CSSE 374: Domain Model Refinements and Iteration 3 Preparations

Shawn Bohner

Office: Moench Room F212

Phone: (812) 877-8685

Email: bohner@rose-hulman.edu

Q1

Learning Outcomes: O-O Design

Demonstrate object-oriented design basics like domain models, class diagrams, and interaction (sequence and communication) diagrams.

- Reflecting on Plus/Delta
- Domain Model Refinement
- Conceptual Classes
- Design Studio

http://enterprisegeeks.com/blog/2009/07/

+/∂ Feedback: Lectures

Pace

0 – much too fast

8 – somewhat too fast

5 – Somewhat too slow

0 – much too slow

Working well

- □ Good class exercises (6)
- □ Numerous examples (4)
- □ Group activities (4)
- □ Class slides and material (3)
- □ Daily Quizzes (3)
- □ Teaching style, good explanations (2)

Improvements

- ● On Target (5)
- ● Not 1st hour (1) ☺
- ● More concrete examples (1)
- ● More example not from text (1)
- ● More guidance during examples and more feedback as we go along (1)
- ● More time for long quiz questions (1)
- ● Present examples from previous year’s milestones (1)
- ● Don’t read comics ☹ (1)

+/∂ Feedback: Quizzes

Quizzes

7 – Very helpful

5 – somewhat helpful

1 – somewhat unhelpful

0 – Very unhelpful

Working well

- □ Focuses lecture for me (6)
- □ Good study guide (4)
- □ Questions work well (3)
- □ Modulated with slides (2)
- □ Good length/depth (1)
- □ Everything (1)

Improvements

- Quizzes are fine (6)
- Sometimes poorly worded (1)
- Some questions hard to complete in time provided (1)
- Cut back on long answers (1)
- Be more specific (1)
- More design on paper exercises (1)

+/∂ Feedback: Reading and Homework

Reading

- 0 – all of it
- 2 – most of it
- 7 – little of it
- 4 – none of it

Homework Difficulty

- 0 – much too difficult
- 10 – a bit too difficult
- 3 – a bit too easy
- 0 – much too easy

+/∂ Feedback: Homework Helpfulness

Homework Helpfulness

0 – very helpful

9 – somewhat helpful

4 – somewhat unhelpful

0 – very unhelpful

Working well

- □ Corresponds to Milestones (7)
- □ Re-enforces class material (5)
- □ Frequency about right (2)
- □ Good balance of work (1)
- □ Relevant and encourages exploration of concepts (1)

Improvements

- ● Working well (2)
- ● Eliminate homework and put exercises in milestones (2)
- ● Assign/return homework earlier for more time on MS (2)
- ● More specific instructions (1)
- ● Ditch BBVS – tired of it (1)
- ● Need better UML tool (1)
- ● Homework seemed long (1)
- ● Not time to get good feedback before milestones (1)
- ● Last 2 homeworks bend patterns to apply to limited # of system operations (1)

+/∂ Feedback: Workload

- Workload

  3 – much higher than average

  9 – somewhat higher than average

  1 – somewhat lower than average

  0 – much lower than average

- General Comments
  - □ Good class, just about right (4)
  - □ Move from 1st hour ☹ (1)
  - □ Consider removing iteratively introducing patterns– tell us how to do all patterns then let us design (1)
  - □ Encouragement (10), Neutral (3) Discouragement (0) ☺

Summary of +/∂ Actions

- Lighten homework to focus on Milestones for rest of course
- Better word quiz questions and modulate time
- Position homeworks and milestone assignments better in the future

Starting with Use Cases, you produce analysis models, design models, test cases, and ultimately code.
Why would you go back and refined the domain model?

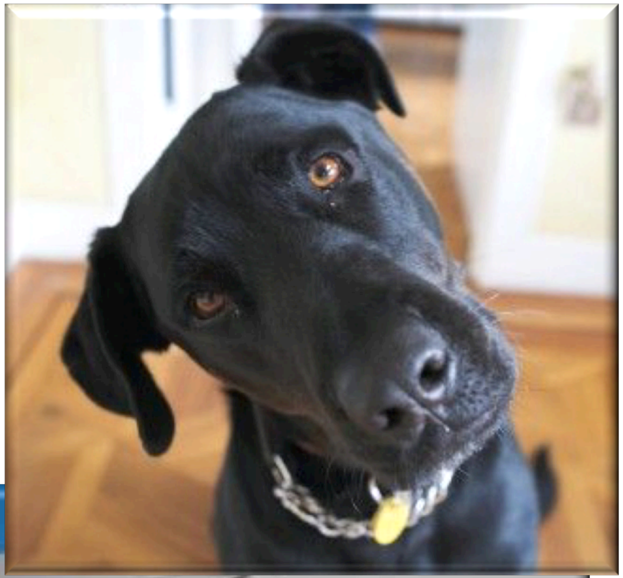
- Think for 15 seconds…
- Turn to a neighbor and discuss it for a minute

Recall: Strategies for Finding Conceptual Classes

1. Reuse or modify existing models

2. Identify noun phrases; linguistic analysis

3. Use a category list



Conceptual Category List on NextGen POS, Iteration 3

Category	Examples
Physical or tangible objects	CreditCard, Check
Transactions	CashPayment, CreditPayment, CheckPayment
Other systems external to ours	CreditAuthorizationService, CheckAuthorizationService
Organizations	CreditAuthorizationService, CheckAuthorizationService
Records of finance, work, contracts, legal matters	AccountsReceivable

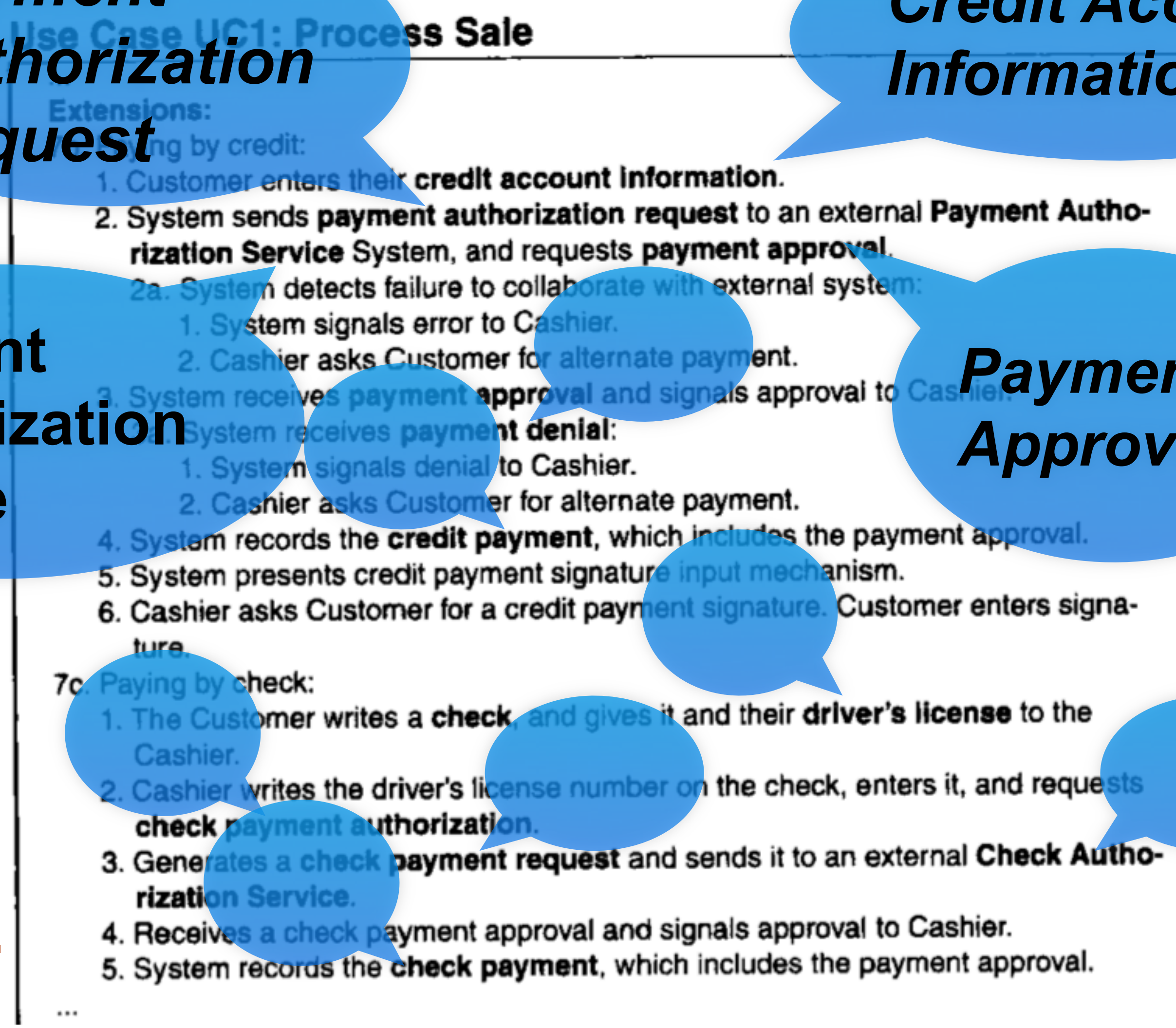
Noun Phrase Identification on NextGen POS, Iteration 3

Payment Authorization Request

Credit Account Information

Payment Authorization Service

Payment Approval



Use Case UC1: Process Sale

Extensions:

Paying by credit:

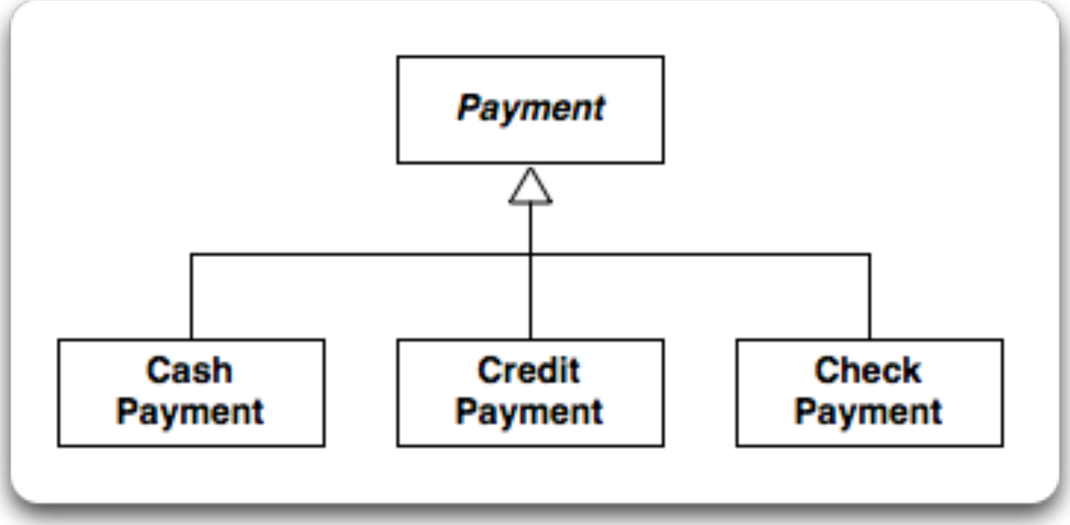
1. Customer enters their **credit account information.**
2. System sends **payment authorization request** to an external **Payment Authorization Service** System, and requests **payment approval.**
   - 2a. System detects failure to collaborate with external system:
     1. System signals error to Cashier.
     2. Cashier asks Customer for alternate payment.
3. System receives **payment approval** and signals approval to Cashier.
   - 3a. System receives **payment denial**:
     1. System signals denial to Cashier.
     2. Cashier asks Customer for alternate payment.
4. System records the **credit payment**, which includes the payment approval.
5. System presents credit payment signature input mechanism.
6. Cashier asks Customer for a credit payment signature. Customer enters signature.

7c. Paying by check:

1. The Customer writes a **check**, and gives it and their **driver's license** to the Cashier.
2. Cashier writes the driver's license number on the check, enters it, and requests **check payment authorization.**
3. Generates a check **payment request** and sends it to an external **Check Authorization Service.**
4. Receives a check payment approval and signals approval to Cashier.
5. System records the **check payment**, which includes the payment approval.

...

Generalization-Specialization Class Hierarchy



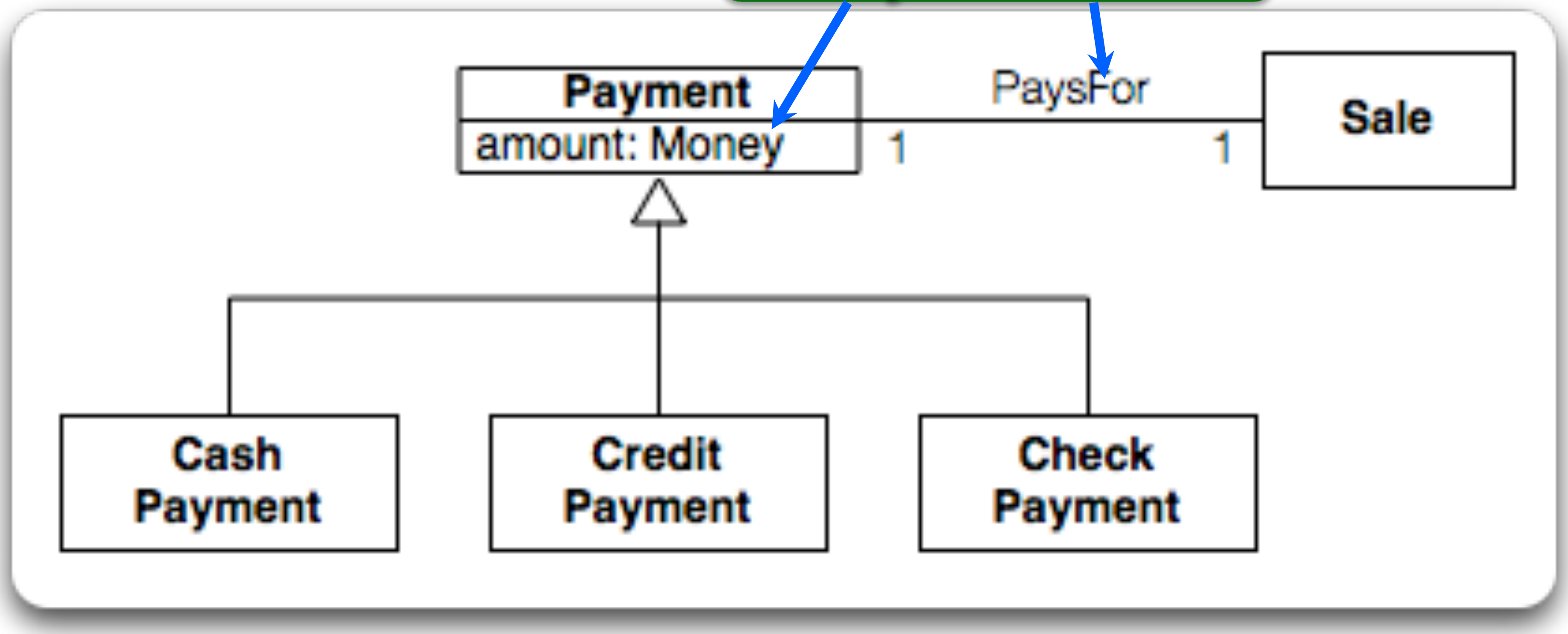
Why? We can understand concepts in more general terms.

Payment gives our brains less to deal with.

- Conceptual classes, not software classes
- *Generalization*: finding commonalities among concepts
  - □ *Superclass*: general concept
  - □ *Subclass*: specialized concept

Q2

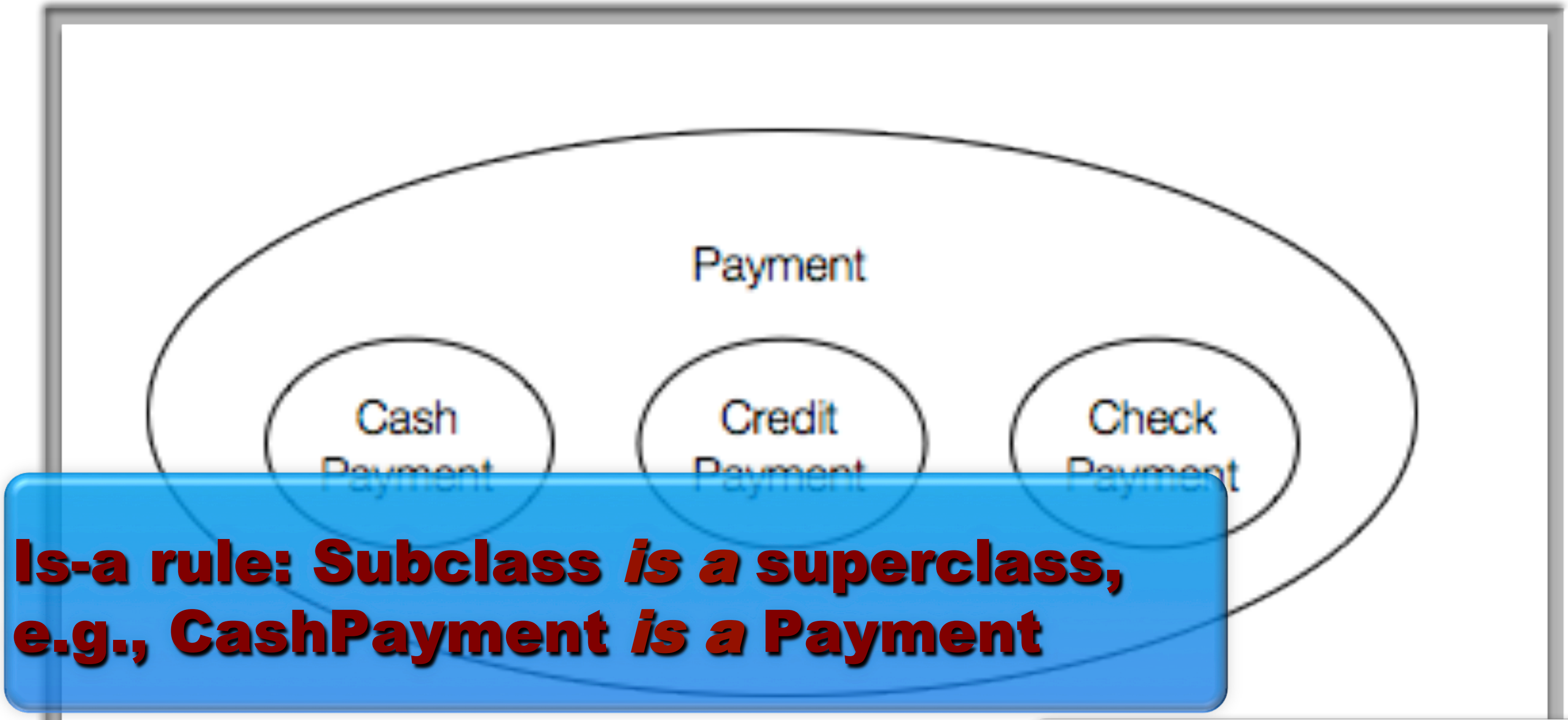
Generalization



Common features of Payments

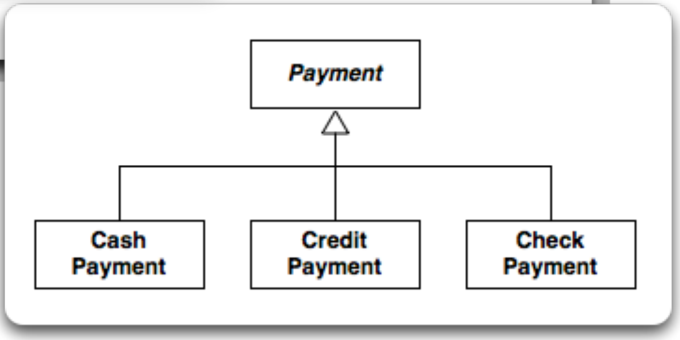
Q2

Genaralizations and Sets



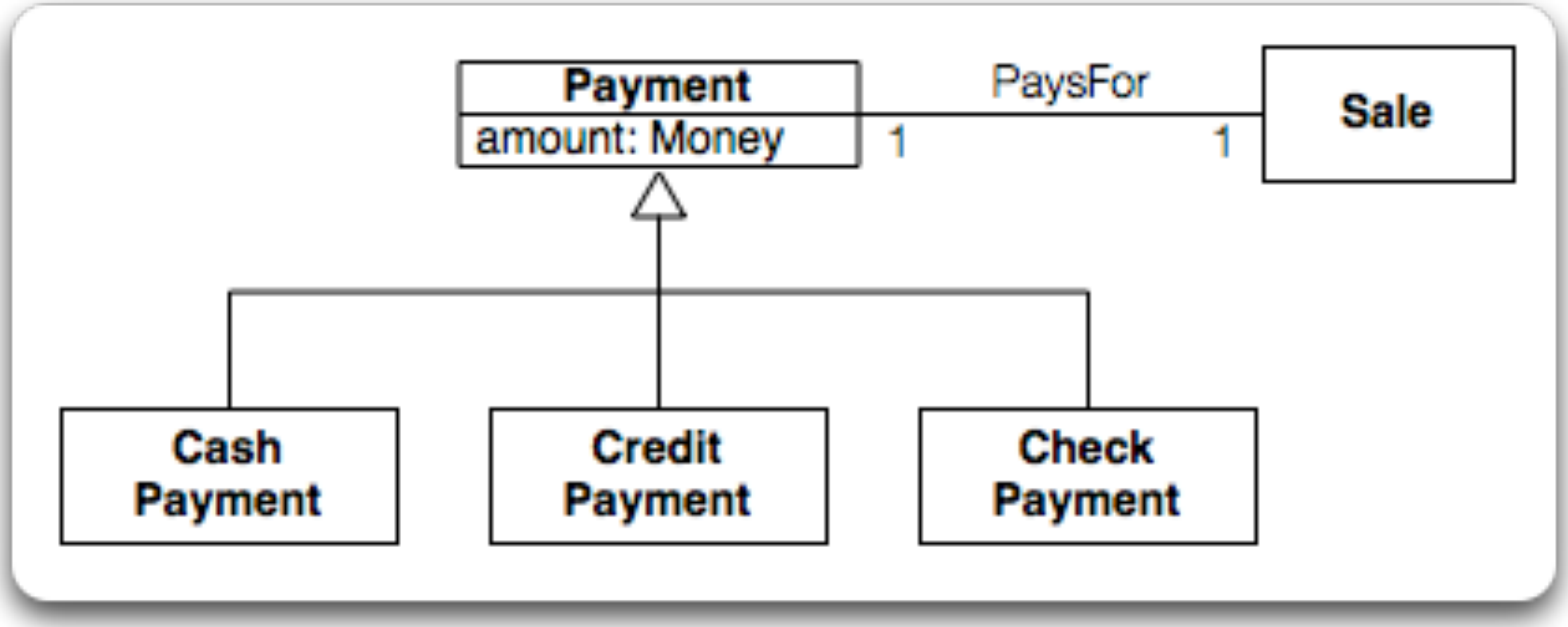
Is-a rule: Subclass *is a* superclass, e.g., CashPayment *is a* Payment

All members of a conceptual subclass set are members of their superclass set



Q3

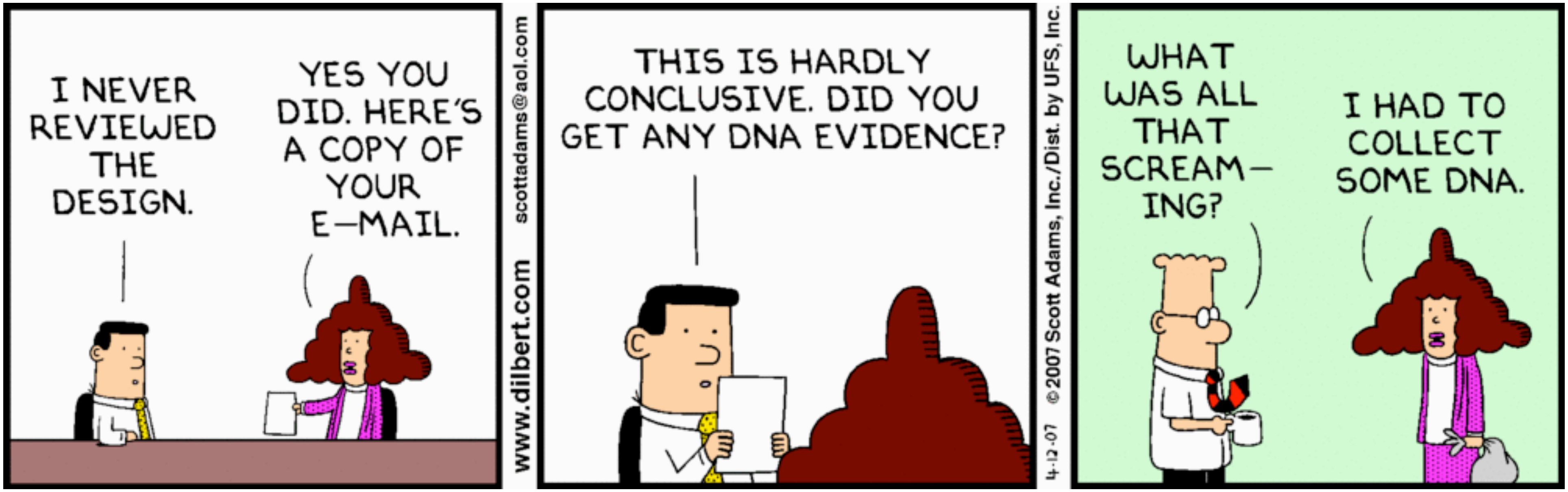
Subclass Conformance—The 100% Rule



The subclass must conform to all of the superclass's attributes and associations.

Q4

Sometimes Reviews are Vital…



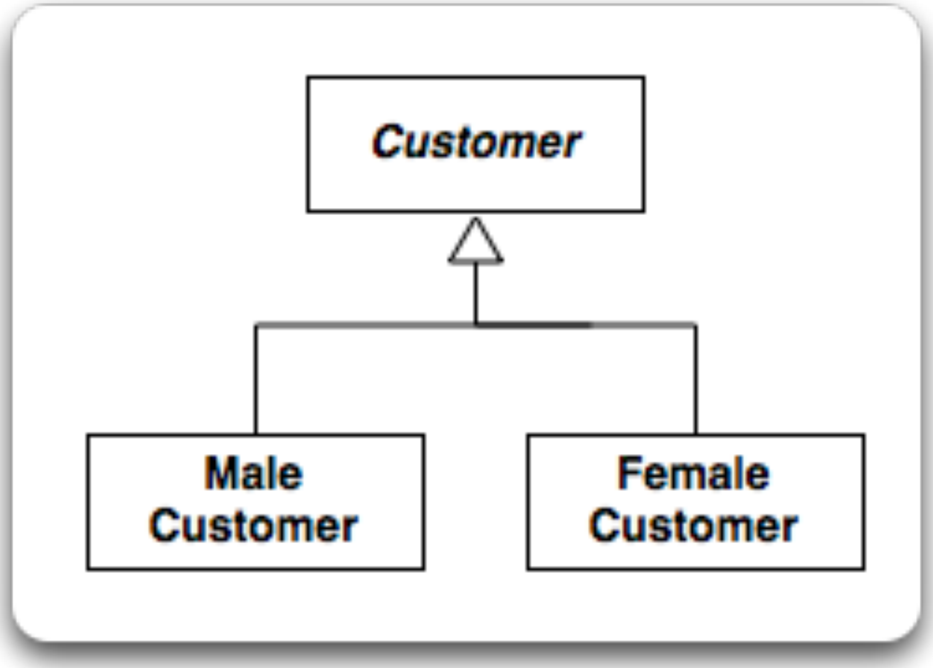
I NEVER REVIEWED THE DESIGN.
YES YOU DID. HERE'S A COPY OF YOUR E-MAIL.
www.dilbert.com scottadams@aol.com
THIS IS HARDLY CONCLUSIVE. DID YOU GET ANY DNA EVIDENCE?
4-12-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.
WHAT WAS ALL THAT SCREAM-ING?
I HAD TO COLLECT SOME DNA.

Whatcha' got in the bag… oh, that… it just some DNA (Design Never Assured) evidence.

When should we define a Conceptual Subclass? 1/2

Does this make sense…

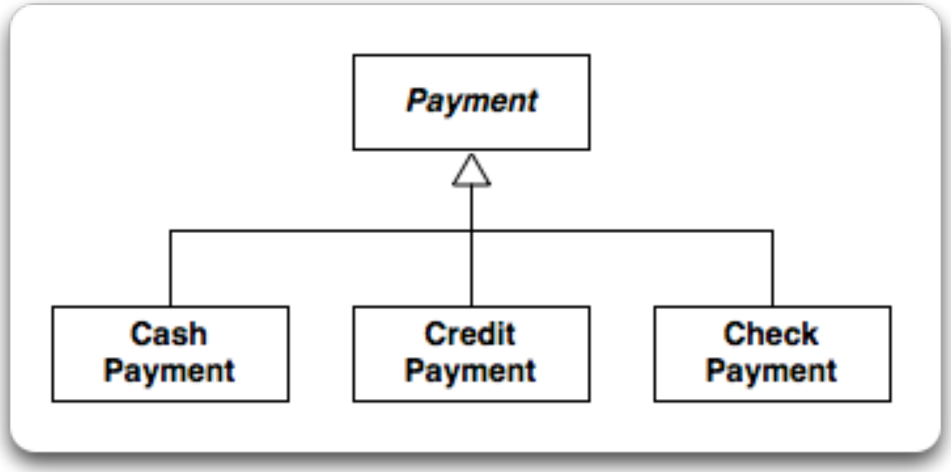
- □ for NextGen POS?
- □ for other domains?



When should we define a Conceptual Subclass? 2/2

When the subclass…

1. has additional attributes
2. has additional associations
3. is operated on or handled differently
4. represents an animated thing that behaves differently



Which of these apply here?

Q5

When should we group classes and extract a superclass?

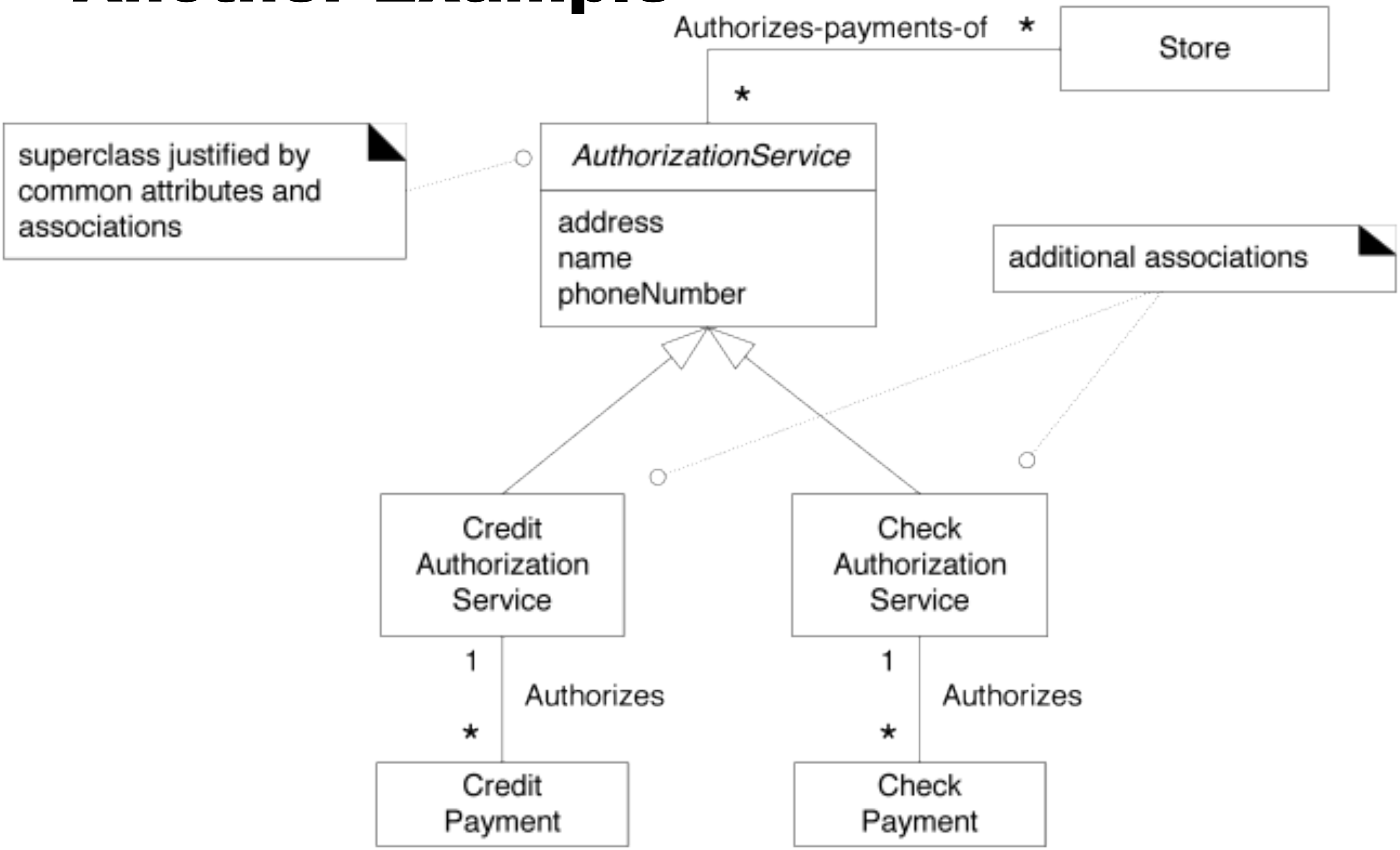
Create a superclass when:

1. Potential *subclasses represent variations* of a similar concept
   (e.g., Video, Game → RentableItem)

2. Subclasses will conform to *100% and is-a rules*

3. There are *common attributes or associations* that could be pulled into superclass

Another Example



Authorizes-payments-of *

Store

*

AuthorizationService

address
name
phoneNumber

superclass justified by common attributes and associations

additional associations

Credit Authorization Service

Check Authorization Service

1

Authorizes

*

1

Authorizes

*

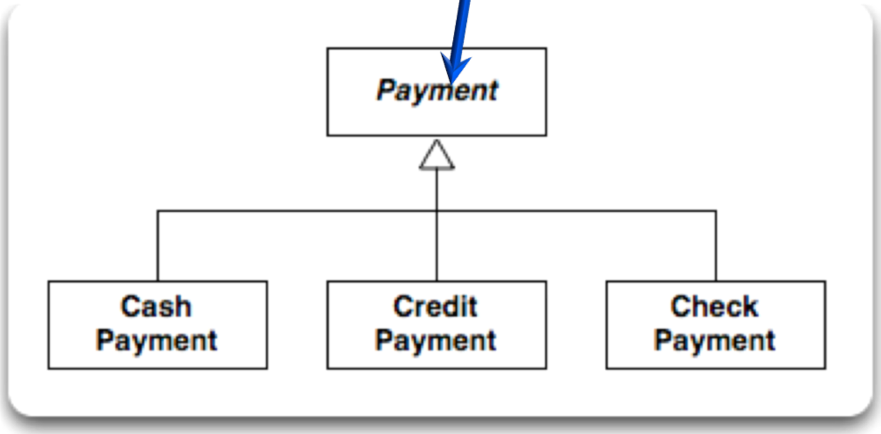
Credit Payment

Check Payment

Abstract Conceptual Classes

If every member of a class C must be a member of a subclass, then C is an *abstract conceptual class*

Italics (or {abstract} keyword) indicate abstract class



What are some examples of Abstract Conceptual Classes?

Q7

Thinking Ahead Exercise

- Break up into your teams
- Begin identifying what features you will plan to implement in your third (and final) iteration this term. Your goal, at minimum, should be to have a basic but workable system implemented. Think of this as a system that you can share with your client to demonstrate progress and that you and others can continue to build and refine in CSSE 375 this spring.

- Brainstorm a list of all the features remaining to be implemented for your system.

Design Studios

Objective is to share your design with others to communicate the approach or to leverage more eyes on a problem.

- Minute or so to set up…
- 5-6 minute discussion
- 1-2 minute answering questions

1. Team 2.5 – Academic Paper Cataloging System

Homework and Milestone Reminders

- Read Rest of Chapter 31
- Milestone 5 – Final Junior Project System and Design
  - □ Draft due by 11:59pm on Friday, February 11th, 2011
  - □ Final due by 11:59pm on Friday, February 18th, 2011
- Homework 6 – BBVS Design using GoF Patterns
  - □ Due by 11:59pm Tonight, Tuesday, February 1st, 2011
- Team 2.1 –Interactive Syllabus on Tuesday