Domain Model Refinements and more Iteration 3 Prep.

Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Plan for Today

- Summarize 2/3-term Course Evaluation
- Domain Model Refinements
- Iteration 3

+/∂ Feedback: Lectures

Pace

- 1 – much too fast
- 14 – somewhat too fast
- 2 – Somewhat too slow
- 0 – much too slow

Working well

- Group/interactive examples (3)
- In-class activities (5)
- Diagrams on board (3)
- Right pace and right slides (2)
- In-class examples from project
- Integr. of slides-quiz-exercises
- Helping me learn topics better
- Everything

Improvements

- Choose better examples—explain them more fully (2)
- More group/interactive examples (2)
- More specific reasons behind pattern & design
- Move quiz quest.# to page top
- Less text on slides
- Stop lecturing—teach us the info—tell us only what we need to know
- Keep on improving!

+/∂ Feedback: Quizzes

Quizzes

- 8 – Very helpful
- 8 – somewhat helpful
- 1 – somewhat unhelpful
- 0 – Very unhelpful

Working well

- Focuses lecture for me (9)
- Indicates high points (4)
- Integration with material (6)
- Questions work well (2)
- Sufficient time to answer
- Good study guide

Improvements

- Provide answers (3)
- Easier questions (2)

- More challenging questions (2)
- Less fill-in blank
- Make questions even shorter
- Less goofy (unicorn) questions
- Random questions too random
- Leave general definitions out

+/∂ Feedback: Reading and Homework

Reading

- 1 – all of it
- 2 – most of it
- 12 – little of it
- 2 – none of it

Homework Difficulty

- 3 – much too difficult
- 13 – a bit too difficult
- 1 – a bit too easy
- 0 – much too easy

+/∂ Feedback: Homework Helpfulness

Homework Helpfulness

- 3 – very helpful
- 8 – somewhat helpful
- 4 – somewhat unhelpful
- 2 – very unhelpful

Working well

- Helpful for Milestones (4)
- Re-enforces class material (7)
- Challenging to learn more (2)
- Good feedback (3)
- Good examples
- Instruction have improved
- Homeworks have been a great help in this class

Improvements

- Even specific instructions (4)
- Provide even more examples to clarify assignments (3)
- Simplify project/homework (2)
- Redo's on HW below 8/10
- Shorten them—too much time
- Do away with homework

+/∂ Feedback: Workload

- Workload
  - 2 – much higher than average
  - 12 – somewhat higher than average
  - 3 – somewhat lower than average
  - 0 – much lower than average

- General Comments
  - Things were just about right (3)
  - A lot of information, some irrelevant (2)
  - DeDS not good, but gone
  - Change slide template
  - Thank for not reading XKCD in class
  - Class has improved (2)

Summary of +/∂ Actions

- More active learning
- More examples done on board
- Focus more on Project
- Keep improving quizzes
- Continue to better clarify assignments

Domain Model Refinement

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

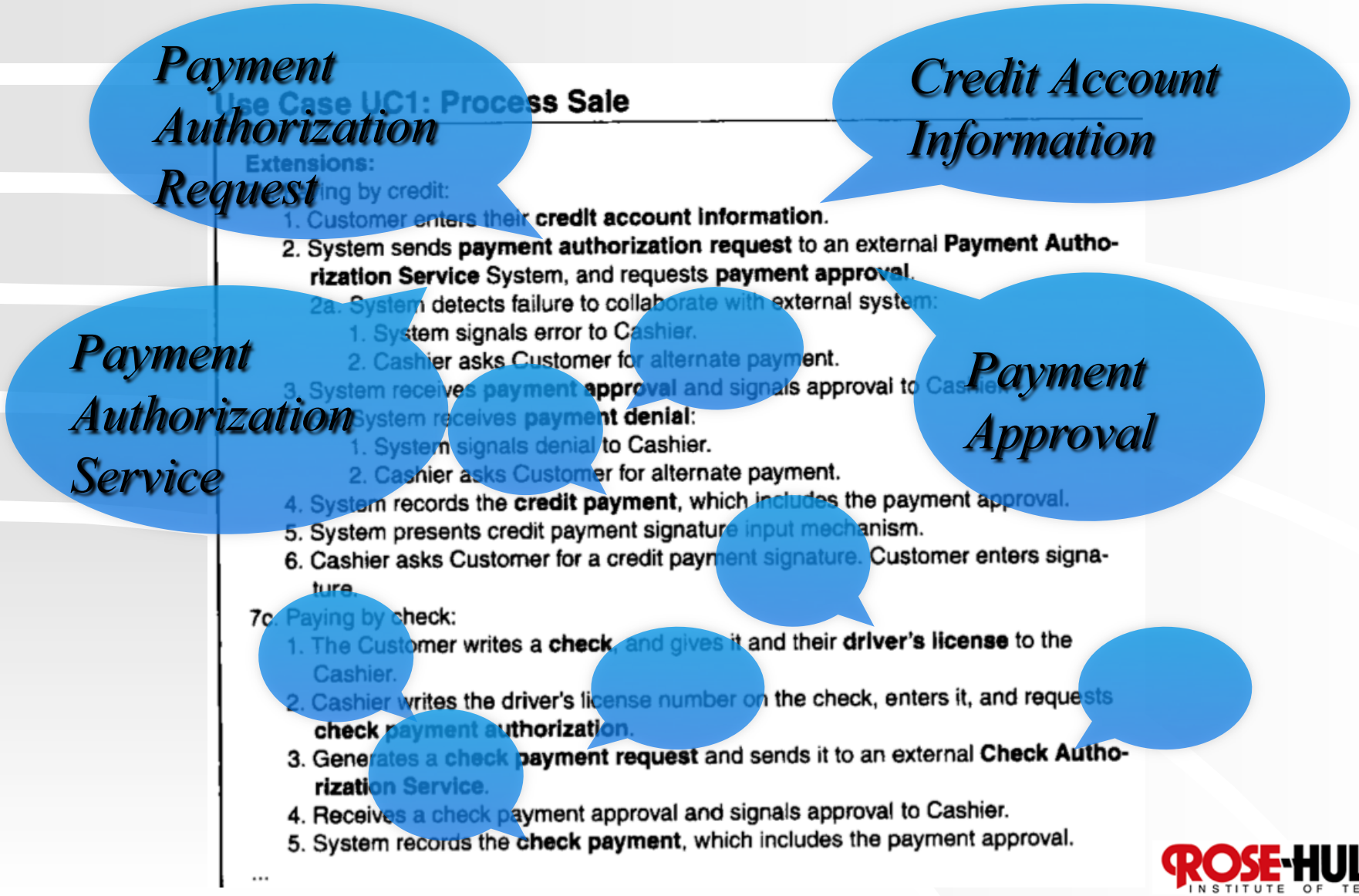
Recall: Techniques for identifying conceptual classes

- Conceptual category lists
- Noun phrase identification
- Existing domain models

Conceptual Category List on NextGen POS, Iteration 3

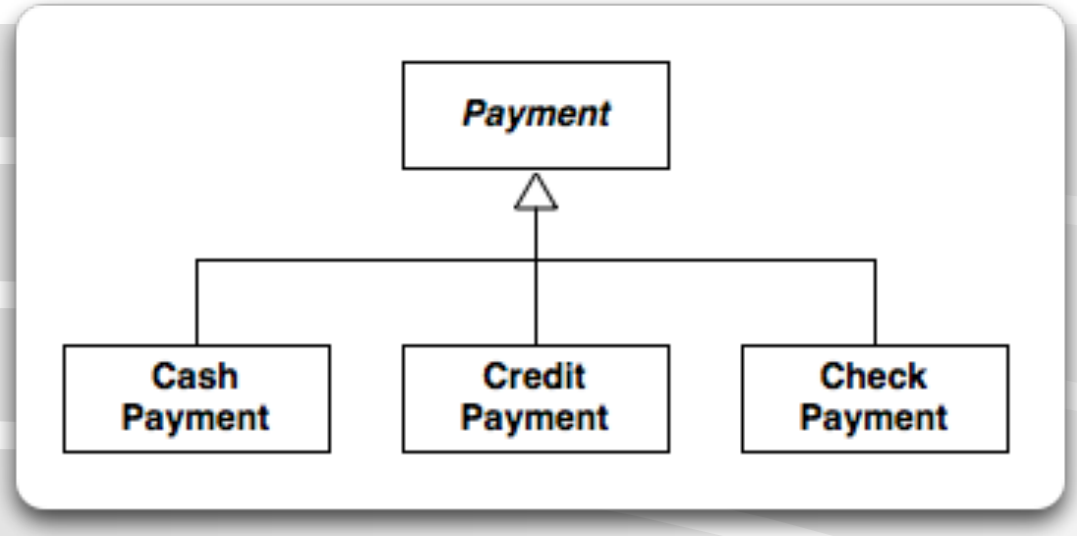
Category	Examples
physical or tangible objects	*CreditCard, Check*
transactions	*CashPayment, CreditPayment, CheckPayment*
other systems external to ours	*CreditAuthorizationService, CheckAuthorizationService*
organizations	*CreditAuthorizationService, CheckAuthorizationService*
records of finance, work, contracts, legal matters	*AccountsReceivable*

Noun Phrase Identification on NextGen POS, iteration 3



ROSE-HULMAN INSTITUTE OF TECHNOLOGY

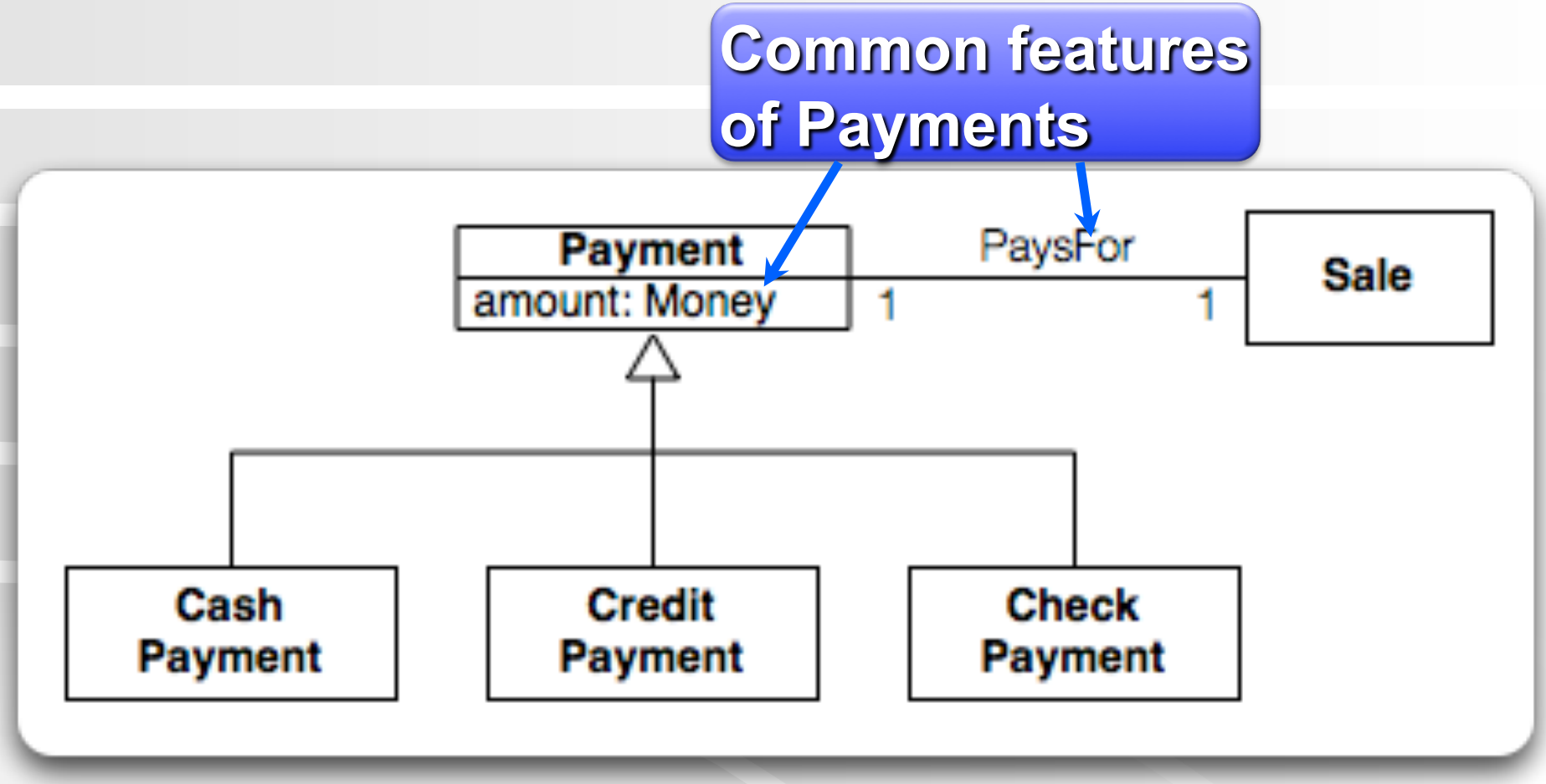
Generalization-Specialization Class Hierarchy



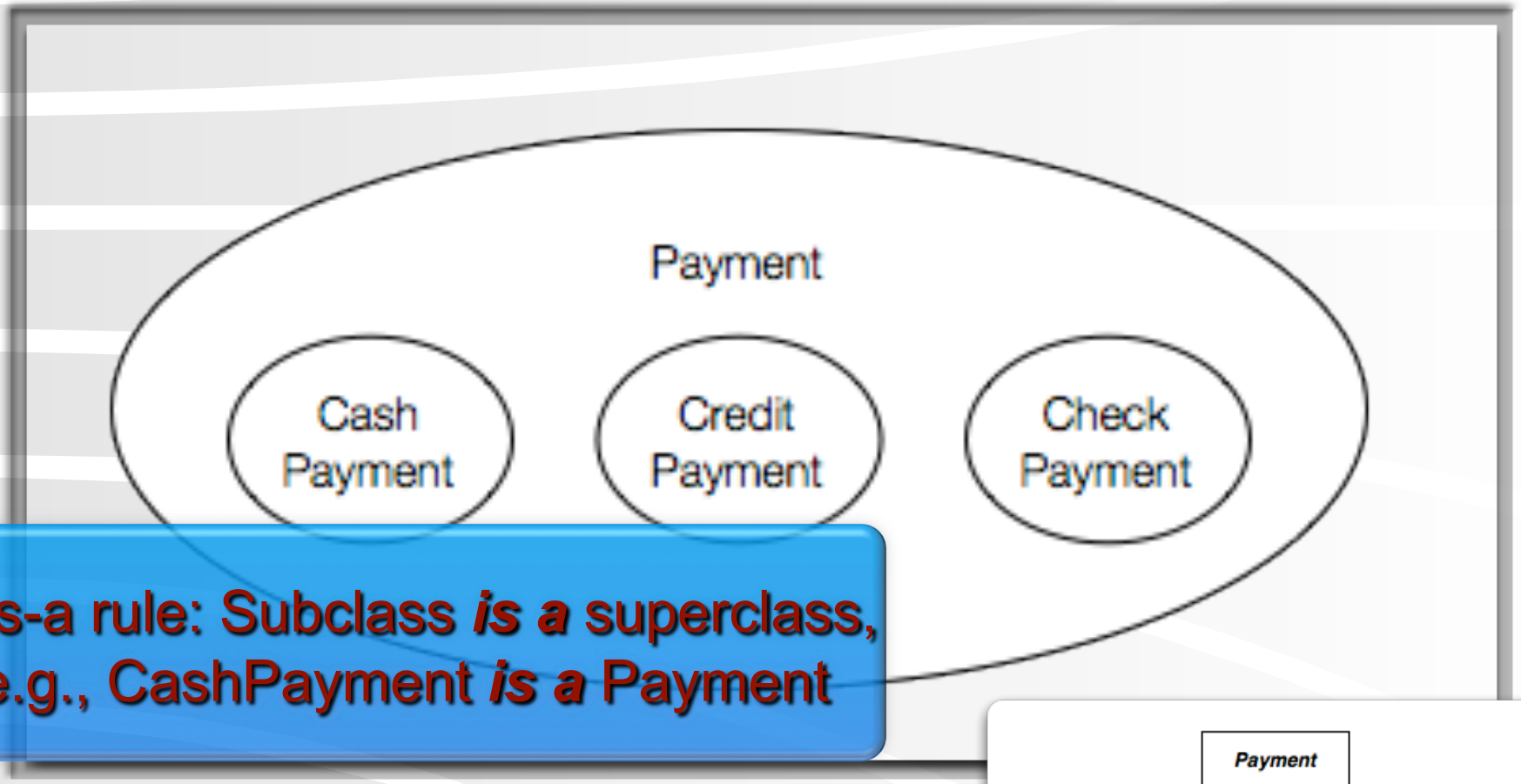
Why? Can understand concepts in more general terms. Payment gives our brains less to deal with.

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

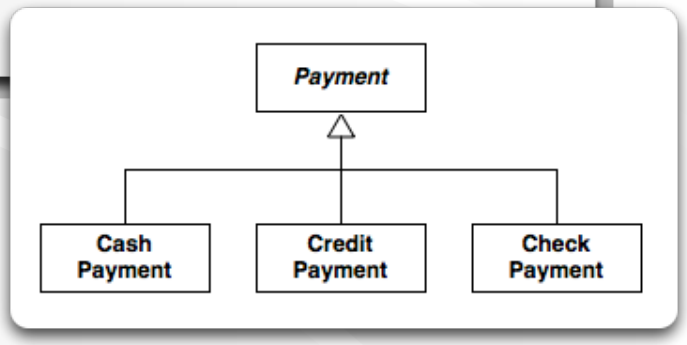
Generalization



Genaralizations and Sets

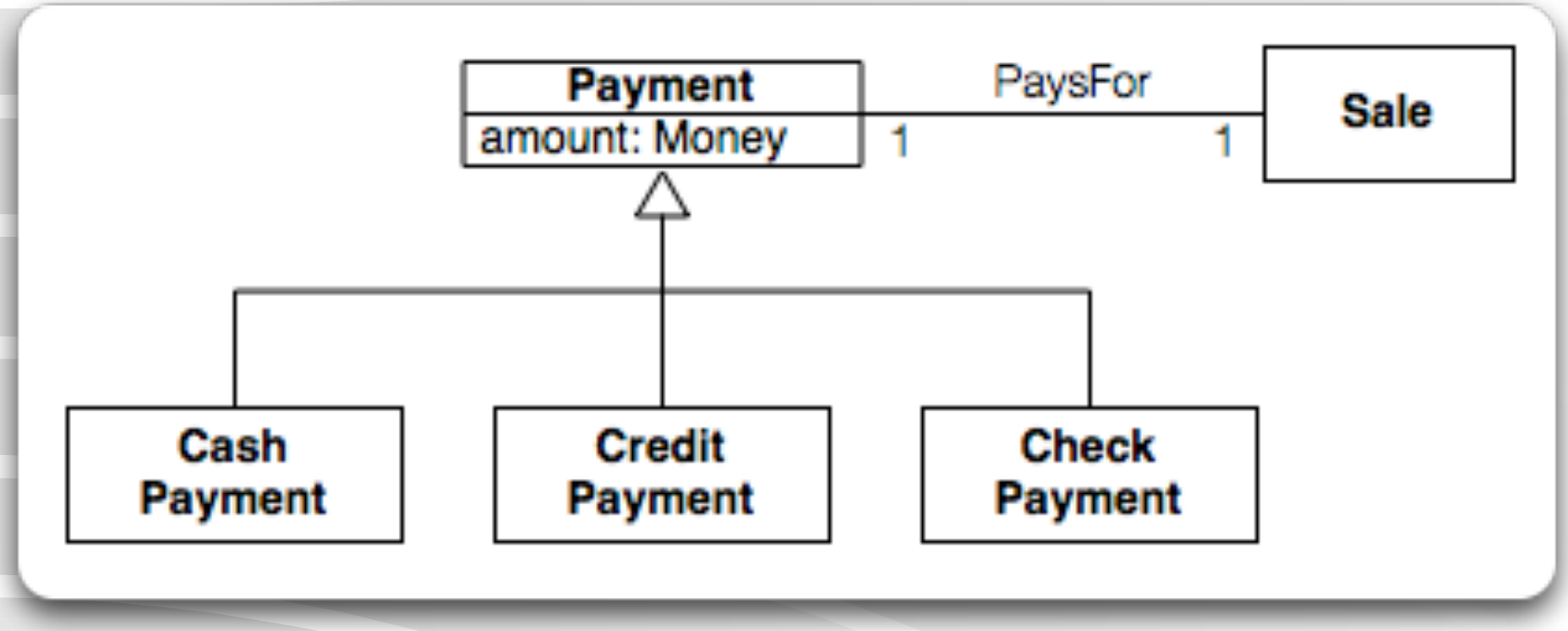


Is-a rule: Subclass ***is a*** superclass, e.g., CashPayment ***is a*** Payment



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

Subclass Conformance—The 100% Rule

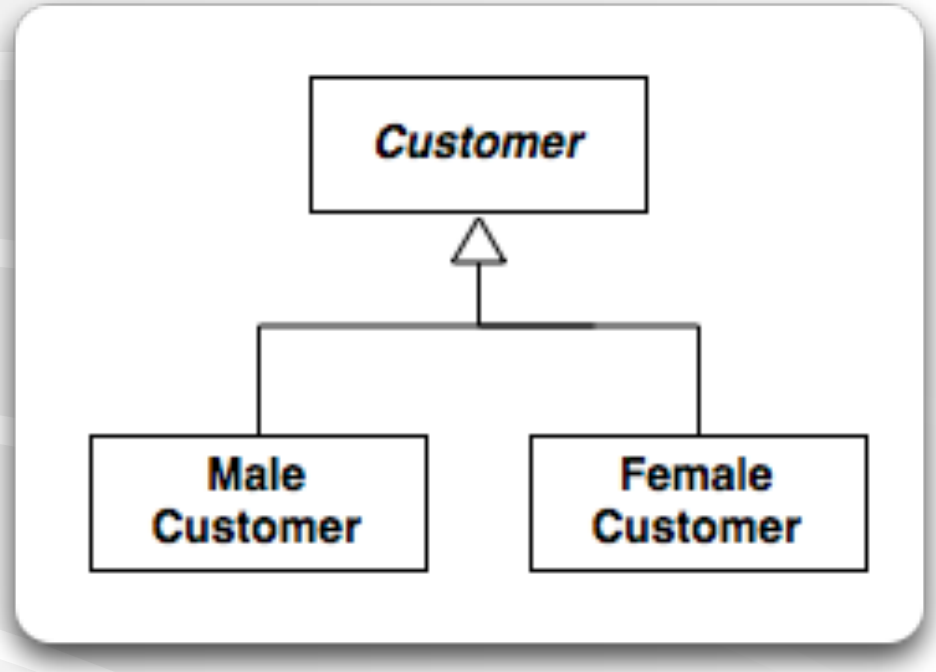


The subclass must conform to all of the superclass's *attributes* and *associations*.

When should we define a Conceptual Subclass?

Does this make sense…

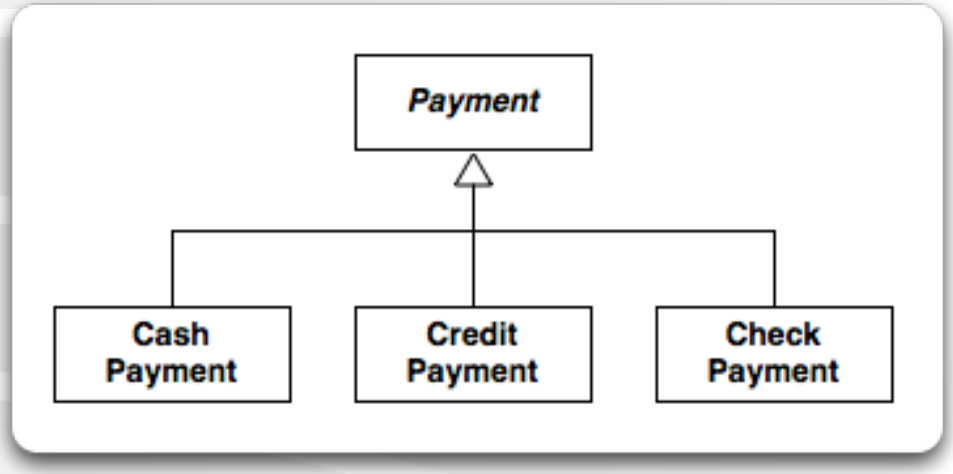
- for NextGen POS?
- for other domains?



When should we define a Conceptual Subclass? (continued)

When the subclass…

- has additional attributes
- has additional associations
- is operated on or handled differently
- represents an animated thing that behaves differently



Which of these apply here?

When should we group classes and extract a superclass?

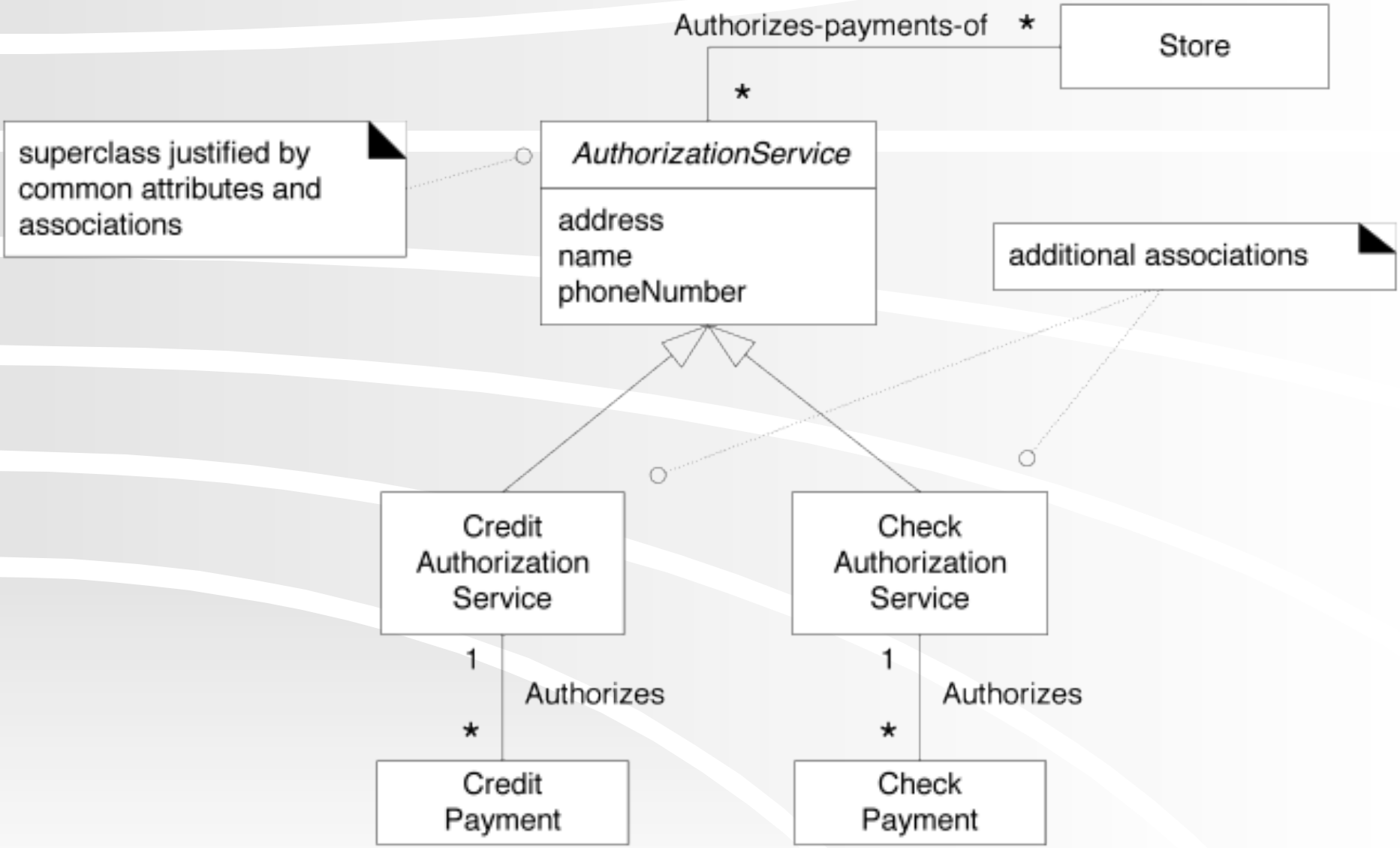
Create a superclass when:

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

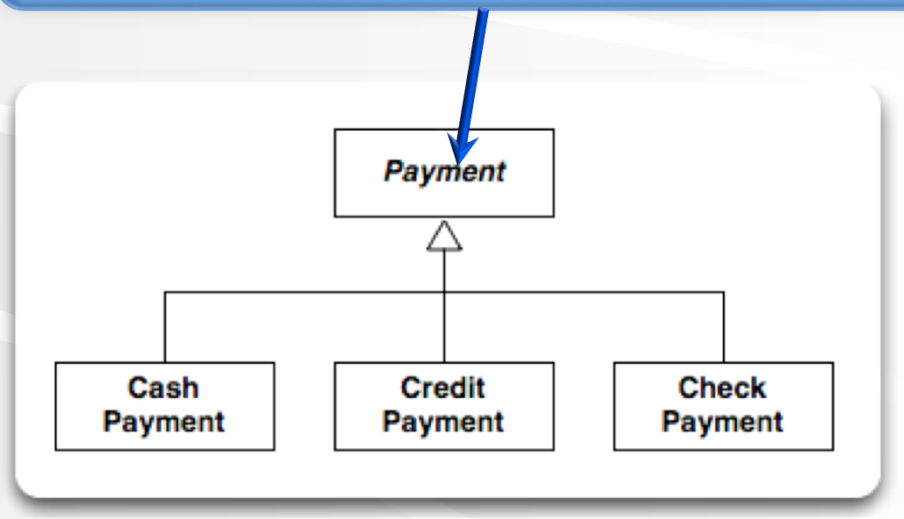
Another Example



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 does this mean in terms of our set idea?

Thinking Ahead…
Work in teams on Q4

Homework and Milestone Reminders

- Read Chapter 31 (rest)
- Homework 7– Gang of Four (GoF) Patterns on Video Store Design
  - Due by 5:00pm Tuesday, February 2nd, 2010