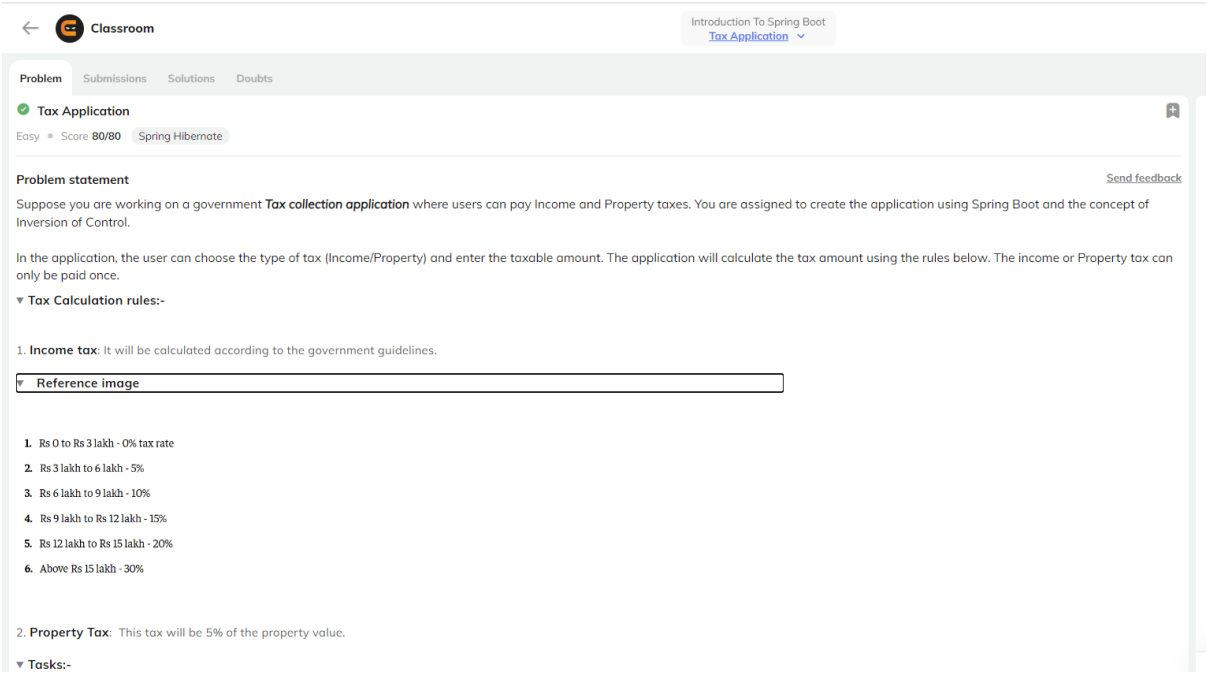
Tax Application I

Problem Statement:



Classroom

Introduction To Spring Boot
Tax Application

Problem | Submissions | Solutions | Doubts

Tax Application

Easy • Score 80/80 Spring Hibernate

Problem statement Send feedback

Suppose you are working on a government ***Tax collection application*** where users can pay Income and Property taxes. You are assigned to create the application using Spring Boot and the concept of Inversion of Control.

In the application, the user can choose the type of tax (Income/Property) and enter the taxable amount. The application will calculate the tax amount using the rules below. The income or Property tax can only be paid once.

▼ **Tax Calculation rules:-**

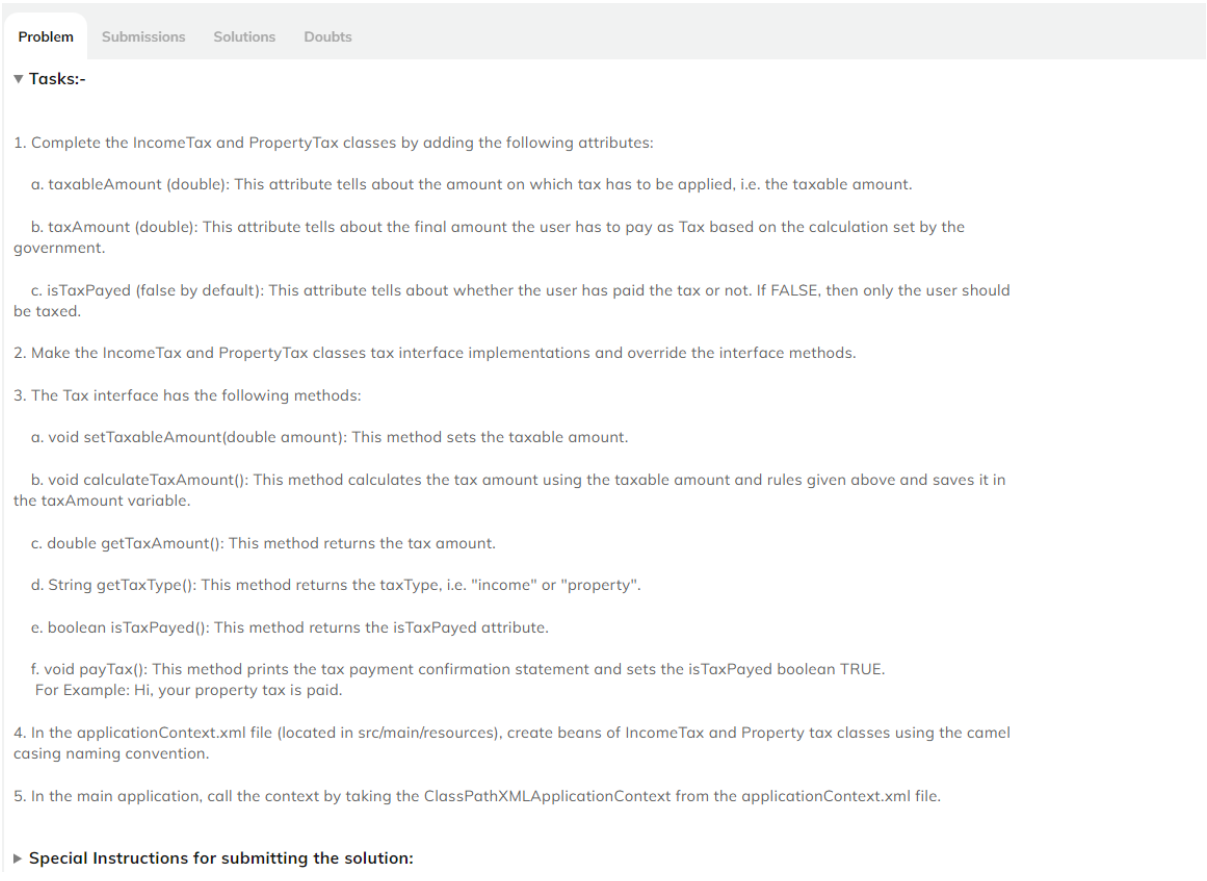
1. **Income tax**: It will be calculated according to the government guidelines.

▼ Reference image

1. Rs 0 to Rs 3 lakh - 0% tax rate
2. Rs 3 lakh to 6 lakh - 5%
3. Rs 6 lakh to 9 lakh - 10%
4. Rs 9 lakh to Rs 12 lakh - 15%
5. Rs 12 lakh to Rs 15 lakh - 20%
6. Above Rs 15 lakh - 30%

2. **Property Tax**: This tax will be 5% of the property value.

▼ **Tasks:-**



Problem | Submissions | Solutions | Doubts

▼ **Tasks:-**

1. Complete the IncomeTax and PropertyTax classes by adding the following attributes:

a. taxableAmount (double): This attribute tells about the amount on which tax has to be applied, i.e. the taxable amount.

b. taxAmount (double): This attribute tells about the final amount the user has to pay as Tax based on the calculation set by the government.

c. isTaxPayed (false by default): This attribute tells about whether the user has paid the tax or not. If FALSE, then only the user should be taxed.

2. Make the IncomeTax and PropertyTax classes tax interface implementations and override the interface methods.

3. The Tax interface has the following methods:

a. void setTaxableAmount(double amount): This method sets the taxable amount.

b. void calculateTaxAmount(): This method calculates the tax amount using the taxable amount and rules given above and saves it in the taxAmount variable.

c. double getTaxAmount(): This method returns the tax amount.

d. String getTaxType(): This method returns the taxType, i.e. "income" or "property".

e. boolean isTaxPayed(): This method returns the isTaxPayed attribute.

f. void payTax(): This method prints the tax payment confirmation statement and sets the isTaxPayed boolean TRUE.
For Example: Hi, your property tax is paid.

4. In the applicationContext.xml file (located in src/main/resources), create beans of IncomeTax and Property tax classes using the camel casing naming convention.

5. In the main application, call the context by taking the ClassPathXMLApplicationContext from the applicationContext.xml file.

▸ **Special Instructions for submitting the solution:**