly before you code.
- Import the given skeleton code into your Eclipse.
- Do not change the Skeleton code or the package structure, method names, variable names, return types, exception clauses, access specifiers etc.
- You can create any number of private methods inside the given class.
- You can test your code from main() method of the program
- Using Spring Core develop the application using Java based configuration.

Assessment Coverage:

**Classes, Objects and Members, Setter Injection
Property Configuration
JUnit**

Purpose:

Patient Information:

A famous city hospital has to manage the patient records. This helps the doctor to manage the patient records.

S F D :

Import the below attached skeleton code into your eclipse project and implement the required functionalities



PatientManagement-Skeleton.zip

Provided **Patient** class with the following private attributes:

Member Field Name	Type
pid	String
pname	String
disease	String
sex	String
admit_status	String
age	Integer

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

The values for these attributes should **be injected via properties file**.

Provided **PatientService** class with the following private attributes:

Spring Core Hands On - Patient Management

Member Field Name	Type
patient	Patient

Service class should be registered as a **bean** with the spring container via **Java Configuration class**.

Main.java: main(String[] arg):void

- Using ApplicationContext, get PatientService bean.
- Invoke the getPatientDetails() method using PatientServiceBean.

Technical Specifications:

Component Name	Method Name	Input	Output	Exception
PatientService	getPatientDetails		void	
AppConfig.class	Contains all the Java configurations related to Service			

Business Rules & Validations:

In **PatientService class** includes the following private attribute **and inject via setter based injection**.

```
private Patient patient;
```

Getter and setter methods for the above attribute has been provided as a part of code skeleton

Service 1: getPaientDetails(): void

- Use Patient Bean, get the Patient information.

Limitations and Constraints:

- Patient class** should be in **com.cts.patient.model** package.
- PatientService** class should be in **com.cts.patient.service** package.
- Main class** should be in **com.cts.patient** package.
- All of the above mentioned java classes to be configured as beans in **AppConfig.java** file
- patient** should be declared as an **inner bean** in the **PatientService** class and should be injected by autowire inside **AppConfig.java**.
- Use ONLY **AppConfig.java** for all configurations.

Spring Core Hands On - Patient Management

Sample output

Patient Id : P1000
Patient Name : ABC
Disease : fever
Sex : female
Admit Status : yes
Age : 55