Object Classes: Notes on designing Shopping Cart

Additions for Garfield HS AP CS to supplement: Building Java Programs, by Stuart Reges and Marty Stepp (http://www.buildingjavaprograms.com/). (March 2013, Mr. Bergquist)

Write down your thoughts

- First let’s look at the Shopping Cart Project – you will be writing four Classes that interact.

- Take about 5 minutes to answer the following questions:

  - **What is the point of object oriented programming?**

  - **Discuss 3-4 concepts you have learned that will apply to the Shopping Cart assignment**

NullPointerException

- Before a variable of an object type is initialized with new, it holds a reference to `null` (nothing!)

- Calling a method on null gives a NullPointerException

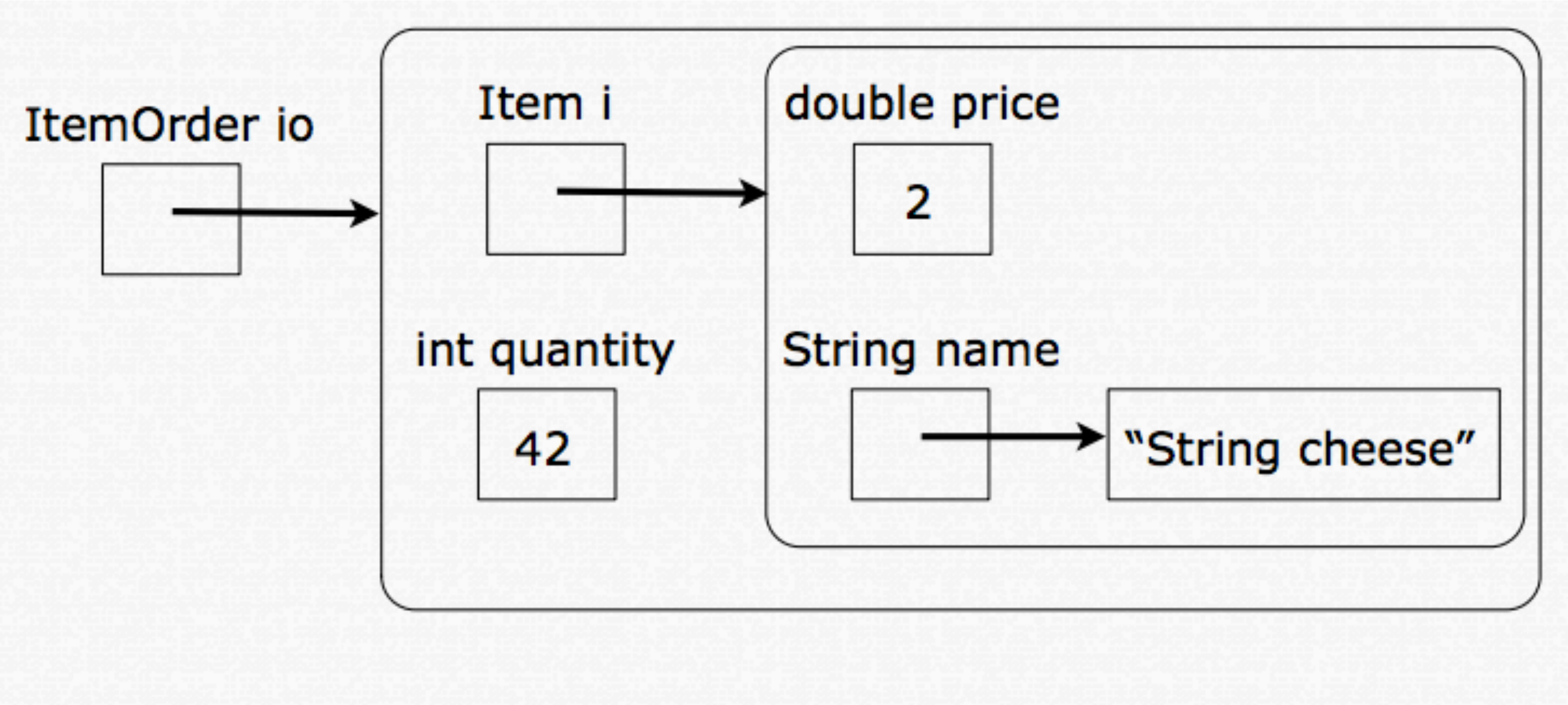
- Make sure to initialize fields in your constructor – 2 phase creation

```
public class Student {
   private ArrayList<Integer> grades; // phase 1

   public Student() {   // Student Constructor
      grades = new ArrayList<Integer>();  // phase 2
   }
   ...
}
```

Reference to an Object

- Variables don’t directly hold objects, they hold the memory address of objects.



Object-Oriented Design

- Design is critical in OOD and key to success; take the time to structure your design (i.e. draw & think it out!)
- Think first what are the **nouns** in the specification? (ex: Item) These are likely your Objects or possibly key data in an object.
- Which **responsibilities** belong to which nouns? These are likely to become their methods.
- What are the **relationships** between nouns (Objects)? Draw it out that will help you visualize how they are interconnected.

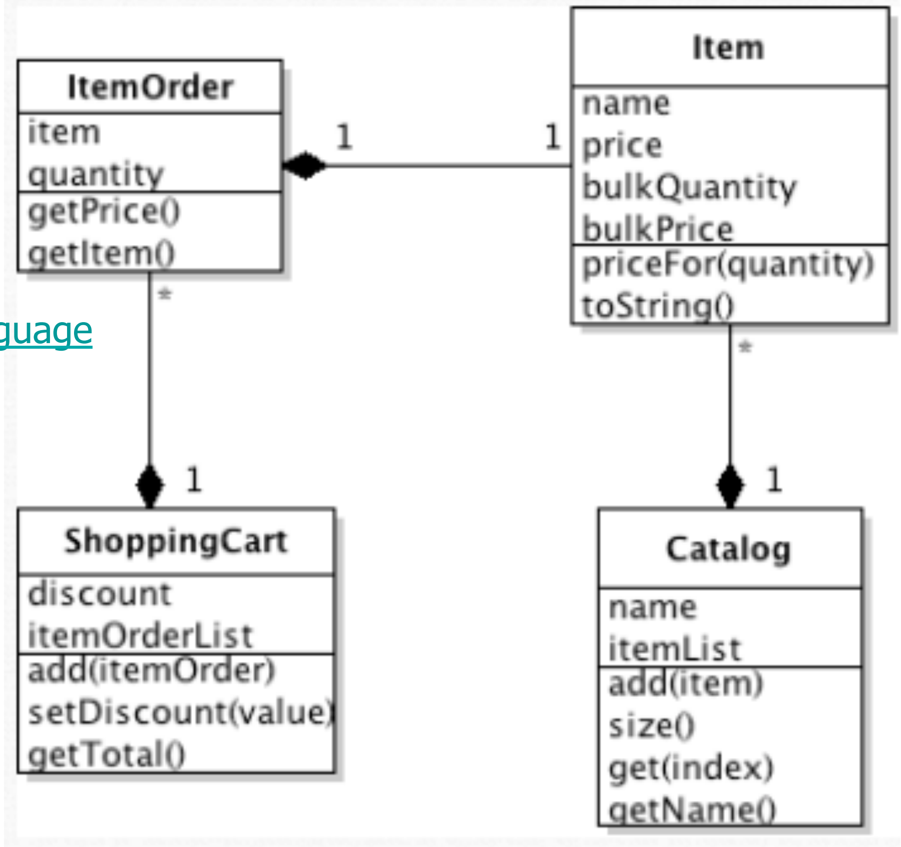
Objects Composed of Objects

- You will find that several Objects have a “Has-a” relationship
  - An ItemOrder “has an” Item (one to one relationship)
  - A ShoppingCart “has many” ItemOrders (one to many relationship)

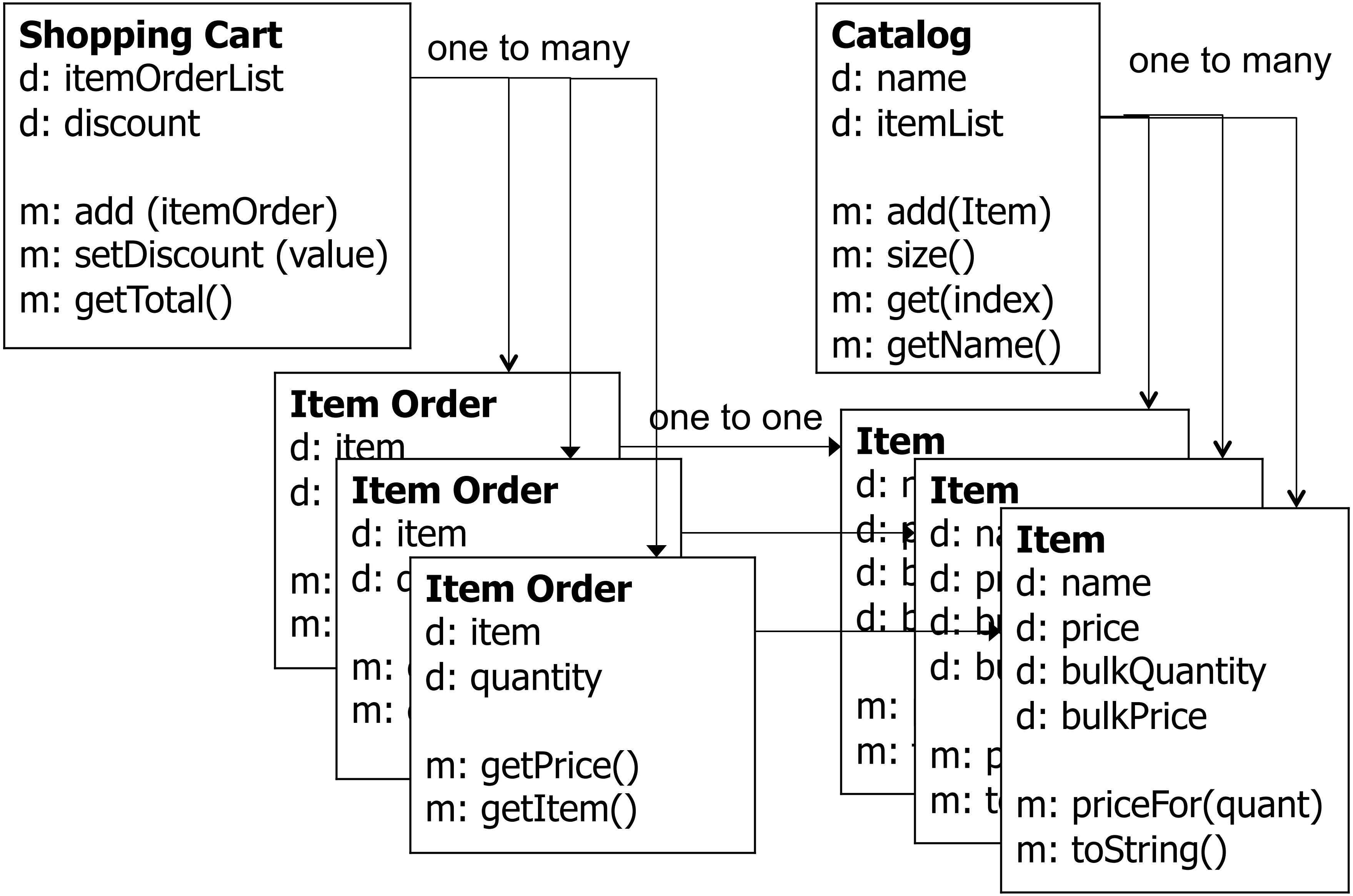
Put in an UML Diagram:

Unified Modeling Language

http://en.wikipedia.org/wiki/Unified_Modeling_Language



Top Down Design



Shopping Cart
d: itemOrderList
d: discount

m: add (itemOrder)
m: setDiscount (value)
m: getTotal()

one to many

Catalog
d: name
d: itemList

m: add(Item)
m: size()
m: get(index)
m: getName()

one to many

Item Order
d: item
d:
m:
m:

Item Order
d: item
d: q
m:
m:

Item Order
d: item
d: quantity

m: getPrice()
m: getItem()

one to one

Item
d: n
d: p
d: b
d: b
m:
m:

Item
d: na
d: pr
d: bu
d: bu
m: p
m: t

Item
d: name
d: price
d: bulkQuantity
d: bulkPrice

m: priceFor(quant)
m: toString()

Bottom Up Implementation

- Write the simplest classes without dependencies first – the lowest level. Which is that for Shopping Cart?
- **Test** each new Object with a Client test to make sure it works. You are required to turn in `ShoppingTest.java`
- Link these simple system pieces to the next level up, larger Object subsystems, and keep building upward.
- **Test** at each level to make sure you're not building off a bad foundation – mistakes could cause:
  - take more time finding where the error lies (what level??)
  - you to need to fix more code at multiple levels

Additional Help?

- Any new questions about the Shopping Cart Project?
- Reminder – do not collaborate working with classmates on the project – I want you to solve this. Only talk in concepts not code.

RESOURCES:

- Class Presentations & Lesson handouts
- Class Object Concepts & Vocabulary Sheet

http://www.garfieldcs.com/wordpress/wordpress/wp-content/uploads/2013/02/Class-Objects-Vocabulary-2013.pdf

- Object Class Review Slides

http://www.garfieldcs.com/wordpress/wordpress/wp-content/uploads/2012/02/2012-Feb-27-Object-Class-Review.pdf

- Java Syntax Sheet (now with ArrayList Details)

http://www.garfieldcs.com/wordpress/wordpress/wp-content/uploads/2013/03/AP-CS-Java-Syntax-Summary-5.pdf

More Resources

- Textbook, chapters 8: Objects and 10: ArrayLists
- Supplemental Videos (see our class website for links)
  - Defining a Class
  - Constructors
  - Advanced Instance Methods
  - Encapsulation
  - Removing from an ArrayList
  - Adding to an ArrayList of Integers

- Ask Mr. Bergquist!

Best of luck on the project!