System Requirements Specification Index

For

Python Income Tax SystemConsole Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

1. Project Abstract:

Provide a code solution to calculatethe Income tax of the Tax Payer considering the income slab and standard exemptions.

2.Common Constraints with description :

1. Take input details of n Tax Payersand store in a collection.
    a. Make sure the PAN(Permanent Account Number) is unique, else throw a custom exception(PANAlreadyExistsException)

2. Create "TaxPayer" class with
    a. Tax Payer name asString
    b. PAN as String
    c. Age as int
    d. Email as String
    e. grossSalary as float(per annum)
    f. provident Fund (PF) (It should be 3% on gross salary per annum)
    g. professionalTax as float (It should be 200 per month 2400 per annum)

3. Show Total deductions based on PAN card
    a. Take PAN number from keyboard
        i. PANmust exist in collection, else throw a custom exception (PANDoesNotExistsException)

4. Create "Deductions" class with
    a. PAN
    b. deduction_Sec_80C
    c. houseRent (Max 50000, if entered more than 50000, take it 50000 only)
    d. totalDeduct (totalDeduct=pf+pt+deduction_sec_80C+house_rent)

5. Create a class "TaxDetails" with (Put in the collection)
    a. PAN
    b. taxableSalary
    c. totalTax

6. Class IncomeTaxCalculator
    a. taxPayerList,panList,deductionList,taxDetailsList asCollection data members.
    b. addTaxPayer(TaxPayertaxPayer) as method.
    c. totalDeductions(String PAN)as method.
    d. calTax(String PAN) as method.
    e. showTaxableSalary(String PAN) as method.
    f. showTotalTax(String PAN)as method.

3. Template Code Structure

Resources (Instances)

Class	Description	Status
Deductions	This class contains all the properties of the Deductionsin the constructor.	Already implemented.
TaxDetails	This class contains all the properties of the TaxDetailsin the constructor.	Already implemented.
TaxPayer	This class contains all the properties of the TaxPayerin the constructor.	Already implemented.

Resources (Business Logic)

Class	Description	Status
IncomeTaxCalculator	This class contains all the methods which are used to write the business logic for the application.	Partially implemented (available only template code).

Resources (Exceptions)

Class	Description	Status
PANAlreadyExistsError	Custom Exception to be thrown if pan is already available in the collection while taking the input of the tax payer from the console.	Already created.
PANDoesNotExistsError	Custom Exception to be thrown if pan does not exist when we want to show total deductions based on PAN card.	Already created.

Resources (Main class)

Class	Description	Status
MainClass	This is the class which is having methods to be used to read data from the user or display the data. All the business logic methods of the **IncomeTaxCalculator** class will be called from this class.	Partially implemented (available only template code).

Note:
Consider every amount (salary, house rent, pf etc) is annual based.

Calculate tax based on new regime tax slabs as given in the below table.

Income Tax Slab	**Tax rates as per new Regime**
₹0 - ₹2,50,000	Nili.e tax =0
₹2,50,001 - ₹ 5,00,000	5%
₹5,00,001 - ₹ 7,50,000	₹12500 + 10% of total income exceeding ₹5,00,000
₹7,50,001 - ₹ 10,00,000	₹37500 + 15% of total income exceeding ₹7,50,000
₹10,00,001 - ₹12,50,000	₹75000 + 20% of total income exceeding ₹10,00,000
₹12,50,001 - ₹15,00,000	₹125000 + 25% of total income exceeding ₹12,50,000
Above ₹ 15,00,000	₹187500 + 30% of total income exceeding ₹15,00,000

4. Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to

   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.

3. The editor Auto Saves the code.
4. If you want to exit(logout) and to continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To run application for use case1 use the following command

   ```
   python3 main.py
   ```

7. Mandatory: Before final submission run the following command

   ```
   python3 -m unittest
   ```

8. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.
9. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

--------*---------