-

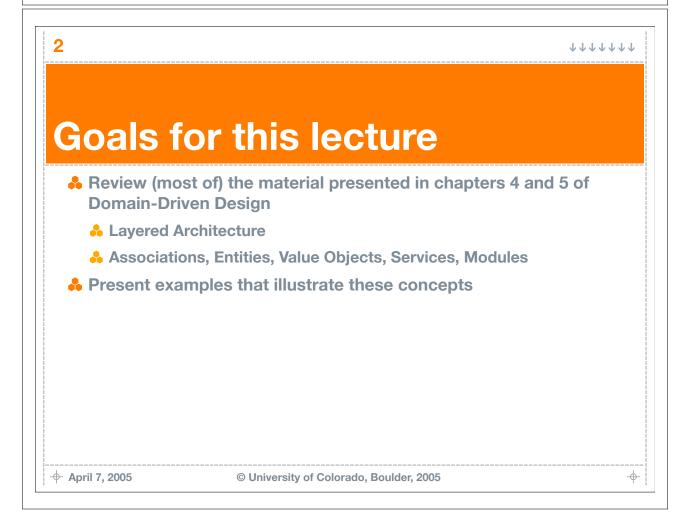
## Lecture 24: Domain-Driven Design (Part 2)

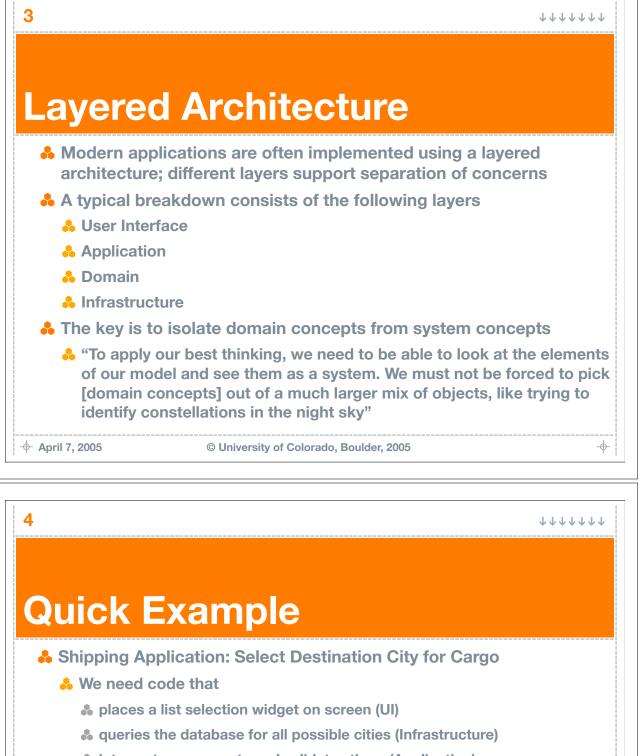
Kenneth M. Anderson

**Object-Oriented Analysis and Design** 

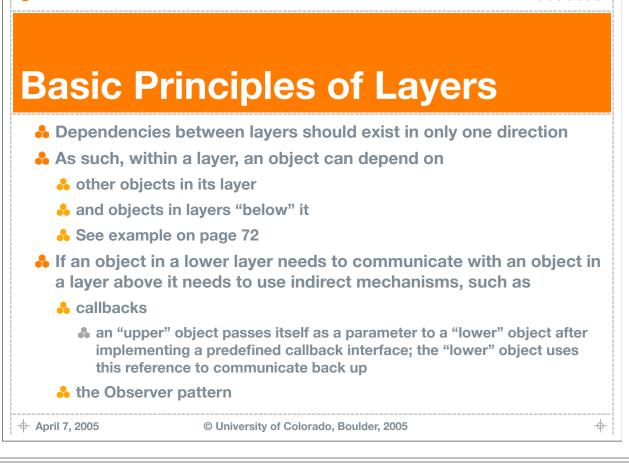
CSCI 4448/6448 - Spring Semester, 2005

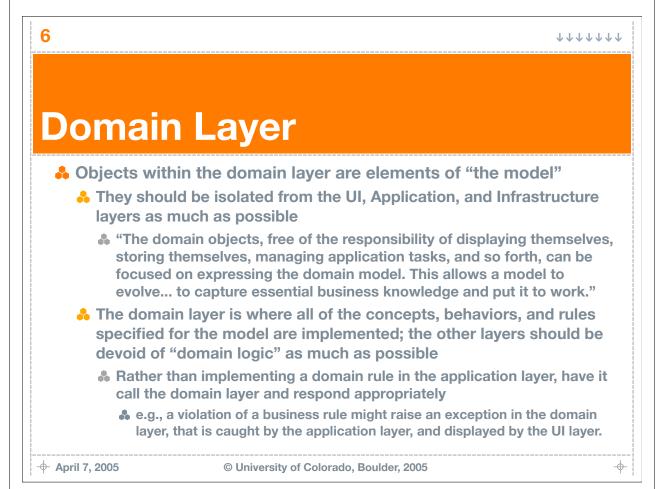
-\$

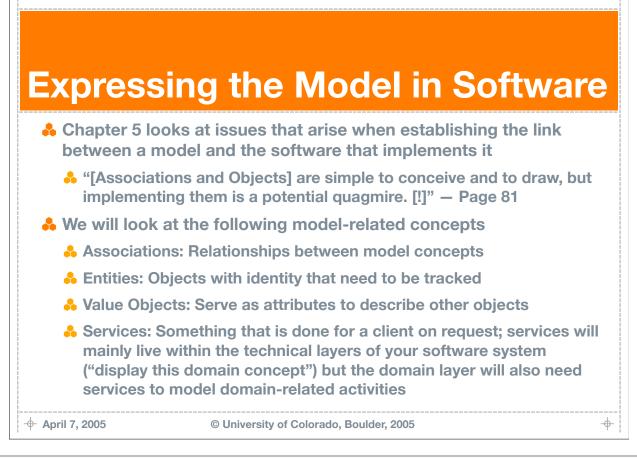


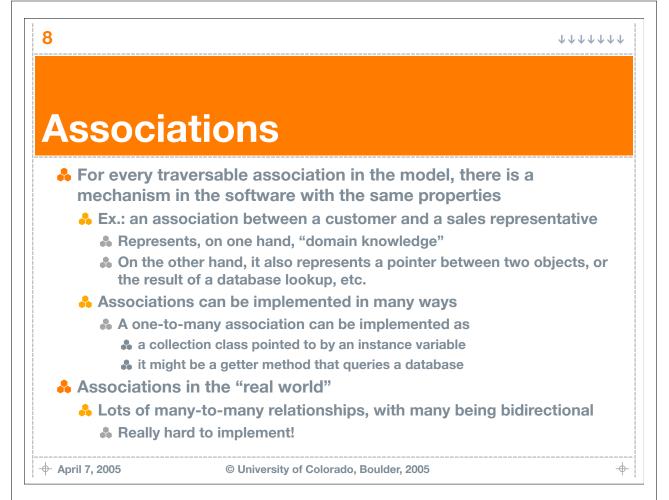


- interprets user events and validates them (Application)
- associates the selected city with the cargo (Domain)
- commits change to database (Infrastructure)
- The domain layer constitutes only a small portion of the entire software system, yet its importance is disproportionate to its size
  - (for reasons covered in lecture 23)



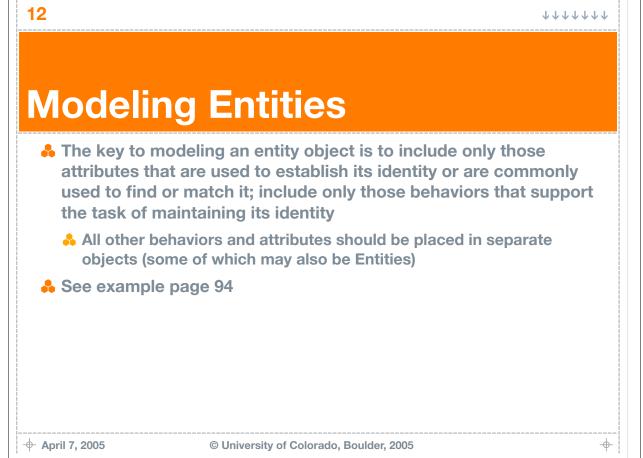






	$\psi \psi \psi \psi \psi \psi$
Dealir	ng with Associations
	e three techniques for making associations managable
	e a traversal direction
	qualifier, effectively eliminating or reducing multiplicity
	ate nonessential associations (as dictated by the problem you ing to solve)
🔒 See exar	nples of the first two techniques on pages 84-88
→ April 7, 2005	© University of Colorado, Boulder, 2005
10	$\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$
10	$\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$
10	$\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$
Entitie	<b>BS</b> Djects are not defined primarily by their attributes. They
Entitie	<b>OS</b> Djects are not defined primarily by their attributes. They at a thread of identity that runs through time and often
Entitie Some ob represent across d	<b>Opects are not defined primarily by their attributes. They at thread of identity that runs through time and often istinct representations</b>
Entitie Some ob represent across d Consid Cus	es ojects are not defined primarily by their attributes. They at a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history
Entitie Some ob represent across d Consid Cus if	ess ojects are not defined primarily by their attributes. They at a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history its good, "status" will accrue; if its bad, the customer's information may be
Entitie Some ob represent across d Consid Cus in tu The	Dijects are not defined primarily by their attributes. They that a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history its good, "status" will accrue; if its bad, the customer's information may be ransferred to a bill-collection agency same customer may be in the contact management software used by
<ul> <li>Some ob represent across d</li> <li>Consid</li> <li>Consid</li> <li>Cus</li> <li>if</li> <li>tr</li> <li>The your</li> </ul>	Dijects are not defined primarily by their attributes. They to a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history its good, "status" will accrue; if its bad, the customer's information may be ransferred to a bill-collection agency
Entitie Some ob represent across d Consid Cus if tr The your If but	eS bjects are not defined primarily by their attributes. They at a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history its good, "status" will accrue; if its bad, the customer's information may be ransferred to a bill-collection agency same customer may be in the contact management software used by r company's sales force customer may be "squashed flat" for storage in a database usiness stops, the customer may be placed in an archive
Entitie Some ob represent across d Consid Cus if tr The your If bu Each a	essesting the second state of the customer may be implemented in multiple ways,
Entitie Some ob represent across d Consid Consid Cus if tr The your The Lif bu Each a using d The	Dijects are not defined primarily by their attributes. They at a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history its good, "status" will accrue; if its bad, the customer's information may be cansferred to a bill-collection agency same customer may be in the contact management software used by r company's sales force customer may be "squashed flat" for storage in a database isiness stops, the customer may be placed in an archive respect of the customer may be implemented in multiple ways, different representations and/or programming languages y all represent the SAME customer however, and some means must
Entitie Some ob represent across d Consid Cus if tr The Joint The If bu Each a using d The	Dijects are not defined primarily by their attributes. They at a thread of identity that runs through time and often istinct representations der the notion of "customer" in a typical business system tomer may have a payment history its good, "status" will accrue; if its bad, the customer's information may be ransferred to a bill-collection agency same customer may be in the contact management software used by r company's sales force customer may be "squashed flat" for storage in a database usiness stops, the customer may be placed in an archive aspect of the customer may be implemented in multiple ways, different representations and/or programming languages

Entitie	s, continued
Å An object	defined primarily by its identity is called an Entity
🔥 They ha	ve life cycles that can radically change their form and content
🔥 Their ide	entities must be defined so that they can be effectively tracked
progra	otion of identity is DIFFERENT from the identity mechanisms of amming languages; i.e., it is different from "a == b" and uals(b)" that OO languages provide
🔒 Example	ž
same	eposits of the same amount made to the same bank account on th day are NOT identical; they are two separate entity objects in the ng domain
	objects representing the amounts ARE identical, however, and are most ely Value Objects (discussed next)
April 7, 2005	© University of Colorado, Boulder, 2005



## **Designing the Identity Operation**

Such that two instances of the same entity can be distinguished from one another, even if they both contain the same descriptive attributes (like our bank deposits from slide 11)

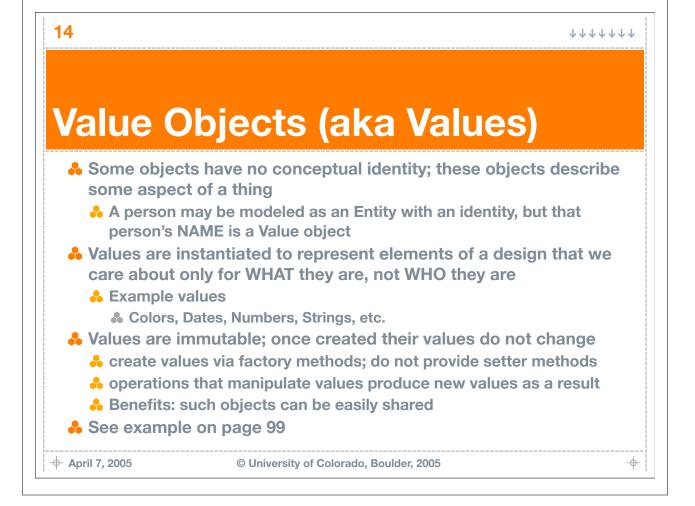
## Identity is often operationally established by

- sensuring that a single attribute has a unique id
- or ensuring that some combination of attribute values always produce a unique key
- Often the means for establishing identity require a careful study of the domain; what is it that humans do to distinguish the real-world counterparts of the entity object?

```
- April 7. 2005
```

© University of Colorado, Boulder, 2005

+



15	$\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$
Services	
includes operation object; Rather the	s, the clearest and most pragmatic design ons that do not conceptually belong to a single an force the issue, we can follow the natural roblem space and include SERVICES explicitly in
	f you give up too often on finding a home for an vill end up with a procedural programming solution
fit that object's	nd, if you force an operation into an object that doesn't definition, you weaken that object's cohesion and fficult to understand
 - April 7, 2005	© University of Colorado, Boulder, 2005 -+
φ April 7, 2005	
16	$\psi \psi \psi \psi \psi \psi$
Services	continued
	Continucu
	peration offered as an interface that stands alone in efined purely in terms of what it can do for a client
Services tend to nouns)	be named for what they can do (verbs rather than

- A good service has three characteristics
  - The operation relates to a domain concept that is not a natural part of an entity or value object
  - The interface is defined in terms of other elements of the domain model
  - The operation is stateless (does not maintain or update its own internal state in response to being invoked)

17		$\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$
Module	S	
Modules are on a model	groupings of model elements; They prov	vide two views
🔒 one view pi	ovides details within an individual module	
the second modules	view provides information about relationsh	ips between
🔒 We shoot for	modules with high cohesion and low co	upling
high cohes purpose	on: elements within a module all support the	ne same
	g: elements within a module primarily refer ; references to objects outside the module	
∲- April 7, 2005	© University of Colorado, Boulder, 2005	
18		$\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$

## What's Next

- Review the material of Chapter 6 of Domain-Driven Design
  - Life Cycles of Domain Objects
    - **Aggregates**
    - Factories
    - Repositories