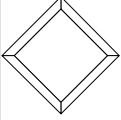



# CS481 - Object-Oriented Programming

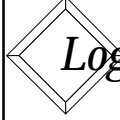


**C++**  
**Object-Oriented  
Programming (CS-481)**

Prof. Jeffrey J. Blessing  
Office: MSOE, S-336  
Phone: 277-7194  
Email: [blessing@msoe.edu](mailto:blessing@msoe.edu)  
<http://www.msoe.edu/~blessing>




1



**Logistics**

- ❖ Schedule
- ❖ Text, code, & notes:
  - [www.awl.com/cp/stroustrup3e](http://www.awl.com/cp/stroustrup3e)
  - [www.msoe.edu/~blessing/cs481.html](http://www.msoe.edu/~blessing/cs481.html)
- ❖ Compiler
- ❖ Exams

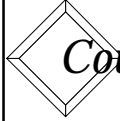
2



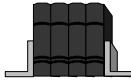
**Logistics**

- ❖ Programs
  - 5 or 6 throughout the quarter
- ❖ Grading
  - 50% exams, 50% programs/quizzes
- ❖ Weekly Quizzes
  - Last day of each week (≈ 15 min. of class)

3




**Course Topics**



- ❖ Object-Oriented concepts
- ❖ C++ language
- ❖ Non-goals (?)
  - OO Design
  - MS Windows
  - STL or Std Lib components


5




**Assumptions**

- ❖ Good general knowledge of C
  - Data types (built-in)
  - Control statements
  - Data structuring
  - Pointers
  - Structs (User-defined data types)

6



**Origins of C++**




- ❖ Bjarne Stroustrup
  - "C with Classes" 1980
    - ◆ Class = set of objects with common characteristics & behaviors
- ❖ Language influences
  - Simula 67, Algol 68, C, Smalltalk, Clu

7

# CS481 - Object-Oriented Programming

**C++ Standardization**

- ❖ Annotated Reference Manual (ARM)
- ❖ ANSI (X3J16, 1989)
- ❖ ISO (SC22-WG21, 1991)



8

**C++ Newer Features**

Not all features supported by all compilers

- ❖ Templates
- ❖ Exception handling
- ❖ New casts
- ❖ Namespaces
- ❖ Miscellaneous

9

**Object Concepts**

- ❖ Encapsulation

**E**

10

**Object Concepts**

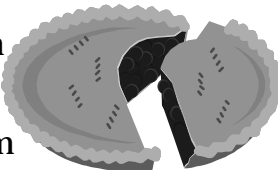
- ❖ Encapsulation
- ❖ Inheritance

**I E**

11

**Object Concepts**


- ❖ Encapsulation
- ❖ Inheritance
- ❖ Polymorphism



**PIE**

12

**Encapsulation**



- ❖ Packaging
  - Data and operations together
- ❖ User data types
- ❖ Implementation hiding
  - Abstract data types (ADT's)

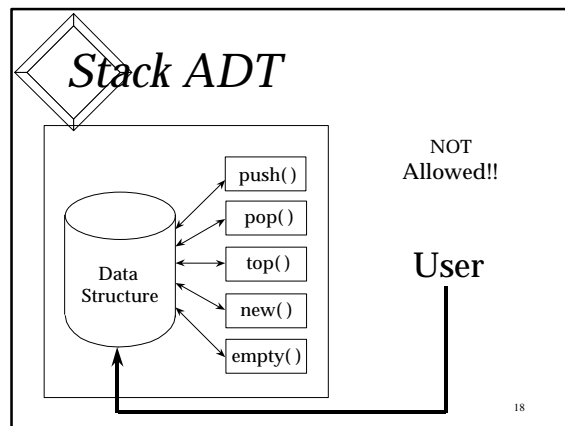
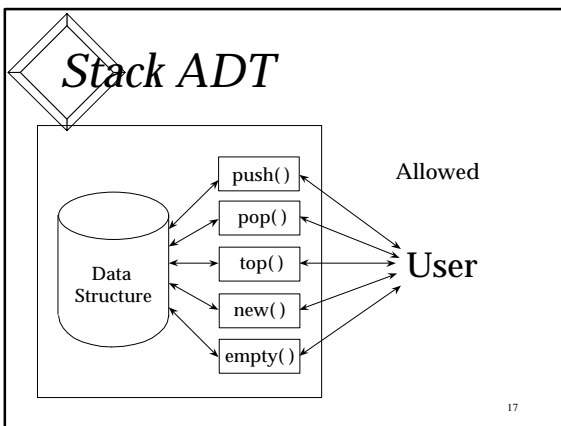
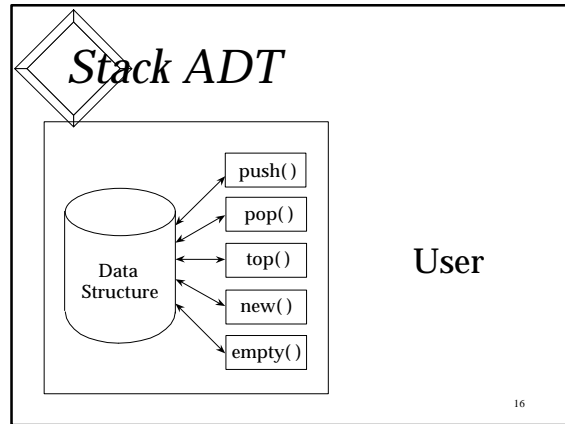
14

# CS481 - Object-Oriented Programming

## Data Encapsulation

- ❖ Also called:
  - Data Abstraction
  - Abstract Data Types (ADT's)
- ❖ C++ implements data abstraction via "classes"

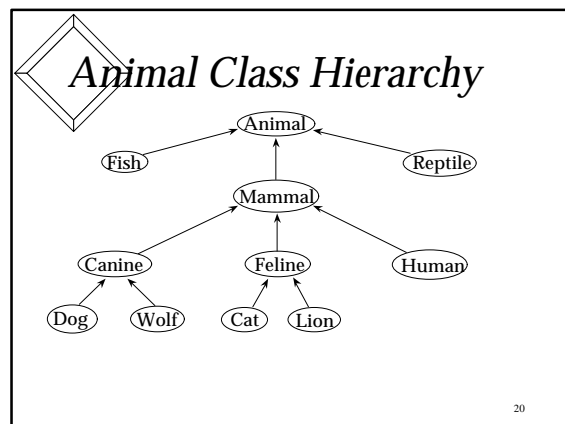
15



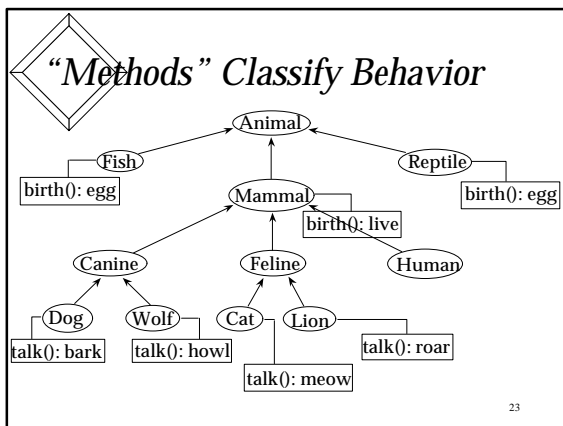
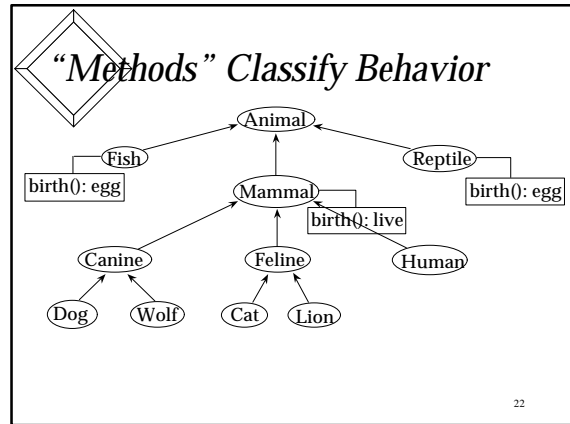
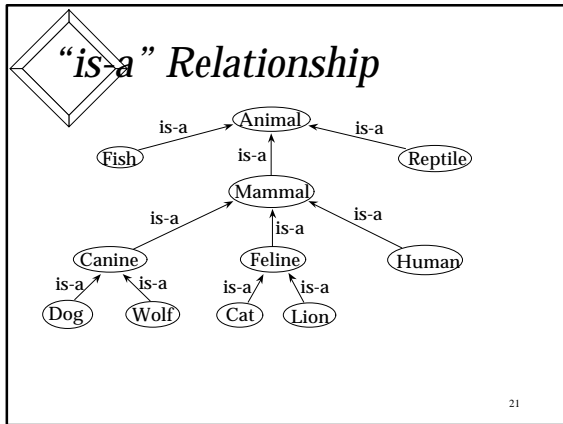
## Inheritance

- ❖ "Is a" relationship
  - Strain gauge is a sensor
    - ◆ Shares common sensor qualities
- ❖ Specialization
  - Strain gauge is more than a sensor
    - ◆ E.g., measures a specific variable

19



# CS481 - Object-Oriented Programming

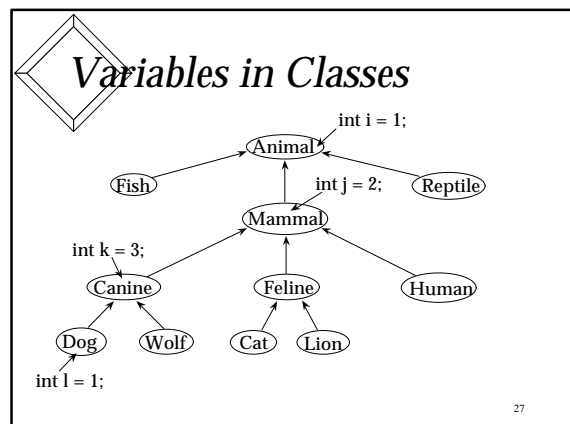
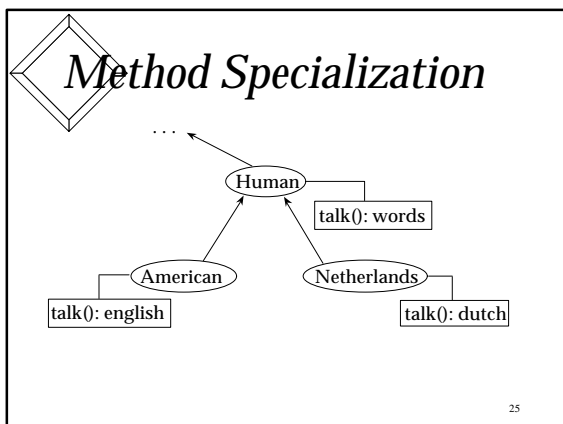


### Polymorphism

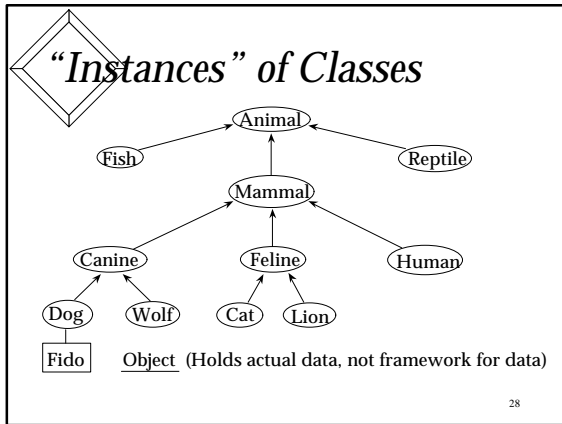
"Multiple forms"

- ❖ Derived "is a" base type
  - Implements common behavior, but ...
- ❖ Implemented differently
  - E.g., sensor calibration sequence
- ❖ "Variant over specialization"

24



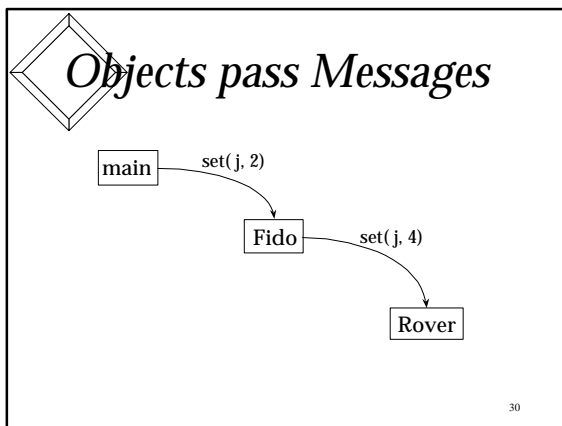
# CS481 - Object-Oriented Programming



## Fido Object (symbolic)

is-a : Dog, Canine, Mammal, Animal
i: 1
j: 2
k: 3
l: 1
talk(): <see class definition>

data members are owned by the object



- ## OO Terminology
- Instances  $\cong$  Objects
  - Classes  $\cong$  Data Types (w/Inheritance)
  - Instance vars  $\cong$  vars inside an object
  - Methods  $\cong$  Functions (form interface)
  - Messages  $\cong$  Function calls (invoke methods)

- ## C++ Terminology
- Instances  $\cong$  Objects
  - Classes  $\cong$  class (ADT)
  - Instance vars  $\cong$  data members
  - Methods  $\cong$  Member Functions
  - Messages  $\cong$  Member Function calls

