```java
abstract class Item {
    protected double basePrice;
    double getPrice() { return basePrice; }
    boolean isPerishable() { return false; }
}


class Foodstuff extends Item {
    Foodstuff(double price) { this.basePrice = price; }
    boolean isPerishable() { return true; }
}


class Book extends Item {
    private String title;
    private String ISBN;
    Book(String title, String ISBN, double price) {
        this.basePrice = price;
        this.title = title;
        this.ISBN = ISBN;
    }
}


class ShoppingCart {
    private ArrayList<Item> contents;
    private User owner;
    private PricingStrategy strat;

    ShoppingCart(User owner) {
        this.owner = owner;
        if (!owner.isPrimeUser()) {
            strat = new DefaultPricingStrategy();
        } else {
            strat = new FixedDiscountPricingStrategy(.2);
        }
        contents = new ArrayList<Item>();
    }

    double getPrice() {
        return strat.computePrice(this);
    }

    void add(Item item, int quantity) {
        for (int i=0; i<quantity; i++) { contents.add(item); }
    }

    void add(Item item) { add(item,1); }

    void setPricingStrategy(PricingStrategy newStrategy) {
        strat = newStrategy;
    }

    public String toString() {
        return "A " + contents.size() + "-item, $" + getPrice() +
            " cart owned by " + owner.getName();
    }

    ArrayList<Item> getContents() {
        return contents;
    }
}
```

```java
interface PricingStrategy {
    double computePrice(ShoppingCart sc);
}


class DefaultPricingStrategy implements PricingStrategy {
    protected DefaultPricingStrategy() { }
    public double computePrice(ShoppingCart sc) {
        double price = 0.0;
        for (Item item : sc.getContents()) {
            price += item.getPrice();
        }
        return price;
    }
}


class DiscountPerishablePricingStrategy implements PricingStrategy {
    private double perishableDiscount;
    DiscountPerishablePricingStrategy(double d) {
        this.perishableDiscount = d;
    }
    public double computePrice(ShoppingCart sc) {
        double price = 0.0;
        for (Item item : sc.getContents()) {
            if (item.isPerishable()) {
                price += (1 - perishableDiscount) * item.getPrice();
            } else {
                price += item.getPrice();
            }
        }
        return price;
    }
}


class FixedDiscountPricingStrategy extends DefaultPricingStrategy implements PricingStrategy {
    private double fixedDiscount;
    FixedDiscountPricingStrategy(double d) {
        super();
        this.fixedDiscount = d;
    }
    public double computePrice(ShoppingCart sc) {
        return super.computePrice(sc) * (1 - fixedDiscount);
    }
}


class User {
    private String name;
    private boolean isPrime;
    User(String name) {
        this(name, false);
    }
    User(String name, boolean isPrime) {
        this.name = name;
        this.isPrime = isPrime;
    }
    public String getName() { return name; }
    boolean isPrimeUser() { return isPrime; }
}
```

```java
class Main {

    private static ArrayList<ShoppingCart> carts =
        new ArrayList<ShoppingCart>();

    public static void main(String args[]) {

        Book harryPotter = new Book("Harry Potter", "ABC123", 9.99);
        Book gameOfThrones = new Book("A Game of Thrones", "XYZ567", 12.99);
        Book hungerGames = new Book("Hunger Games", "DEF790", 8.99);

        Foodstuff carrots = new Foodstuff(0.56);
        Foodstuff okra = new Foodstuff(0.78);
        Foodstuff kCups = new Foodstuff(29.99);

        User offset = new User("Offset");
        User cardiB = new User("Cardi B", true);

        ShoppingCart cardisCart = new ShoppingCart(cardiB);
        ShoppingCart cardisOtherCart = new ShoppingCart(cardiB);
        ShoppingCart offsetsCart = new ShoppingCart(offset);
        carts.add(offsetsCart);
        carts.add(cardisCart);
        carts.add(cardisOtherCart);

        cardisCart.add(gameOfThrones,1);
        cardisCart.add(hungerGames,3);
        cardisOtherCart.add(carrots,20);
        cardisOtherCart.add(okra,5);
        cardisOtherCart.add(okra,5);
        cardisOtherCart.add(kCups);
        offsetsCart.add(gameOfThrones);
        offsetsCart.add(harryPotter);
        offsetsCart.add(hungerGames);

        printCarts();
        enableHolidayDiscounts();
        System.out.println("The holiday season is here!");
        printCarts();
    }

    private static void printCarts() {
        for (ShoppingCart cart : carts) {
            System.out.println(cart);
        }
    }

    private static void enableHolidayDiscounts() {
        for (ShoppingCart cart : carts) {
            cart.setPricingStrategy(
                new DiscountPerishablePricingStrategy(.5));
        }
    }
}
```