Object Oriented Design

COP3330 – Fall 2006
University of Central Florida

H. Schwartz

Principles

Problem Solving Steps:

(from book)

1. Analyze the problem in order to break it up into simpler components.
2. Design a solution.
3. Implement the solution. (write code)
4. Test and fix any problems. (debug)

Principles

Encapsulation:

Placing characteristics and routines within

-some hidden from the outside to reduce complexity

-some that are designed to be seen from outside

Abstraction:

Determining characteristics and routines which are pertinent yet not restrictive in order to produce a more simplified view.

Principles

Object Oriented World:

In order to handle complexity, we perceive the world as objects with properties.

What are some examples?

Objects and Classes

`Class:`

Describes characteristics and variables to be filled in by instances and routines to act on these characteristics

`Object:`

An instance of a class where characteristics are filled in and routines can actually be run

Objects and Classes

`Classes can be abstracted:`

`abstract (general)` `restrictive`

`many possibilities` `few possibilities`

Classes may be designed to **inherit** from classes which are more abstract.

Objects and Classes

`Constructor:`

Creates an instance / object from a class.

`Destructor:`

Guess….
(no need to worry about in Java)

Objects and Classes

`Relationships:`

`is-a` (inheritance of more abstract class)

`has-a` (use an instance of another class)