
System Requirements Specification Index

For

Python Cricket Kit ContributionConsole Application

Version 1.0

Problem Statement : Provide a code solution to find out wholesale purchases for Cricket Kit Contribution using Python coding with Custom Exception.

Description : Create Purchase class with the variables purchase_id, purchase_date, user, total_amount, with a parameterized constructor and implement one static method as below:

1. @staticmethod
obtain_purchase_with_amount(str) throws InvalidWholeSaleError, which accepts String Object as input and returns a Purchase object if the number of items in the input is greater than or equal to 5, else throws InvalidWholeSaleError.
2. Here InvalidWholeSaleError is custom exception, you have to create it.

Here,

total_amount = cost of item1 * quantity1 + cost of item2 * quantity2 + cost of item3 * quantity3 ...

Note:

1. The purchase details (input details) should be in the given format.
purchase_id, purchase_date, user, item_name1, item_cost1, item_quantity1,
item_name2, item_cost2, item_quantity2,...

Ex:

1011,12-12-2021,Venu, Bat,2500.00,3,Helmet,5000.00,2,...

2. Display Purchase Id User Total Amount of purchase details in a row format.

Ex:

```
print("Purchase ID: {} USER: {} Total AMOUNT: {}".format(purchase_id,user,total_amount))
```

3. One custom exception InvalidWholeSaleError class with one parameterized constructor which throw the error message "Purchase [purchase id] is not a wholesale", where [purchase id] indicated the id of the purchase. This exception is raised when the number of item in a purchase is less than 5.

4. One separate MainClass with read_data(self) instance method.
read_data() method is for reading the Purchase Details from input devices and call the obtain_purchase_with_amount (str) method to find out whole sale purchases for Cricket Kit Contribution.

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.

-----*-----