ASSIGNMENT – PYTEST – 08/06/2024

- Vinesh Kumar M V

Shopping Cart Class :

```
list_item = [
  {"name": "Tea", "price": 10},
  {"name": "Coffee", "price": 15},
  {"name": "Cake", "price": 30},
  {"name": "Latte", "price": 20},
  {"name": "Doughnut", "price": 15}
]

class ShoppingCart:
  def __init__(self, list_item):
    self.list_item = {item['name']: item['price'] for item in list_item}
    self.cart = {}

  def show_cart(self):
    return self.cart

  def add_to_cart(self, name, quantity):
    if name not in self.list_item:
      print(f"Item '{name}' not found.")
      return
    if quantity <= 0:
      print("Quantity must be positive.")
      return
    if self.list_item[name] <= 0:
      print("Price must be positive.")
      return
    if name in self.cart:
        self.cart[name] += quantity
    else:
        self.cart[name] = quantity
    print(f"Added {quantity} of {name} to cart.")

def update_cart(self, name, quantity):
    if name not in self.cart:
        print(f"Item '{name}' is not in the cart.")
        return
    if quantity < 0:
        print("Quantity cannot be negative.")
        return
    if quantity == 0:
        del self.cart[name]
        print(f"Removed {name} from cart.")
    else:
        self.cart[name] = quantity
        print(f"Updated {name} quantity to {quantity}.")

def remove_from_cart(self, name):
    if name in self.cart:
        del self.cart[name]
        print(f"Removed {name} from cart.")
    else:
        print(f"Item '{name}' is not in the cart.")

def generate_bill(self):
    bill = []
    gtotal = 0
    bill.append(('Item', 'Quantity', 'Price', 'Total'))
```

```
        bill.append(('----------', '----------', '----------', '----------'))

        for item_name, qty in self.cart.items():
            price = self.list_item[item_name]
            total_price = qty * price
            gtotal += total_price
            bill.append((item_name, qty, price, total_price))

        bill.append(('----------', '----------', '----------', '----------'))
        bill.append(('', '', 'Grand Total', gtotal))

        return bill
```

PYTEST CODE

```
import pytest
from shopping_cart import ShoppingCart, list_item

@pytest.fixture
def cart():
    return ShoppingCart(list_item)

def test_add_single_item(cart):
    cart.add_to_cart("Tea", 1)
    assert cart.show_cart() == {"Tea": 1}

def test_add_multiple_items(cart):
    cart.add_to_cart("Tea", 1)
    cart.add_to_cart("Coffee", 2)
    assert cart.show_cart() == {"Tea": 1, "Coffee": 2}

def test_remove_existing_item(cart):
    cart.add_to_cart("Tea", 1)
    cart.remove_from_cart("Tea")
    assert cart.show_cart() == {}

def test_remove_non_existing_item(cart):
    cart.add_to_cart("Tea", 1)
    cart.remove_from_cart("Coffee")
    assert cart.show_cart() == {"Tea": 1}

def test_remove_from_empty_cart(cart):
    cart.remove_from_cart("Tea")
    assert cart.show_cart() == {}

def test_total_bill_empty_cart(cart):
    bill = cart.generate_bill()
    assert bill[-1] == ('', '', 'Grand Total', 0)

def test_total_bill_single_item(cart):
    cart.add_to_cart("Tea", 1)
    bill = cart.generate_bill()
    assert bill[-1] == ('', '', 'Grand Total', 10)

def test_total_bill_multiple_items(cart):
    cart.add_to_cart("Tea", 1)
    cart.add_to_cart("Coffee", 2)
    bill = cart.generate_bill()
    assert bill[-1] == ('', '', 'Grand Total', 40)
```

```
def test_add_zero_quantity(cart):
    cart.add_to_cart("Tea", 0)
    assert cart.show_cart() == {}

def test_add_negative_quantity(cart):
    cart.add_to_cart("Tea", -1)
    assert cart.show_cart() == {}

def test_add_zero_price():
    items = [{"name": "Water", "price": 0}]
    cart = ShoppingCart(items)
    cart.add_to_cart("Water", 1)
    bill = cart.generate_bill()
    assert bill[-1] == ('', '', 'Grand Total', 0)

def test_add_negative_price():
    items = [{"name": "Magic Potion", "price": -10}]
    cart = ShoppingCart(items)
    cart.add_to_cart("Magic Potion", 1)
    bill = cart.generate_bill()
    assert bill[-1] == ('', '', 'Grand Total', 0)

if __name__ == "__main__":
    pytest.main()
```