EX - 7 INHERITANCE

T SADAKOPA RAMAKRISHNAN | 3122225002109 | IT - B

Q1) Write a program to model a real-time online shopping system using inheritance. The base class should be called Product, and it should have attributes for the name, price, and quantity of the product. The derived classes should be ElectronicProduct and ClothingProduct, which inherit from Product. Each derived class should have additional attributes specific to that type of product, such as the brand and model for ElectronicProduct, and the size and color for ClothingProduct. Implement methods in each class to display the product information. Additionally, override the display_information() method in the derived classes to include the specific attributes of each product type. Also, implement a function in the derived classes to calculate the total price based on the quantity of the product. Finally, overload the '+' operator in the derived classes to allow adding two products together offering a combo pack with the summed-up price tag

Aim:

To execute a python program to create a parent class and two child classes using inheritance concept.

Code:

```
# -*- coding: utf-8 -*-
"""
This module contains a program applying the concept
of Inheritance of Classes. This is a part
of the exercises given under the course UIT2201 (Programming
and Data Structures).

In this source code I've executed my own logic and may contain
bugs.
The source code has followed good coding practices.

Your comments and suggestions are welcome.

Created on Wed May 17 2023

Revised on Wed May 28 2023
```

```
Original Author: T. Sadakopa Ramakrishnan
<sadakopa2210221@ssn.edu.in>
"""


class Product:
    '''
    This is the parent class.
    '''
    def __init__(self,name, quantity):
        '''
        Constructor to initialize variables
        '''
        self._name = name
        self._price= data[name][0]
        self._quantity = quantity

    def display_information(self):
        '''
        Method to display information
        '''
        print(f"Name: {self._name} \nPrice: {self._price}
\nQuantity: {self._quantity}", end = '')

class ElectronicProduct(Product):
    '''
    Child Class of the Parent class Product
    '''
    def __init__(self,name,quantity,brand,model):
        '''
        Constructor to initialize the variables
        '''
        super().__init__(name,quantity)
        self._brand = data[name][1]
        self._model = data[name][2]

    def display_information(self):
        '''
        Overriding the display_information() method from Parent
class
        '''
```

```
        super().display_information()
        print(f"\t\tBrand: {self._brand}\t\tModel:
{self._model}")

    def __add__(self, other):
        '''
        Method to create Combo Products
        '''
        name = self._name + " and " + other._name
        price = self._price + other._price
        quantity = self._quantity + other._quantity
        total_price = price * quantity
        brand = self._brand + other._brand
        model = self._model + other._model
        data[name] = [total_price, brand, model]
        return ElectronicProduct(name, quantity,brand,model)

class ClothingProduct(Product):
    '''
    Child class of the Parent class
    '''
    def __init__(self,name,quantity, size, color):
        '''
        Constructor to initialize variables
        '''
        super().__init__(name, quantity)
        self._size = data[name][1]
        self._color = data[name][1]

    def display_information(self):
        '''
        Method to display information
        '''
        super().display_information()
        print(f"\t\tSize: {self._size}\t\tColor: {self._color}")

    def __add__(self, other):
        '''
        Method to create combo products
        '''
        name = self._name + " and " + other._name
```

```
        price = self._price + other._price
        quantity = self._quantity + other._quantity
        total_price = price * quantity
        size = self._size + other._size
        color = self._color + other._color
        data[name] = [total_price, size, color]
        return ClothingProduct(name, quantity,size,color)

#DataBase
data = {"Earpods":[5000, 'JBL', '250NC'],
        "iPad":[60000,'Apple', 'Pro'],
        "Shirt":[350, 'M', 'Black'],
        "Pant":[2000, 'S', 'Dark Blue']}

#Test cases for the above code
if __name__ == "__main__":
    #This part of the program will not be executed when the file
is imported.

    #Creating Electronic product object 1
    e1 = ElectronicProduct('Earpods', 5,'JBL','250NC')
    e1.display_information()
    print()

    #Creating Electronic product object 2
    e2 = ElectronicProduct('iPad', 10, 'Apple', 'Pro')
    e2.display_information()
    print()

    #Combo product object
    e3 = e1 + e2
    e3.display_information()
    print()

    #Creating clothing product object 1
    c1 = ClothingProduct("Shirt", 100, 'XL', "Red")
    c1.display_information()
    print()

    #Creating clothing product object 2
    c2 = ClothingProduct("Pant", 50, "L", "Black")
```

```
c2.display_information()
print()

#Combo Product object
c3 = c1 + c2
c3.display_information()
print()
```

Output:

```
Name: Earpods
Price: 5000
Quantity: 5     Brand: JBL      Model: 250NC

Name: iPad
Price: 60000
Quantity: 10        Brand: Apple        Model: Pro

Name: Earpods and iPad
Price: 975000
Quantity: 15        Brand: JBL and Apple        Model: 250NC and Pro

Name: Shirt
Price: 350
Quantity: 100       Size: M     Color: M

Name: Pant
Price: 2000
Quantity: 50        Size: S     Color: S

Name: Shirt and Pant
Price: 352500
Quantity: 150       Size: M and S       Color: M and S
```