System Requirements Specification Index

For

Python Bonus Calculator -Console Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Problem Statement : Provide a code solution to calculate bonus to employees of a company.

Description : A Company wants to give away bonus to its employees. You have been assigned as a programmer to automate this process. You have to key in empid and data of birth, and based on the age of the employee bonus must be calculated.

Create a class BonusCalculator with one static method

1. calculate_bonus(dictionary1, dictionary2), which accepts dictionary1, dictionary2 objects as input and returns a dictionary object.

Note:

1. Build two dictionaries. The first dictionary contains employee id as key and DOB as value, and the second dictionary contains same employee id as key and salary as value.
2. If the age of the employee in the range of 25 to 30 years (inclusive), the employee should get bonus of 20% of his salary and in the range of 31 to 60 years (inclusive) should get 30% of his salary. Store the result in dictionary in which Employee ID as key and revised salary as value. Assume the age is calculated based on the date 01-09-2014.
3. Other Rules:

   a. If Salary is less than 5000 add100 to the salary.

   b. If the age is less than 25 or greater than 60 add 200 to the salary.

   c. a takes more priority than b i.e both if a and b are true then add 100 to the salary.

and another MainClass with one static method

In this static method,

1. Read number of employees from keyboard.

2. Read the Employee Details from input device where details would include id, DOB (date of birth) and salary in the given order. The datatype for id is integer, DOB is string and salary is integer.

3. Call the calculate_bonus() method to calculate bonus to employees of a company.

4. Display final results of all employees.

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 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 the application use the following command

   python3 mainclass.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.

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