Joseph Affleck

Professor Yang

CMPSC 132

15 March, 2025

Project-1: Enhanced E-commerce Shopping Cart Project

Description:

The project uses python to create an e-commerce shopping cart with support for physical and digital products. It has a user specific shopping cart, fixed-amount discounts, and percent discounts. The discounts can be applied to the cart total. Items in the cart can be removed, added, or checked out.

Structure:

Class Product: four class attributes; product_id, name, price, quantity

Update quantity method: one argument; updates the quantity of a product to the new quantity

Get product info method: no arguments; prints the info of the product

Class digital product: derived from product class; six class attributes; product_id, name, price, quantity, file_size, download_link

Get product info method; no arguments; obtains info for digital products

Class physical product: derived from product class; seven class attributes; product_id, name, price, quantity, weight, dimensions, shipping_cost

Get product info method: no arguments; obtains info for physical products

Class cart: one class attribute; cart_items

Add product method: one argument; adds a product to the cart

Remove product method: one argument; removes product by product_id

View cart method: no arguments; returns items in the cart

Calculate total method: no arguments; sums the cost of all items in the cart

Class user: three class attributes; user_id, name, instance of the cart class

Add to cart method: one argument; add product to the user's cart

Remove from cart method: one argument; removes product from the user's cart based on product_id

Checkout method: no arguments; calculates total price of the cart

Class discount: one class attribute; total_amount; this class is used as a base for other discount classes

Class percentage discount: derived from discount class; one class attribute; percentage

Apply discount method: one argument; applies the percentage discount to the total

Class fixed amount discount: derived from discount class; one class attribute; amount

Apply discount method; one argument; applies the fixed discount amount to the total

Instructions:

1. Add products (they can be either digital or physical) by inputting relevant information depending on whether the product is physical or digital.
2. Change the product quantity by inputting a new quantity with the update quantity method
3. Create a user by inputting the user id and username
4. Add the products you wish to buy to the cart by inputting the name of the product
5. You can view the cart by accessing the view cart method with the user object
6. You can add discount (fixed or percentage based) by either entering the amount or percentage respectively.
7. Checkout by inputting user name and accessing the checkout method

Verification:

```
[23] #digital products
     ebook=DigitalProduct(1,"The Hunger Games", 9.99, 30, "10mb", "www.downloadhgbook.com")
     ebook2=DigitalProduct(2, "Harry Potter", 14.99, 25, "15mb","www.downloadhpbook.com")

     #physical products
     shoes=PhysicalProduct(3, "Running shoes", 94.99, 16, "0.5lbs", "10x5x6 in.", 10.00)
     shirt=PhysicalProduct(4, "White T-shirt", 8.99, 100, "0.32lbs", "28x20 in.", 3.00)
     laptop=PhysicalProduct(5, "Laptop", 699.99, 18, "4lbs", "14x10x1 in.", 20)

     #2 user instances
     user1=User(1,"Bob")
     user2=User(2,"Jill")

     #adding products to user1's cart
     user1.add_to_cart(ebook)
     user1.add_to_cart(ebook2)

     #adding products to user2's cart
     user2.add_to_cart(shoes)
     user2.add_to_cart(shirt)
     user2.add_to_cart(laptop)
```

```
[73] #verifying the user's carts
     print(f"First user's cart: {user1.cart.view_cart()}")
     print(f"Second user's cart: {user2.cart.view_cart()}")

     First user's cart: ['The Hunger Games', 'Harry Potter']
     Second user's cart: ['Running shoes', 'White T-shirt', 'Laptop']
```

```
[63] #percentage discount instance
     percent=PercentageDiscount(10) # 10% discount

     #fixed amount discount
     fixed=FixedAmountDiscount(5) # $5 discount
```

```
[64] #creating discounts
     user1total=user1.cart.calculate_total()
     user2total=user2.cart.calculate_total()
     user1discount=percent.apply_discount(user1total)
     user2discount=fixed.apply_discount(user2total)
```

```
[65] #checkout before discount
     user1.checkout()
     user2.checkout()

     Total amount: $24.98
     Total amount: $803.97
```

```
[66] #checkout with discount
     print(f"User 1 discounted total: {user1discount}")
     print(f"User 2 discounted toal: {user2discount}")

     User 1 discounted total: $22.48
     User 2 discounted toal: $798.97
```

```
     #verifying clearing of the carts
     print(f"First user's cart: {user1.cart.view_cart()}")
     print(f"Second user's cart: {user2.cart.view_cart()}")

     First user's cart: []
     Second user's cart: []
```

Conclusion:

This project gave me a better understanding of classes in python. I was able to use encapsulation to keep the cart class private, polymorphism to create different types of discounts, inheritance to create physical and digital products based on the original product class, and abstraction in the user class to hide the complexity of the cart operations. Overall, this project helped me understand classes, encapsulation, polymorphism, inheritance, and abstraction.