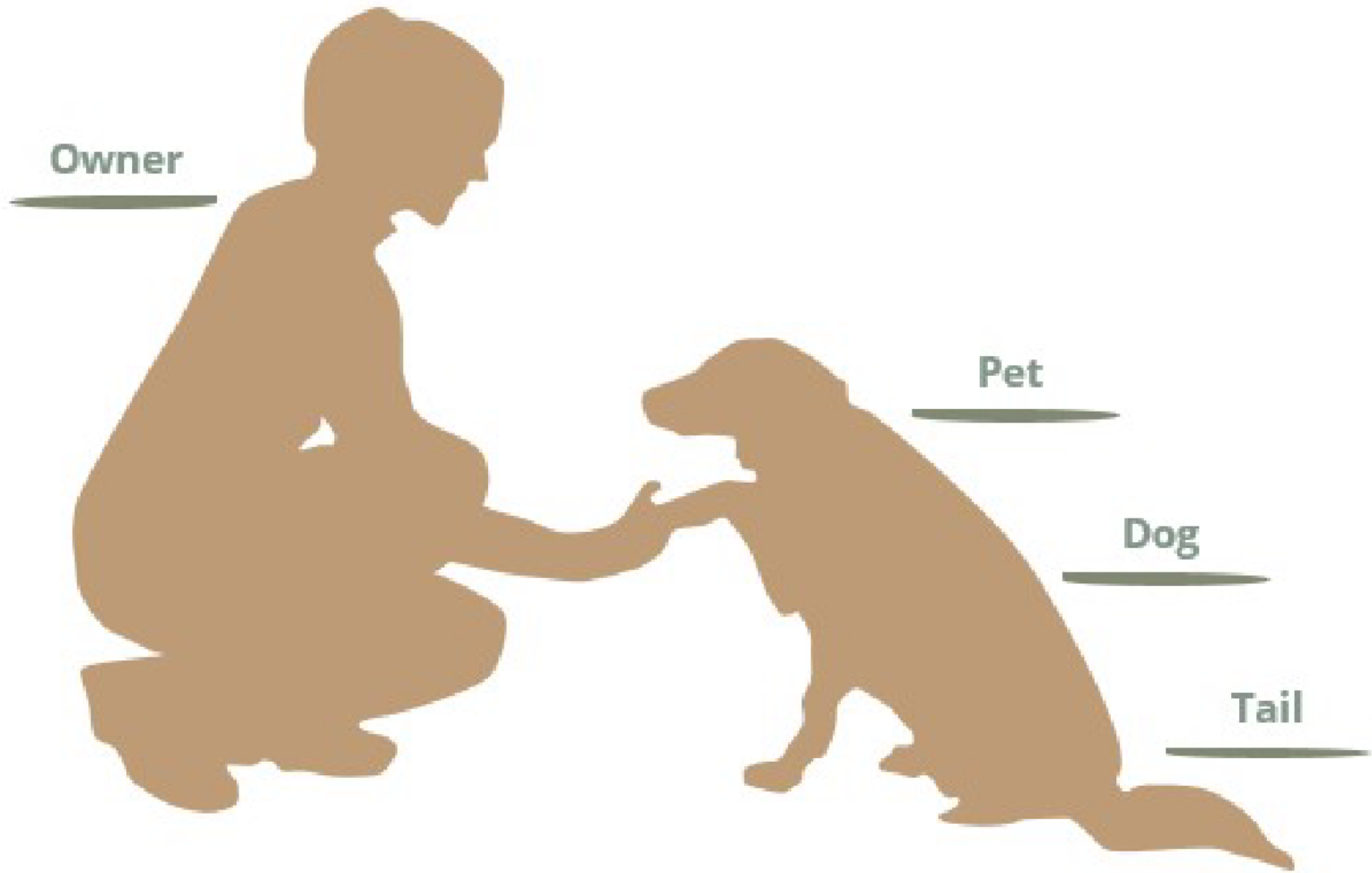
ICS 141 Programming with Objects

Jessica Maistrovich

Metropolitan State University

Object-Oriented Design

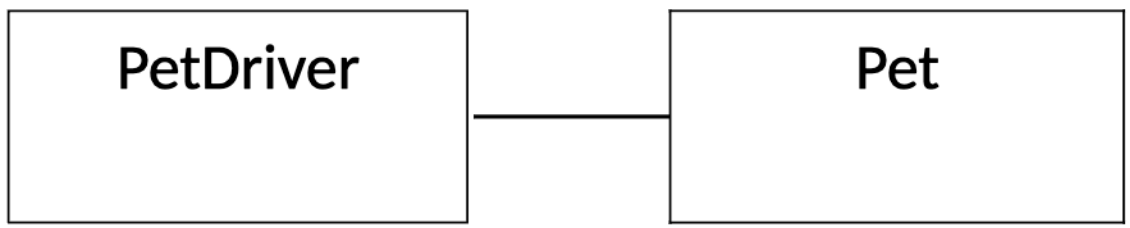
- Class is basic building block
- Programs consist of multiple classes
- Relationships between classes
  - Uses
  - Association (HAS-A)
  - Inheritance (IS-A)
  - Realizes (interface)
- UML structure diagrams are used to visually document relationships between classes



https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/

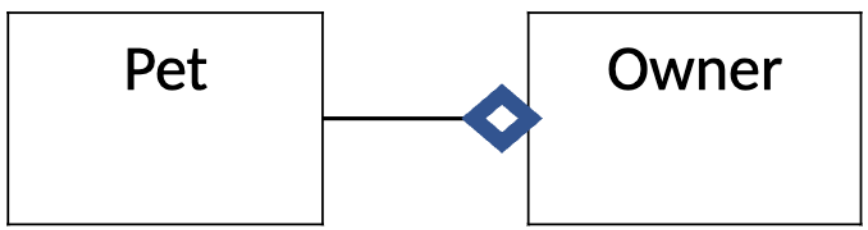
Association (uses)

- **Association** is the loosest form of relationship
- Example: A driver class with a main() method may use a number of different classes
- The UML symbol for the **uses** relationship is a straight line



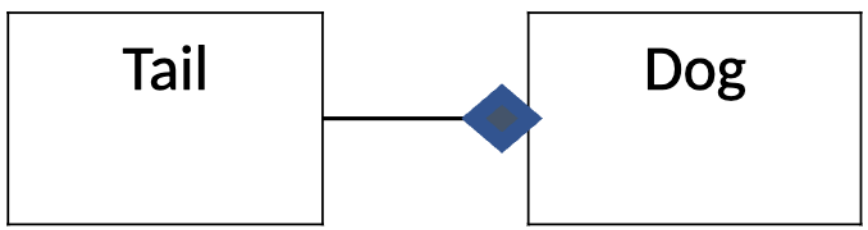
Aggregation (has-a independent)

- A subset of association (uses)
- Aggregation == HAS-A
- Examples:
  - An owner has a pet
  - A class has several students
- Implementation: One class has an instance variable for another class that **can exist on it's own**
- Has-a relationships are shown in UML structure diagrams as straight lines with a diamond head next to the containing class – open for aggregation



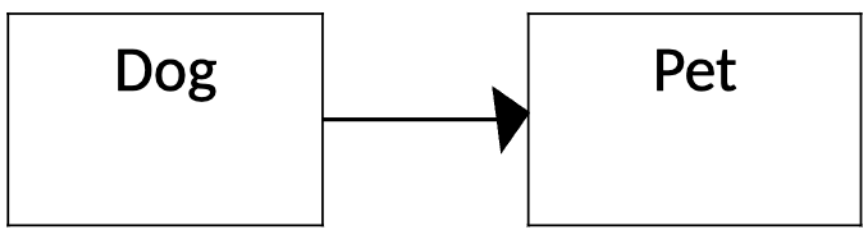
Composition (has-a dependent)

- A subset of association (uses)
- Composition == HAS-A
- Examples:
  - A dog has a tail
  - A person has several fingers
- Implementation: One class has an instance variable for another class that **can’t exist on it’s own**
- Has-a relationships are shown in UML structure diagrams as straight lines with a diamond head next to the containing class – solid for composition



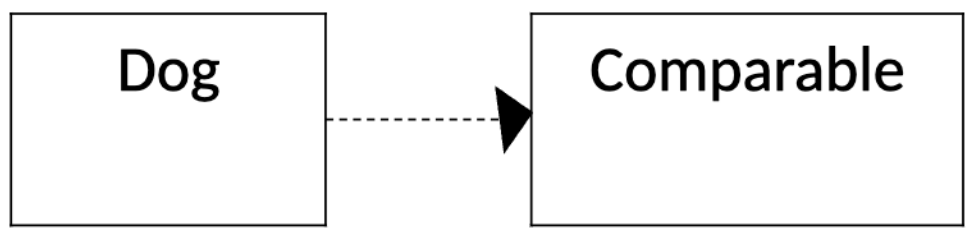
Inheritance (is-a)

- Inheritance == IS-A
- Examples:
  - A dog is a pet
  - A student is a person
- Implementation: One class (child) extends another class (parent)
- Is-a relationships are shown in UML structure diagrams as a solid line with an arrow head next to the parent class



Realization

- Implementation: One class implements an interface
- Interfaces are shown in UML structure diagrams as a dotted line with an arrow head next to the implemented class



UML Class Diagrams

- UML structure diagrams show the relationships between classes
- UML class diagrams provide the details for one class

Point Class Diagram

Point
-x : int -y : int
+Point() +Point(int, int) +getX() : int +getY() : int +setX(int) : void +setY(int) : void +shift(int, int) : void +toString() : String +equals(Object) : boolean +distance(Point, Point) : double +midPoint(Point, Point) : Point

Structure Chart for Point-Line-Shape Application