Domain Models: Associations and Attributes

Curt Clifton

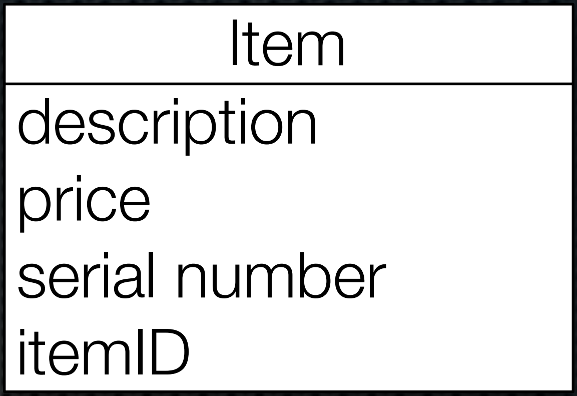
Rose-Hulman Institute of Technology

Q1

Description Classes

- A **description class** contains information that describes something else, e.g., *ProductDescription*
- Example…

Consider…

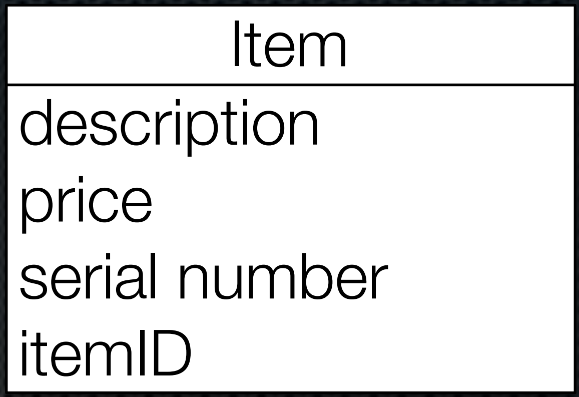


- Assume an *Item* instance represents a physical item in a store
- Item data only recorded within *Item* instances
- When a real-world item is solid, we remove the software *Item* from a collection and it’s garbage collected

Amps that go to 11 are sold out!

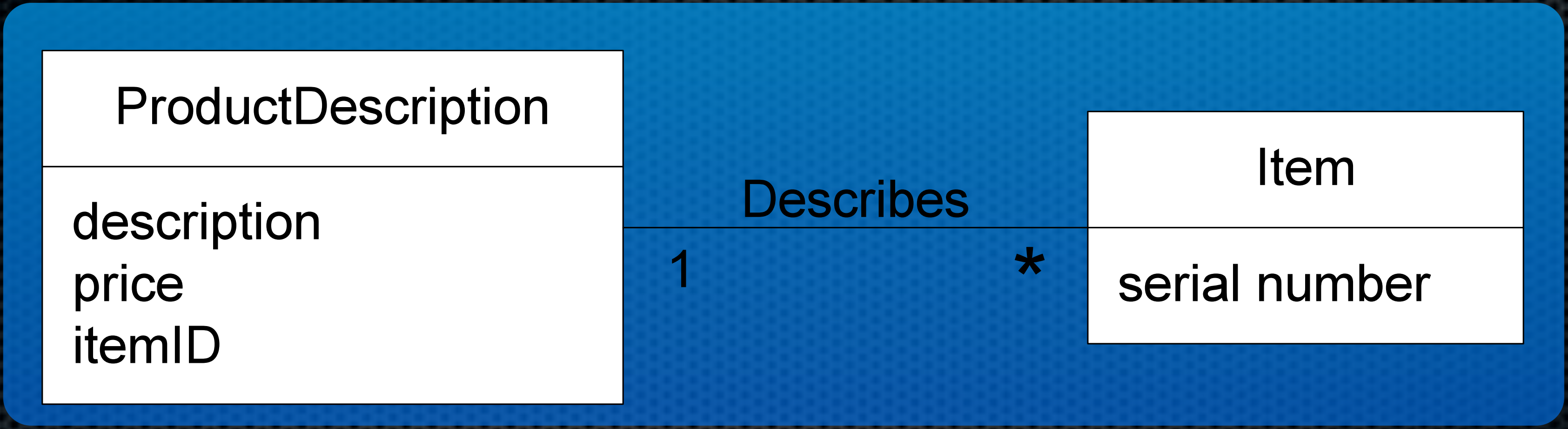
How much for an Amp that goes to 11?

Problems



- Lose memory of the price, etc., if no *Item* instances remain in the system
- Duplicate data
  - Wasted space
  - Error-prone

Solution: Use Description Class



- When information must be retained independent of existence of instances of the described item
- When deleting the described item could result in info. loss
- When it reduces redundant information

Associations

Be Parsimonious

- A relationship between classes that indicate some meaningful connection between **instances** of the classes
- Says that we need some **memory** of the **relationship**

however transient

- A memory in the **real world**, not a software need
- **Not** about data flows, foreign key relationships, instances variables, or software pointers

Q2

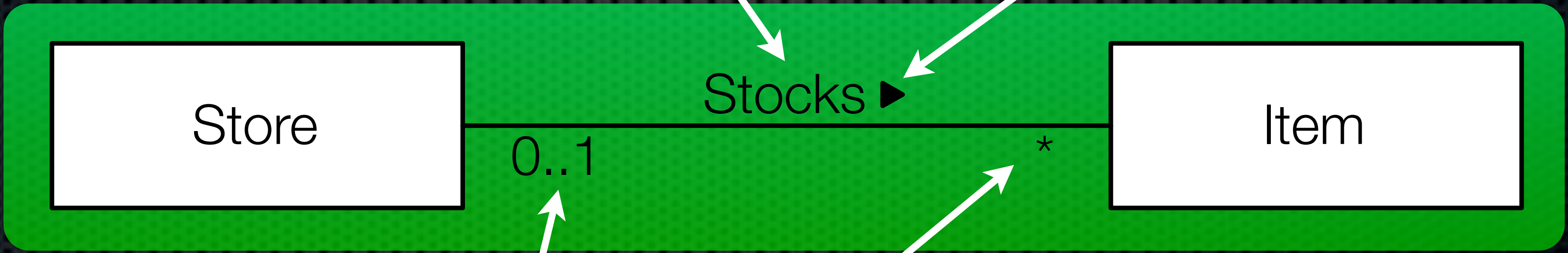
Association Notation

Association name:

- capitalize
- typically camel-case or hyphenated
- use verb phrase
- avoid “has”, “use”

Reading direction:

- typically exclude if association reads left-to-right or top-to-bottom



Multiplicity:

- ‘*’ means “many”
- x..y means from x to y inclusively

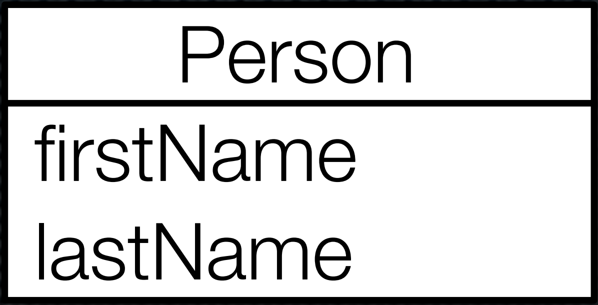
Q3

Common Association Lists

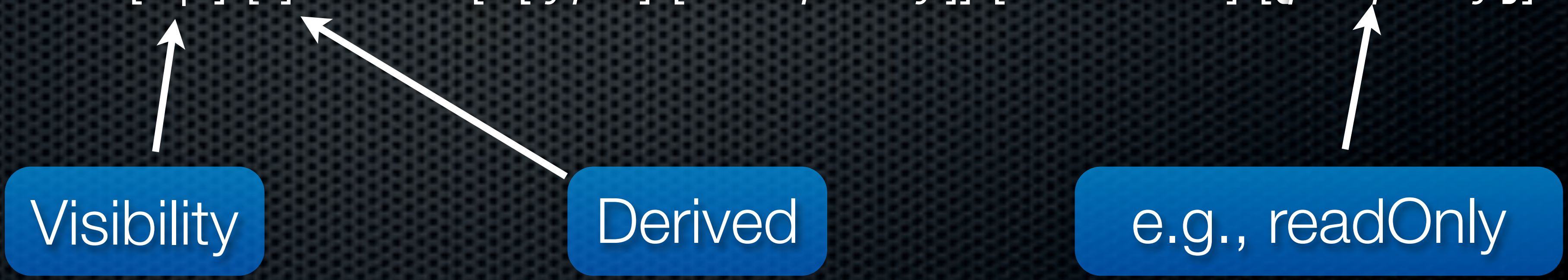
Association Category	POS Examples
A is a transaction related to another transaction *B*	Payment **PaysFor** Sale
A is a line item of a transaction *B*	SalesLineItem **ContainedIn** Sale
A is known/logged/recorded in/on B	Sale **CapturedOn** Register
...	...

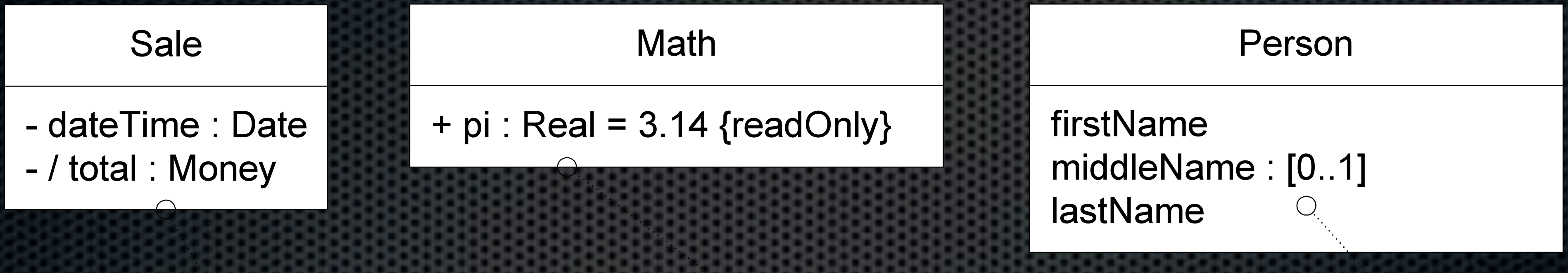
Q4

Attributes



- Include attributes that the **requirements** suggest **need to be remembered**
- Notation (square brackets indicate optional parts):
    - [**+**|**-**] [**/**] *name* [**:** [*type*] [*multiplicity*]] [**=** *default*] [**{***property***}**]



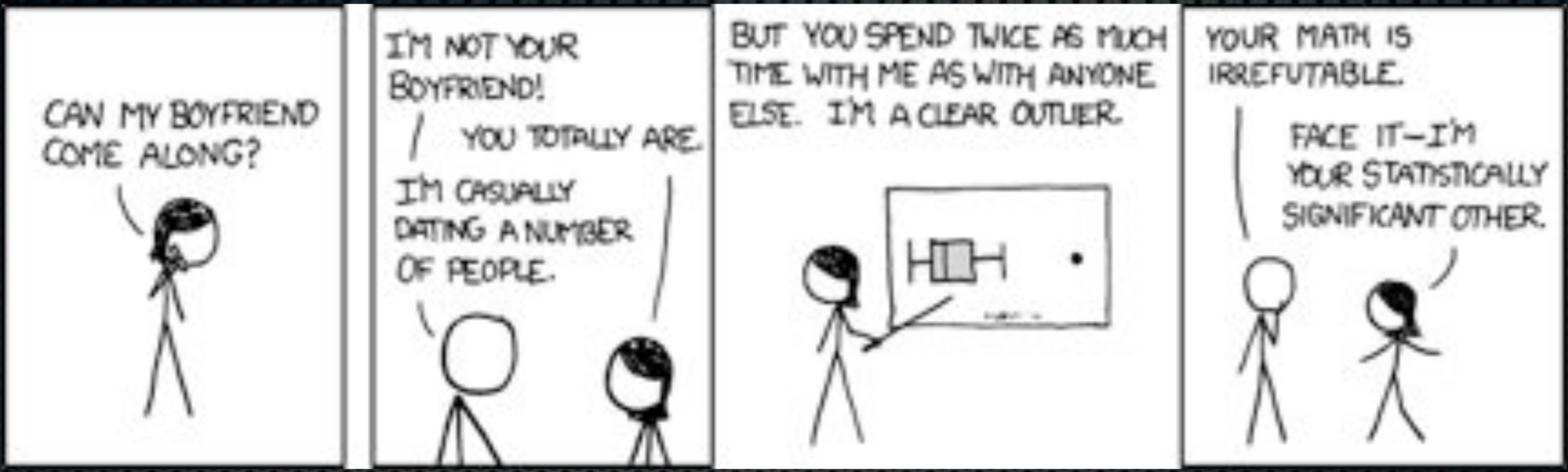


Attribute Examples

What does each part mean?

Q5

Cartoon of the Day


CAN MY BOYFRIEND COME ALONG?
I'M NOT YOUR BOYFRIEND!
YOU TOTALLY ARE.
I'M CASUALLY DATING A NUMBER OF PEOPLE.
BUT YOU SPEND TWICE AS MUCH TIME WITH ME AS WITH ANYONE ELSE. I'M A CLEAR OUTLIER.
YOUR MATH IS IRREFUTABLE.
FACE IT—I'M YOUR STATISTICALLY SIGNIFICANT OTHER.

http://xkcd.com/539/

… okay, but because you said that, we're breaking up.

Related?

In Domain Model, Use **Data Type** Attributes

- Primitive data types:
  - Boolean, String, Real, Integer, …
- Sometimes more complex, but not domain specific:
  - Address, Color, Phone Number, …
- If it's domain specific, use a class and association

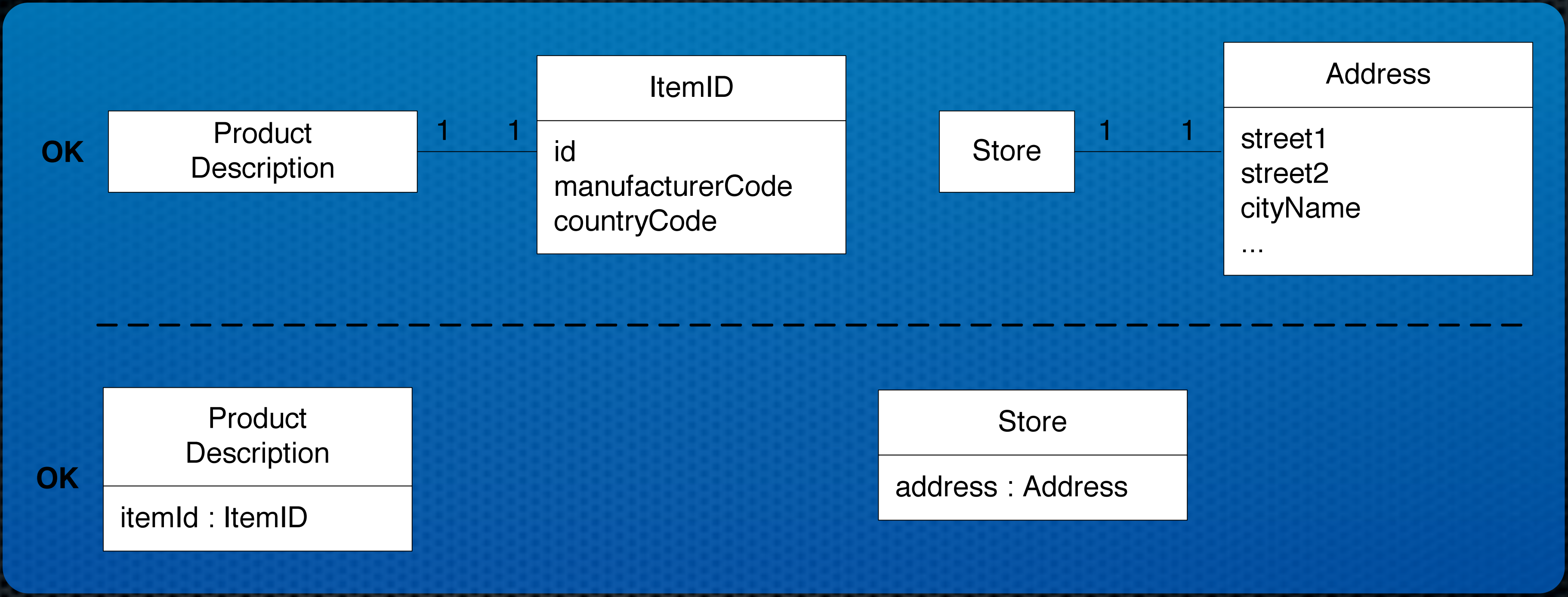
Intuition from code: a ***data type*** is a primitive type, or a complex type where for instances *a* and *b*, ***a.equals(b)* doesn't imply *a == b***

Q6

Create Your Own Complex Data Type When

- It has attributes of its own
- There are operations associated with it (e.g., validation)
- It’s a quantity with a unit

Showing Data Type Attributes



Choose the representation that best communicates with the stakeholders

Q7

Example....

Q8

Domain Model Guidelines, Summarized

- Classes first, then associations and attributes
- Use existing models, category lists, noun phrases
- Include “report objects”, like Receipt, if they’re part of the business rules
- Use terms from the domain
- Don’t send an attribute to do a conceptual class’s job
- Use description classes to remember info. independent of instances and to reduce redundancy
- Use association for relationship that must be remembered
- Be parsimonious with associations
- Name associations with verb phrases, not “has” or “uses”
- Use common association lists
- Use attributes for information that must be remembered
- Use data type attributes
- Define new data types for complex data
- **Communicate with stakeholders**