Project 1 Proposal

Cynthia Hu

I will create a program for a coffee shop which enables customer to order various products offered and pay in different methods. Customers can order coffee or desserts; for coffee, they can choose the size and flavor. They can pay using cash, credit/debit card or points to redeem a free cup of coffee.

I Pseudocode major classes and functions below with comments.

```
# product class is for the products offered by the coffee shop which has attributes of name and price and order method

class product:

    def __init__(self, name, price):

        pass

    def order(self, name, price, quantity):

        # return the total price for the products ordered

        pass



# product class is also a parent class for coffee and dessert classes as coffee and dessert are two categories of products offered;

# though they share some attributes and method, coffee and dessert products can have different attributes.

# coffee class has size and flavor attributes

class coffee(product):

    def __init__(self, name, price):

        pass

    def order(self, name, price, quantity):

        # product name could be at lower granular level, such as latti with soymilk, medium size

        # return the total price for the coffee ordered
        pass


class dessert(product):
    def order(self, name, price, quantity):
        # return the total price for the dessert ordered
        pass


# payment class is for customers' payment option, like cash, credit/debit card and points and has pay method
class payment:
    def __init__(self, option):
        pass
    # return order information and payment confirmation to the customer
    def pay(self, order, option):
        pass


# for each payment option, define a subclass of payment class mainly to customize pay method
class cash(payment):
    def pay(self, order, cash_paid):
        # return  order information and payment confirmation and exchange to the customer
        pass


class card(payment):
    # return order information and payment confirmation to the customer
    def pay(self, order):
        pass

    # for using points to redeem the ordering, need to extract and update customer point data
class point(payment):
    def __init__(self, points):
        pass
    # get current point information of the customer
    def get(self, customer_id):
        pass
    # accumulate points for this order
    def add(self, customer_id, order):
        pass
    # return order information, points redeemed and remained information to the customer
    def pay(self, order, customer_id):
        pass

# Customer class is for maintaining customer information when necessary and can be called by point class
class customer:
    def __init__(self, customer_id, point):
        pass
```

My Questions and Thoughts

1. do I need payment class as the parent class or just use three different classes for payment? Especially I'm not sure whether the purpose of parent's pay method is well defined

2. shall I create order class instead of product class so that I could combine different products in one order easily?

3. It seems that customer class may not be necessary in this scenario. I can define and maintain required customer information within point class

4. I’m not sure which class/subclass shall I add __init__? I should use it wherever I need instance attributes?