

Main.py

Main.py

```
from ShoppingCart import ShoppingCart
from DiscountStrategy import PercentageDiscount, FixedDiscount, ProgressiveDiscount, CouponDiscount
```

```
cart = ShoppingCart()
cart.add_item(100)
cart.add_item(50)
print("Total without discounts:", cart.get_total())
assert cart.get_total() == 150
```

```
cart_with_percentage = ShoppingCart(discounts=[PercentageDiscount(10)])
cart_with_percentage.add_item(100)
cart_with_percentage.add_item(50)
print("Total with 10% discount:", cart_with_percentage.get_total())
assert cart_with_percentage.get_total() == 135.0
```

```
cart_with_fixed = ShoppingCart(discounts=[FixedDiscount(20)])
cart_with_fixed.add_item(100)
cart_with_fixed.add_item(50)
print("Total with $20 fixed discount:", cart_with_fixed.get_total())
assert cart_with_fixed.get_total() == 230.0
```

```
cart_with_progressive_high = ShoppingCart(discounts=[ProgressiveDiscount(1)])
cart_with_progressive_high.add_item(600)
print("Total with progressive discount on $600:",
      cart_with_progressive_high.get_total())
```



```

assert cart-with-progressive-high.get-total() == 510.0
cart-with-progressive-low = ShoppingCart(discount=[ProgressiveDiscount(1)])
cart-with-progressive-low.add-item(80)
print("Total with progressive discount on $80:",
      cart-with-progressive-low.get-total())
assert cart-with-progressive-low.get-total() == 76.0
cart-with-progressive-mid = ShoppingCart(discount=[ProgressiveDiscount(1)])
cart-with-progressive-mid.add-item(150)
print("Total with progressive discount on $150:",
      cart-with-progressive-mid.get-total())
assert cart-with-progressive-mid.get-total() == 135.0

```

```

cart-with-coupon = ShoppingCart(discounts=[CouponDiscount("Save 10%")])
cart-with-coupon.add-item(200)
print("Total with Save 10% coupon on $200:", cart-with-coupon.get-total())
assert cart-with-coupon.get-total() == 180.0

```

```

cart-with-multiple = ShoppingCart(
    discounts=[PercentDiscount(10), FixedDiscount(15)])
cart-with-multiple.add-item(300)
print("Total with 10% and $15 fixed discount on $300:",
      cart-with-multiple.get-total())
assert cart-with-multiple.get-total() == 255.0

```

```

cart-with-multiple-coupons = ShoppingCart(
    discounts=[CouponDiscount("Save 20%"), CouponDiscount("Save 30%")])
cart-with-multiple-coupons.add-item(400)
print("Total with Save 20% and Save 30 coupons on $400:", cart-with-multiple-coupons.get-total())
assert cart-with-multiple-coupons.get-total() == 290.0

```

ShoppingCart.py

ShoppingCart.py

```
from DiscountStrategy import DiscountStrategy
from typing import List
```

```
class ShoppingCart:
```

```
    def __init__(self, discounts: List[DiscountStrategy] = []):
        self.items = []
        self.discounts = discounts
```

```
    def add_item(self, price):
        self.items.append({'price': price})
```

```
    def get_total(self) -> float:
        total = sum(it['price'] for i in self.items)
        for discount in self.discounts:
            total -= discount.calculate_discount(total)
        return max(total, 0.0)
```


DiscountStrategy.py

DiscountStrategy.py

```
from abc import ABC, abstractmethod
```

```
class DiscountStrategy(ABC):
```

```
    @abstractmethod
```

```
    def calculate_discount(self, amount: float) -> float:  
        pass
```

```
class PercentageDiscount(DiscountStrategy):
```

```
    def __init__(self, percentage: float):
```

```
        self.percentage = percentage
```

```
    def calculate_discount(self, amount: float) -> float:  
        return amount * (self.percentage / 100)
```

```
class FixedDiscount(DiscountStrategy):
```

```
    def __init__(self, fixed_amount: float):
```

```
        self.fixed_amount = fixed_amount
```

```
    def calculate_discount(self, amount: float) -> float:  
        return min(self.fixed_amount, amount)
```

```

class ProgressiveDiscount(DiscountStrategy):
    def calculate_discount(self, amount: float) -> float:
        if amount > 500:
            return PercentageDiscount(15).calculate_discount(amount)
        elif amount > 100:
            return PercentageDiscount(10).calculate_discount(amount)
        return PercentageDiscount(5).calculate_discount(amount)

```

```

class CouponDiscount(DiscountStrategy):
    def __init__(self, coupon_code: str):
        self.coupon_code = coupon_code
        self.coupons = {
            "save10%": PercentageDiscount(10),
            "save20%": PercentageDiscount(20),
            "save30": FixedDiscount(30),
        }

```

```

    def calculate_discount(self, amount: float) -> float:
        discount_strategy = self.coupons.get(self.coupon_code)
        if discount_strategy:
            return discount_strategy.calculate_discount(amount)
        return 0.0

```