System Requirements Specification Index

For

Python Skills Evaluation

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

USE CASE DESCRIPTION

Use Case 1

Write a Python program to perform all arithmetic operations on 2 input integers using Dunder methods (Magic methods).

Example:

Input: 4 7

Output:

11
-2
28
4
0.571
0
16384

Description

1. Define a class **Point**and define a constructor in the class.
2. Define appropriate Magic methods to perform arithmetic operations.
3. Take 2 numbersas console input in **main ()** method, from the user.
4. Invoke the respective Magic methods with respective operator.
5. Finally return the result from magic methods to main() and display it.

Use Case 2

Write a Python program to demonstrate multilevel inheritance.

Description

1. Defile **Student** as a base class in this define 2 instance methods set_details(self,id,name) to set the data and get_details(self) to get the data.
2. Defile **Marks** class as a sub class to **Student** class in this define 2 instance methods set_marks(self, subject1,subject2,subject3,subject4,subject5) to set the marks data and get_marks (self) to get the data.
3. Defile **Result** as a sub class to Marks class and define one instance method get_result(self), in this calculate total marks and result whether student is passed or failed.
4. Minimum pass marks 35 in each subject.
5. From main method, take console input as id and name, pass to set_details(self,id,name).
6. From main method, take console input of 5 subjects marks and pass to set_marks(self, subject1,subject2,subject3,subject4,subject5).
7. Finally return the details from get_details(self), return the marks from get_marks (self) and return total, result from get_result(self)to main() and display it.

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 magic_methods.py

7. To run application for use case2 use the following command

   python3 multi_inh.py

8. Mandatory: Before final submission run the following command

   python3 -m unittest

9. 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.
10. 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.

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