```java
//CODE BY AKSHAT GARG 12/22/24
import java.util.*;

/**
 * The ShoppingCart class has an ArrayList of Items. You will write a new class
 * DiscountedItem that extends Item. This code is adapted
 * https://practiceit.cs.washington.edu/problem/view/bjp4/chapter9/e10-DiscountBill
 */
public class Tester
{
    public static void main(String[] args)
    {
        ShoppingCart cart = new ShoppingCart();
        cart.add(new Item("bread", 3.25));
        cart.add(new Item("milk", 2.50));
        cart.add(new DiscountedItem("ice cream", 4.50, 1.50));
        cart.add(new DiscountedItem("apples", 1.35, 0.25));

        cart.printOrder();
    }
}

class DiscountedItem extends Item
{
    // add an instance variable for the discount
    double discount;
    // Add constructors that call the super constructor
    public DiscountedItem()
    {
        super();
        discount = 0.0;
    }
    public DiscountedItem(String name, double price, double disc)
    {
        super(name,price);
        discount = disc;
    }
    // Add get/set methods for discount
    public double getDiscount()
```

```
Order Items:
   bread $3.25
   milk $2.50
   ice cream $4.50($1.50)
   apples $1.35($0.25)
Sub-total: $11.60
Discount: $1.75
Total: $9.85
```

Result	Expected	Actual	Notes
Passed	Order Items: bread $3.25 milk $2.50 ice cream $1.50 ($1.50) apples $0.25 ($0.25) Sub-total: $7.50 Discount: $1.75 Total: $5.75	Order Items: bread $3.25 milk $2.50 ice cream $4.50($1.50) apples $1.35($0.25) Sub-total: $11.60 Discount: $1.75 Total: $9.85	Checking that DiscountedItem objects were added to ArrayList
Passed	Order Items: bread $3.25 milk $2.50 ice cream $1.50 ($1.50) apples $0.25 ($0.25) Sub-total: $7.50 Discount: $1.75 Total: $5.75	Order Items: bread $3.25 milk $2.50 ice cream $4.50($1.50) apples $1.35($0.25) Sub-total: $11.60 Discount: $1.75 Total: $9.85	Checking that countDiscountedItems() was added to output
Passed	true	true	Checking that code contains public int countDiscountedItems()
Passed	true	true	Checking that code contains if (* instanceof DiscountedItem)

You got 4 out of 4 correct. 100.00%

Activity: 9.6.1.1 ActiveCode (challenge-9-6-shopping2)

```
    // Add get/set methods for discount
    public double getDiscount()
    {
        return discount; // return discount here instead of 0
    }
    public void setDiscount(double newDiscount)
    {
        discount = newDiscount;
    }

    // Add a toString() method that returns a call to the super toString
    // and then the discount in parentheses using the super.valueToString() method
    public String toString()
    {
        return super.toString() + "(" +  super.valueToString(discount) + ")";
    }
}

// Add a method called countDiscountedItems()
class ShoppingCart
{
    private ArrayList<Item> order;
    private double total;
    private double internalDiscount;

    public ShoppingCart()
    {
        order = new ArrayList<Item>();
        total = 0.0;
        internalDiscount = 0.0;
    }

    public void add(Item i)
    {
        order.add(i);
        total += i.getPrice();
        if (i instanceof DiscountedItem)
            internalDiscount += ((DiscountedItem) i).getDiscount();
    }
```

```
87      //CODE BY AKSHAT GARG
88      public int countDiscountedItems()
89      {
90          int count = 0;
91          for(Item item:order){
92              if (item instanceof DiscountedItem){
93                  count++;
94              }
95          }
96          return count;
97      }
98      public String discountToString()
99      {
100         return orderToString()
```

```
Order Items:
   bread $3.25
   milk $2.50
   ice cream $4.50($1.50)
   apples $1.35($0.25)
Sub-total: $11.60
Discount: $1.75
Total: $9.85
```

Result	Expected	Actual	Notes
Passed	Order Items: bread $3.25 milk $2.50 ice cream $1.50 ($1.50) apples $0.25 ($0.25) Sub-total: $7.50 Discount: $1.75 Total: $5.75	Order Items: bread $3.25 milk $2.50 ice cream $4.50($1.50) apples $1.35($0.25) Sub-total: $11.60 Discount: $1.75 Total: $9.85	Checking that DiscountedItem objects were added to ArrayList
Passed	Order Items: bread $3.25 milk $2.50 ice cream $1.50 ($1.50) apples $0.25 ($0.25) Sub-total: $7.50 Discount: $1.75 Total: $5.75	Order Items: bread $3.25 milk $2.50 ice cream $4.50($1.50) apples $1.35($0.25) Sub-total: $11.60 Discount: $1.75 Total: $9.85	Checking that countDiscountedItems() was added to output
Passed	true	true	Checking that code contains public int countDiscountedItems()
Passed	true	true	Checking that code contains if (* instanceof DiscountedItem)

You got 4 out of 4 correct. 100.00%

Had an issue with the constructors i set up in the previous exercises for DiscountedItem, otherwise a few syntax errors and a lossy conversion issue, and I did pretty okay. No issues with the actual methods and all.