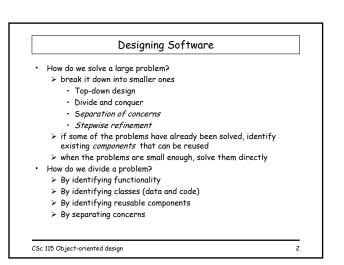
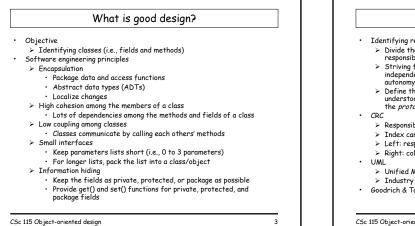
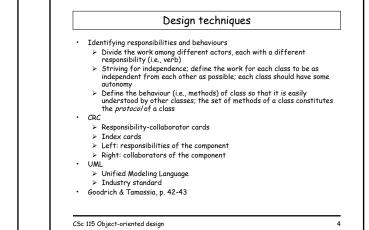
## Object-oriented Design

Reading Assignment Chapters 1-2







	RC cards	
Class Name:		-
Superclasses:		
Subclasses:		
Responsibilities:	Collaborators	
		_
	1	_

Circ		
	le or colour all the important-looking <i>nouns</i>	
· · ·	they become candidates for <i>classes</i> and <i>fields</i>	
➢ Und	erline or colour all the important-looking verbs	
• • •	they become candidates for <i>methods</i>	
⊁ Put	them together into coherent <i>classes</i> write a short description for each class	
> Tde	ntify <i>relationships</i> among the classes	
≻ If c	lasses share characteristics, extract them into erclasses (i.e., <i>generalization)</i>	
	descriptions or relationships use vocabulary not ye ed, iterate through the process.	t

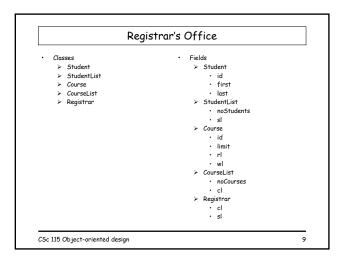
## Design Quiz

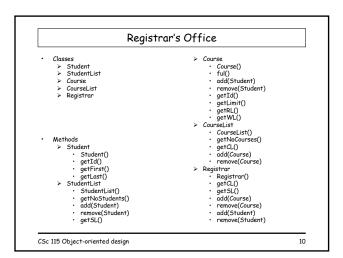
The registran's office is upgrading its course registration system to be written in Java. Students can enroll in courses being offered, but each course has an enrollment limit: once this is reached, all further registrants are placed on a waiting list. Students can also withdraw from a course they are enrolled in.

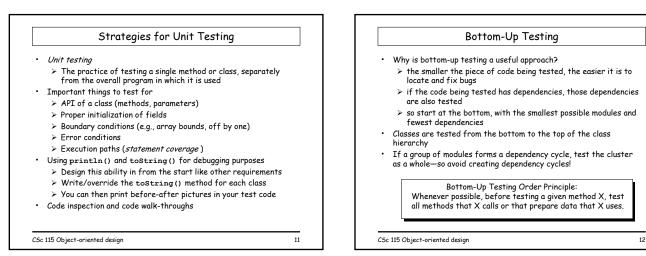
CSc 115 Object-oriented design

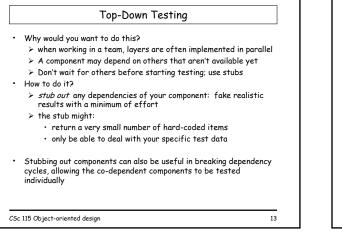
	Design Quiz
5	ce is upgrading its course registration system to
	Students can enroll in courses being offered, but
	enrollment limit: once this is reached, all further
5 .	ed on a waiting list. Students can also withdraw
from a course they	are enrolled in.
Nouns - candidate d	classes, object, fields
Verb - candidate me	thods
Sc 115 Object-oriented d	esian 8

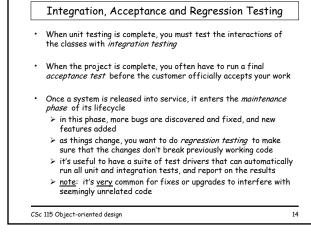
7











<ul> <li>An assertion is the statement of a fact that should be true at a given point in the execution of a program</li> <li>assertions can be written as comments, to document the code:         count;         // assert: count &gt;= 0</li> <li>they can be written as code, to verify assumptions at runtime:         count;         if (!(count &gt;= 0)) throw new AssertionFailure();</li> </ul>	<ul> <li>The class:</li> <li>It has attributes tha         <ul> <li>instance variables</li> <li>It allows some access</li> <li>through public me</li> <li>It allows internal act</li> <li>through private n</li> </ul> </li> </ul>
	ni odgit privare i
<ul> <li>An assertion at the beginning of a method is called a <i>precondition</i></li> <li>it will often validate the method's arguments</li> <li>An assertion at the end of a method is called a <i>postcondition</i></li> <li>it will often validate the method's work and/or result</li> <li>When assertions are stated using a formal logical language, it's sometimes possible to prove a program's correctness; this is called <i>verification</i></li> </ul>	<ul> <li>An <u>object</u> can be anythin</li> <li>A computer interpret</li> <li>car, cartoon draw</li> <li>A collection of data</li> <li>A single data item</li> </ul>

