System Requirements Specification Index

For

Python Staff Selection Console Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Problem Statement Description

The staff selection committee decides every applicant must appear for a selection exam in three subjects as per applicants' choice of subjects. They must qualify minimum 70% of the total. Individual subject pass mark is 50 out of 100.

You need to show all qualified applicants name, subject-wise marks, total and percentage in the descending order of the percentage of applications.

You need to accept the individual applicant's 4 details in comma (,) separated format as a String. They are the name of the applicant, subject1 mark, subject2 mark & subject3 mark.

Eg:

Venu,92, 99, 95
Radhika,92,33,71

1) Create a class Applicant with a constructor and the constructor should accept candidate details as a string parameter of following members.

name,subject1,subject2, subject3, total,percentage

Create getter and setter methods for all members and then override __repr__() method and return all candidate details.

2) Create a class MainClass with the following 2 methods.

a. Define a static method calculate_result(str), it should accept applicant details in string format.
   raise ValueError if input values (name, 3 subject marks) are not in valid format.
   raise IndexError if number of subjects are not 3.
   raise a custom exception OutOfBoundaryMarksErrorif any subject mark is not in the boundary i.e marks>=0 and marks<=100.
   Calculate total i.e total= subject1+subject2+subject3.
   Calculate percentage if all subject's marks greater than or equal to 50.
   Here, percentage =total/300)*100
   Finally return Applicant object if applicant gets minimum 50 marks in 3 subjects and percentage is greater or equal to 70%.

b. Define a static method sort_result(collection), It should accept a collection of Applicant objects.Sort each applicant object by name and then return the collection.

3) Define a main method

a. It should accept first number of applicant's details to enter. Then accept each applicant name, marks for subject1, subject2 and subject3 in a comma separate (,) string format.
b. Invoke calculate_result() method.
c. Finally invoke sort_result() method and display qualified applicants details in the descending order by percentage.

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 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.

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