

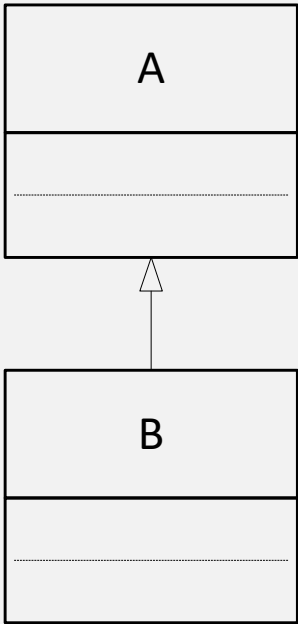


INHERITANCE

Also known as “Generalization”



CLASS ROLES



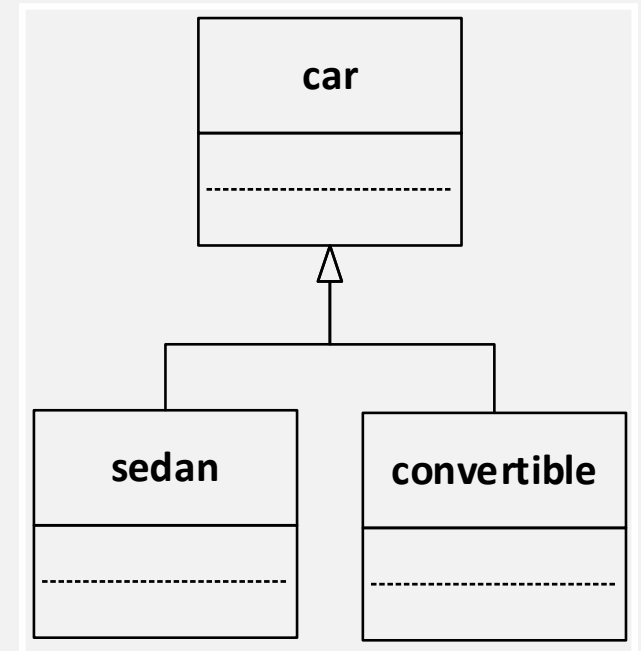
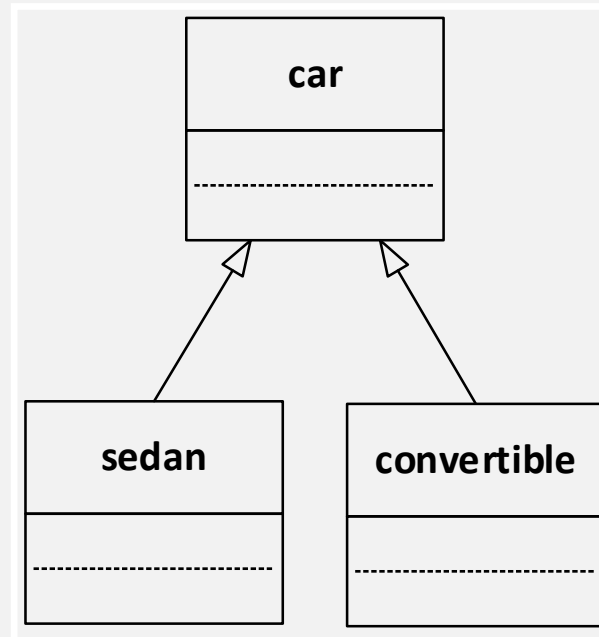
Parent
Superclass
Base class
Ancestor

Child
Subclass
Derived class
Descendant

- Each related class plays a distinct role
 - Parent/Child
 - Superclass/Subclass
 - Base class/derived class
 - Ancestor/Descendant
- UML symbol is a line with a hollow, three-sided arrowhead at one end and undecorated at the other end

UML INHERITANCE SYMBOL

- Two styles used
 - Individual arrows for a few subclasses
 - Shared arrows for many subclasses
- Generalization
 - car is general
 - sedan and convertible are more specific





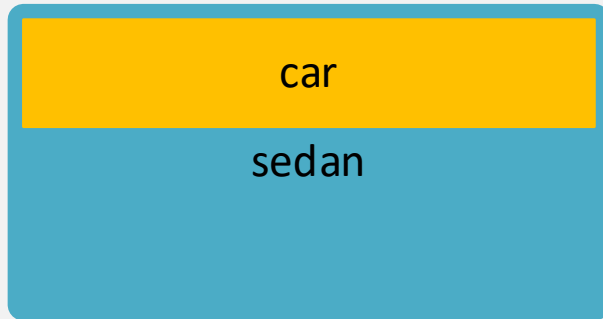
INHERITANCE SEMANTICS / MEANING

- An *is a* relationship
 - A sedan *is a* car
 - A convertible *is a* car
- Each subclass inherits all features (members) defined in the superclass
 - Inheritance is a kind of code reuse – subclasses do not need to redefine features
 - If a car has a color variable, then both sedan and convertible inherit the color
 - If a car has a start function, both sedan and convertible inherit the function

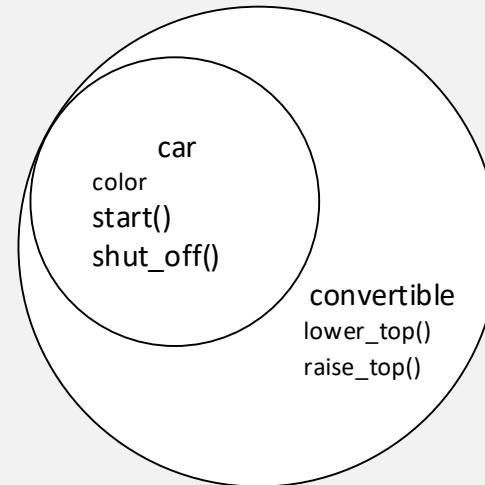


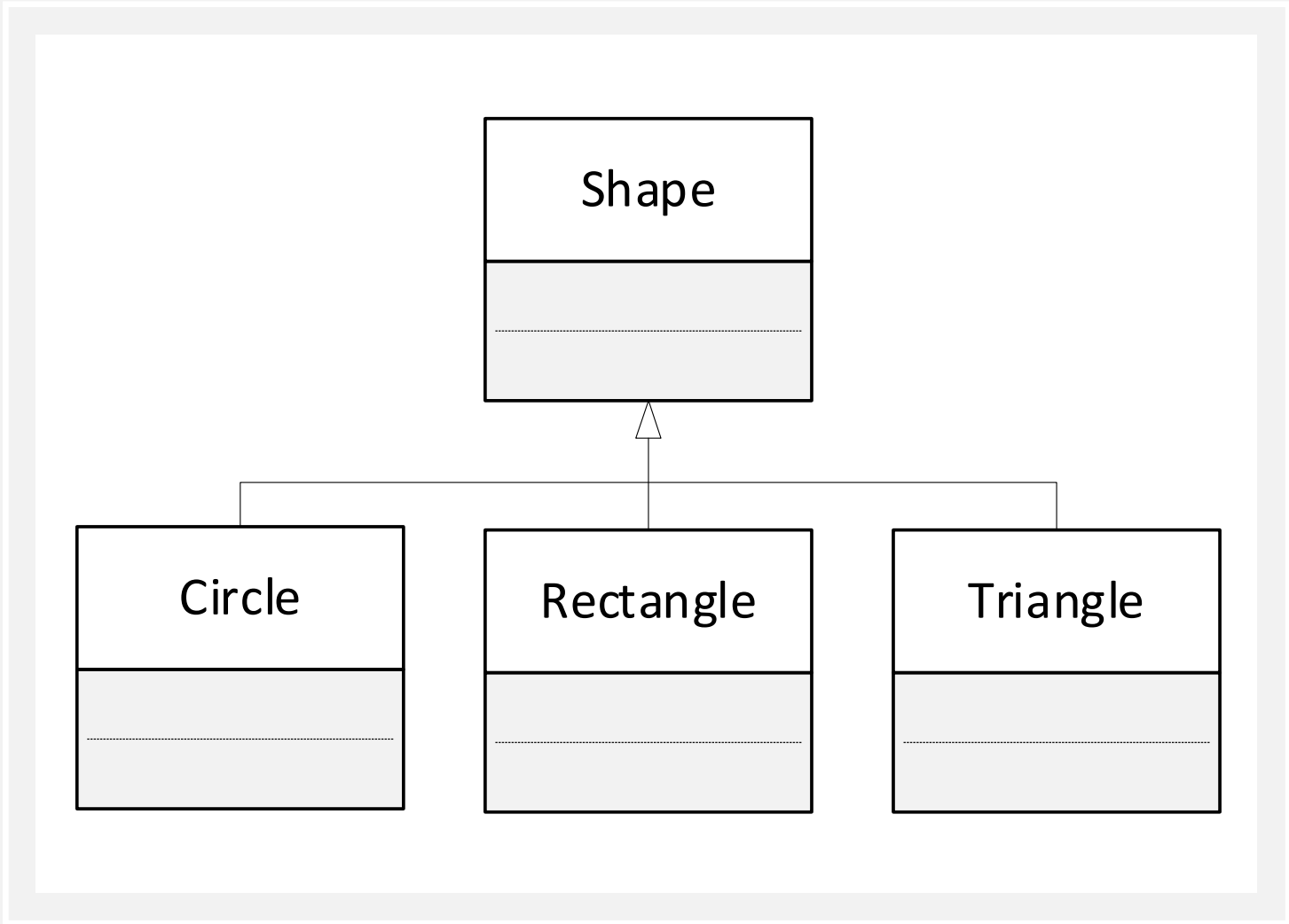
IMPLEMENTING INHERITANCE

OBJECTS



SIMILARITIES & DIFFERENCES





SUBSTITUTABILITY

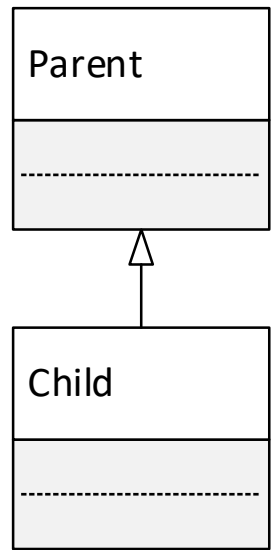
```
void render(Shape* s) { ... }
```

...

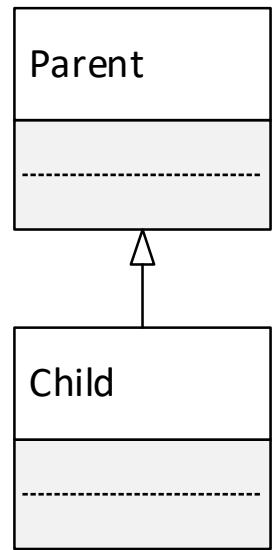
```
Circle c;  
render(&c);
```



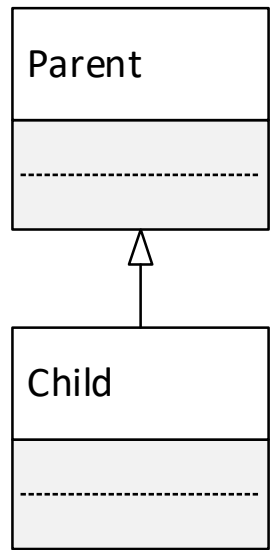
DIRECTIONALITY



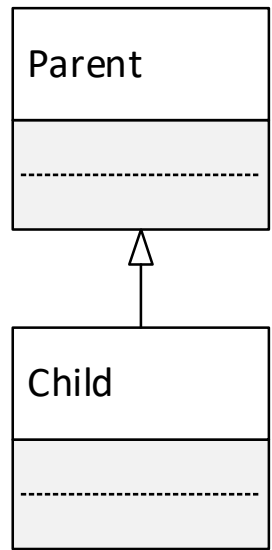
↑
Unidirectional



↑
Messages



↑
"Knows About"



↑
Navigate

BINDING STRENGTH LIFETIME AND SHARING

- Parent is strongly/tightly bound to child
 - Parent and child are created and destroyed simultaneously (they live and die at the same time)
 - The child does not share its parent *object* with any other object

car

sedan