CS 162
Intro to Programming II
Object Oriented Design
Design

• Develop or receive an English language description of the problem or requirements
• Identify input(s)
• Develop algorithms
• How to do this in an OO environment?
Process

• Discover classes
• What will each class do?
• Define relationships between classes
• Brainstorm
  – Identify things (nouns)
    • May not be a physical “thing”, i.e. highway route #
  – Identify actions (verbs)
Nouns

• From brainstorming separate nouns
• They may be candidates for
  – Classes
  – Member variables (data members)
  – Rejection
• Different words may refer to the same “thing”
  – Decide which to use
Class or Variable?

• Is a noun a variable or a class?

• Member Variable if:
  – Simple single piece of information

• Class if
  – There are many pieces of information
  – And/or there are actions also related to that item
Define Class Composition

• Information required
  – The member variables

• Actions performed by the class

• Check cohesion
  – Member data and functions should be related to a single concept or purpose
  – If your class involves multiple distinct concepts you should consider splitting it
Develop the Interface

• What information should be public?
  – Rarely will you have public data members
  – Which of the previously identified actions (verbs) should be available to other classes?
  – Define a function stub (placeholder) for each proposed member function:

    ```cpp
    void deposit(double amount) {
        // empty body
        return false;
    }
    ```

• Add comments first, then write the code
Use Index Cards

• Why?
  – Encapsulates your proposed classes
  – Physical size limits the logical size
  – Helps visualize relationships between classes
  – Easy to swap out one for a proposed alternative
Relationships Between Classes

• Has-a relationship (composition)
  – Class A has objects of class B as members

• Is-a relationship (inheritance)
  – Class A shares structure with objects of class B
    • Class B is a more specialized class

• Uses-a relationship (access)
  – Class A uses objects of class B
  – Class A can access an object B’s private member variables as a friend
Interim Class Hierarchy

- **Animal**
  - string name
  - attack()

- **Is-a**
  - Wolf
  - Has-a Animal
    - attack()
  
- **Is-a**
  - Dog
    - Has-a Animal
    - string breed

- **Has-a**
  - Master
  - Pack
  - Den
Summary

• Brainstorm things and actions in domain
• Separate nouns and verbs
• Identify potential classes by grouping nouns and verbs
• Identify relationships between classes
• Create the class hierarchy to display graphically
• Verify your organization
• Write your code