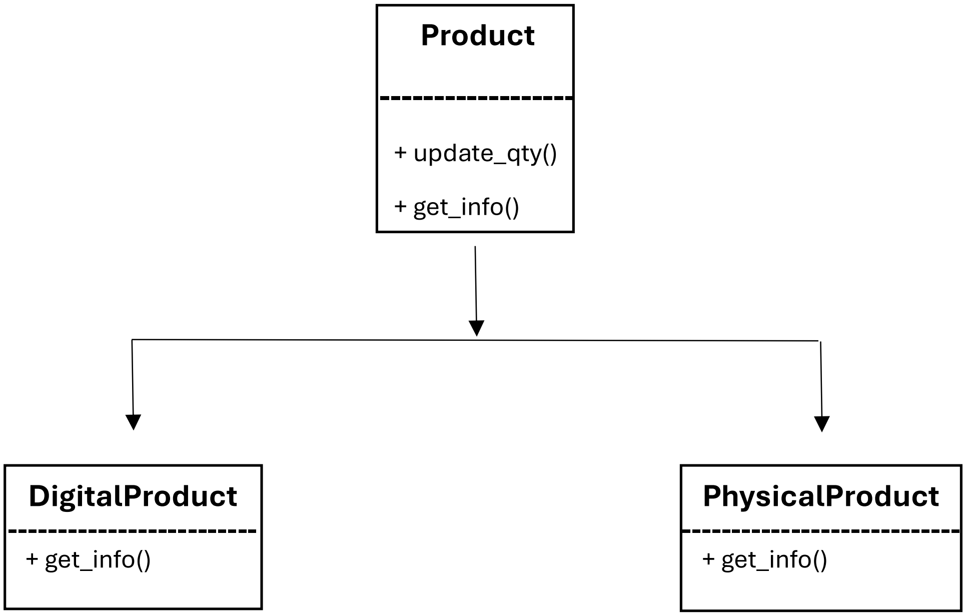
Project Report: Enhanced Shopping Cart

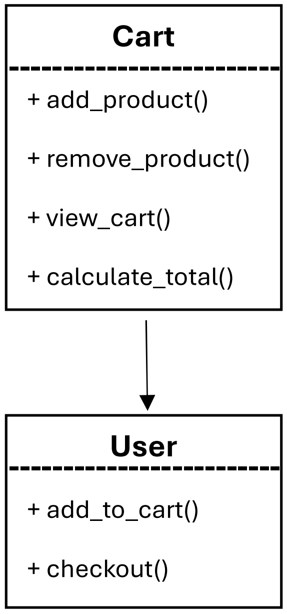
Prasun Gajurel

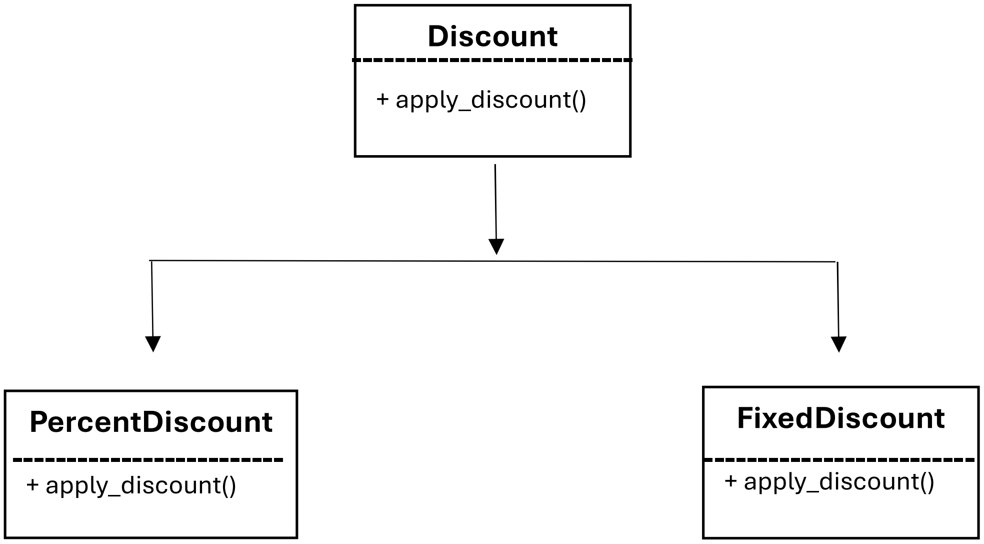
1. Description of the Project:

The project involved designing and programming an enhanced ecommerce shopping cart system using object-oriented programming. The system includes a product catalog, a shopping cart mechanism, user management, and the ability to apply discounts as well. The code was written through the four key object-oriented programming principles, which include abstraction, encapsulation, inheritance, and polymorphism. This system also allows users to add digital and physical products to their shopping cart, apply different types of discounts, and to check out the products in the shopping cart. The system was able to perform these tasks from the well-defined classes that represent products, users, and carts.

2. Structure of the Code:







The following code is structured into many different classes:

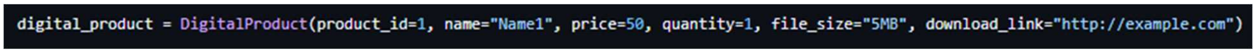
Brief Summary of the Following Developed Classes

- **Product**: A base class with attributes and methods for all products. Contains basic attributes like product ID, name, price, and quantity.
- **DigitalProduct**: Represents downloadable products with a file size and a download link attributes

- **PhysicalProduct**: Represents real products with weight and manufacturer attributes.
- **Cart**: Handles and manages product storage, calculating the total, and applying different types of discounts.
- **User**: Connects a shopping cart to a user and manages the shopping cart tasks.
- **Discount**: Abstract class that defines the percent discount and fixed amount discount methods.
- **PercentDiscount**: Applies a percentage-based discount.
- **FixedAmountDiscount**: Applies a fixed value discount.

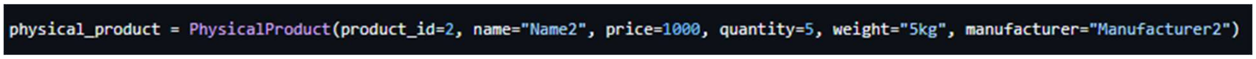
3. Instructions and Examples of the Code to Use the System

1. Instantiate the digital or physical products:
   - Creating a Digital Product:



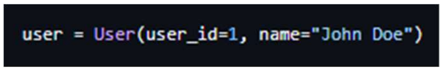
```
digital_product = DigitalProduct(product_id=1, name="Name1", price=50, quantity=1, file_size="5MB", download_link="http://example.com")
```

   - Creating a Physical Product:



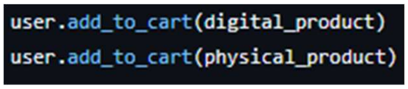
```
physical_product = PhysicalProduct(product_id=2, name="Name2", price=1000, quantity=5, weight="5kg", manufacturer="Manufacturer2")
```

2. Create user instances and add products to their carts:
   - Creating a User:



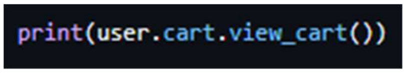
```
user = User(user_id=1, name="John Doe")
```

   - Adding Products to the User's Cart:



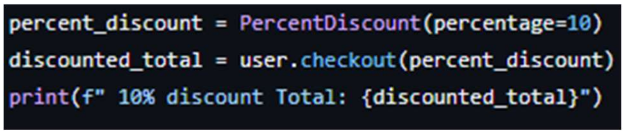
```
user.add_to_cart(digital_product)
user.add_to_cart(physical_product)
```

3. View cart contents:



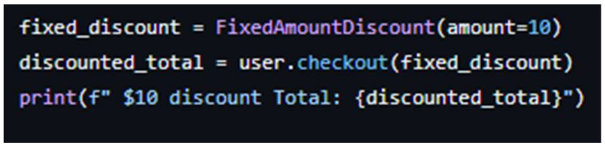
```
print(user.cart.view_cart())
```

4. Apply discounts and proceed to checkout:
   - Applying a Percentage Discount:



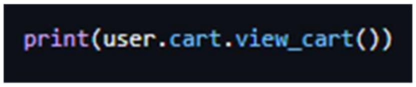
```
percent_discount = PercentDiscount(percentage=10)
discounted_total = user.checkout(percent_discount)
print(f" 10% discount Total: {discounted_total}")
```

- Applying a Fixed Amount Discount:



```
fixed_discount = FixedAmountDiscount(amount=10)
discounted_total = user.checkout(fixed_discount)
print(f" $10 discount Total: {discounted_total}")
```

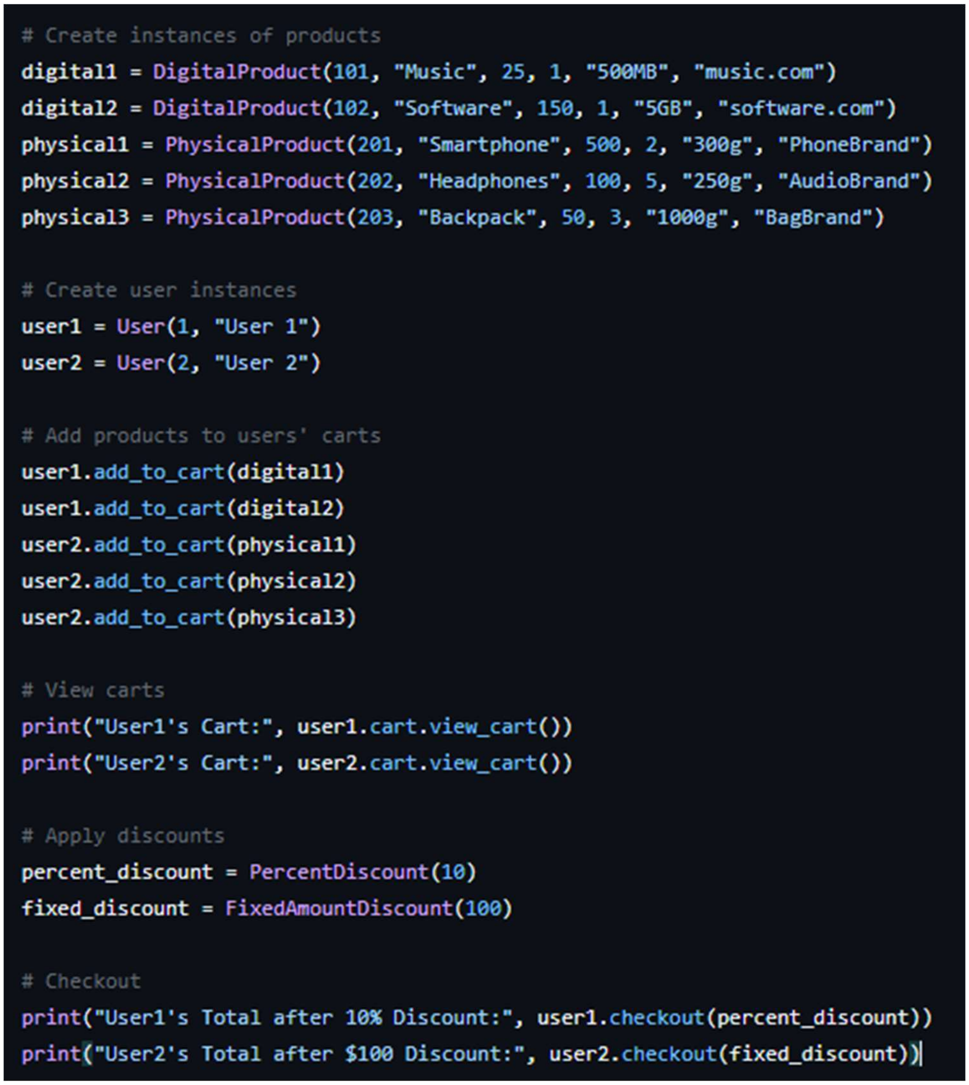
5. The cart is cleared after checkout:



```
print(user.cart.view_cart())
```

4. Verification of Code Functionality

Sample Results:



```
# Create instances of products
digital1 = DigitalProduct(101, "Music", 25, 1, "500MB", "music.com")
digital2 = DigitalProduct(102, "Software", 150, 1, "5GB", "software.com")
physical1 = PhysicalProduct(201, "Smartphone", 500, 2, "300g", "PhoneBrand")
physical2 = PhysicalProduct(202, "Headphones", 100, 5, "250g", "AudioBrand")
physical3 = PhysicalProduct(203, "Backpack", 50, 3, "1000g", "BagBrand")

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

# Add products to users' carts
user1.add_to_cart(digital1)
user1.add_to_cart(digital2)
user2.add_to_cart(physical1)
user2.add_to_cart(physical2)
user2.add_to_cart(physical3)

# View carts
print("User1's Cart:", user1.cart.view_cart())
print("User2's Cart:", user2.cart.view_cart())

# Apply discounts
percent_discount = PercentDiscount(10)
fixed_discount = FixedAmountDiscount(100)

# Checkout
print("User1's Total after 10% Discount:", user1.checkout(percent_discount))
print("User2's Total after $100 Discount:", user2.checkout(fixed_discount))
```

Expected Output:

```
User1's Cart: ['Product ID: 101, Name: Music, Price: 25, Quantity: 1, File Size: 500MB, Download Link: music.com', 'Product ID: 102, Name: Software, Price: 150, Quantity: 1, File Size: 5GB, Download Link: software.com']

User2's Cart: ['Product ID: 201, Name: Smartphone, Price: 500, Quantity: 2, Weight: 300g, Manufacturer: PhoneBrand', 'Product ID: 202, Name: Headphones, Price: 100, Quantity: 5, Weight: 250g, Manufacturer: AudioBrand', 'Product ID: 203, Name: Backpack, Price: 50, Quantity: 3, Weight: 1000g, Manufacturer: BagBrand']

User1's Total after 10% Discount: 157.5

User2's Total after $100 Discount: 550
```

5. Discussion and Conclusion:

From this project, I was able to learn many helpful skills and I became much more capable of programming and creating coding projects. I also found out what my strengths and weaknesses are when it comes to programming a large project such as this one and I was able to learn along the way as I continued with completing the project. Another finding that I had was that I figured out how to think more critically when I come across a problem or challenge that I have trouble solving. Some of the challenges that I faced with this project were making sure that each of the classes was working properly and that I hadn’t made any mistakes that would prevent me from getting the correct result. The other challenge that I faced was using abstraction for the discount classes and subclasses. The reason why I was having a bit of trouble with this is because this was a relatively new topic that we learned in class recently, and I didn’t fully understand it. But as I looked more into it, I was able to figure out what to do and I gained a much better understanding. I believe that we could improve this ecommerce system by including more groups of items that are more specific to what you can buy online, such as a section dedicated for users to buy food or a specific section for users to buy movies and so on. We could also improve it by adding more user interaction such as a rating system or customer support. Overall, I thought that this project was very helpful, and I learned many important skills that I will continue to use on other programming projects and assignments.