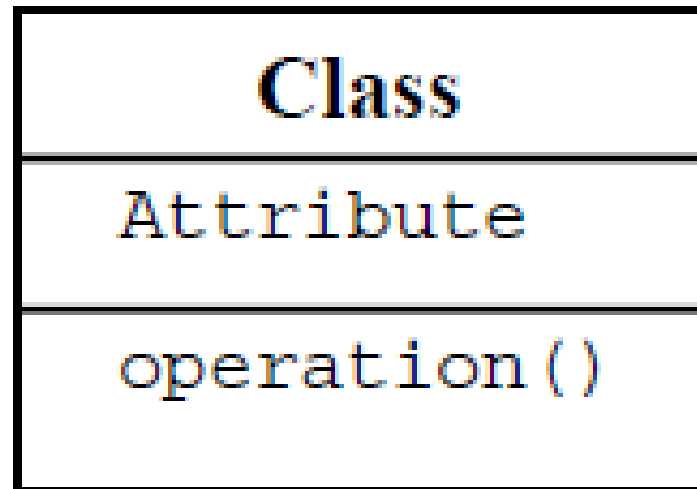


Object Oriented Approach

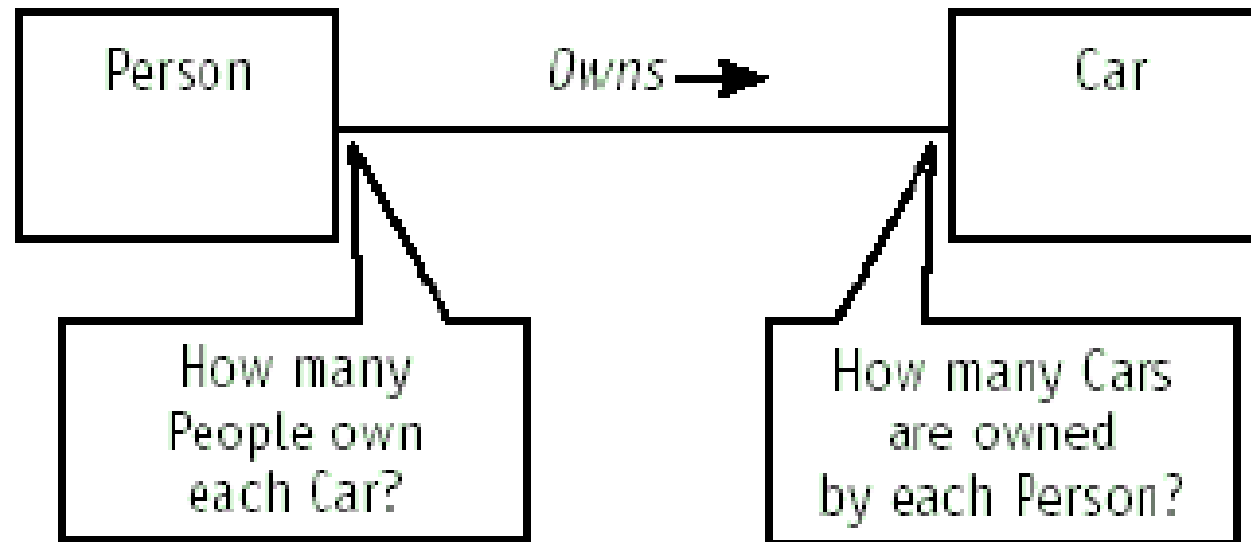
Chapter4

UML Class Diagram Notation



Association

- Association: static relationship shared among the objects of two classes



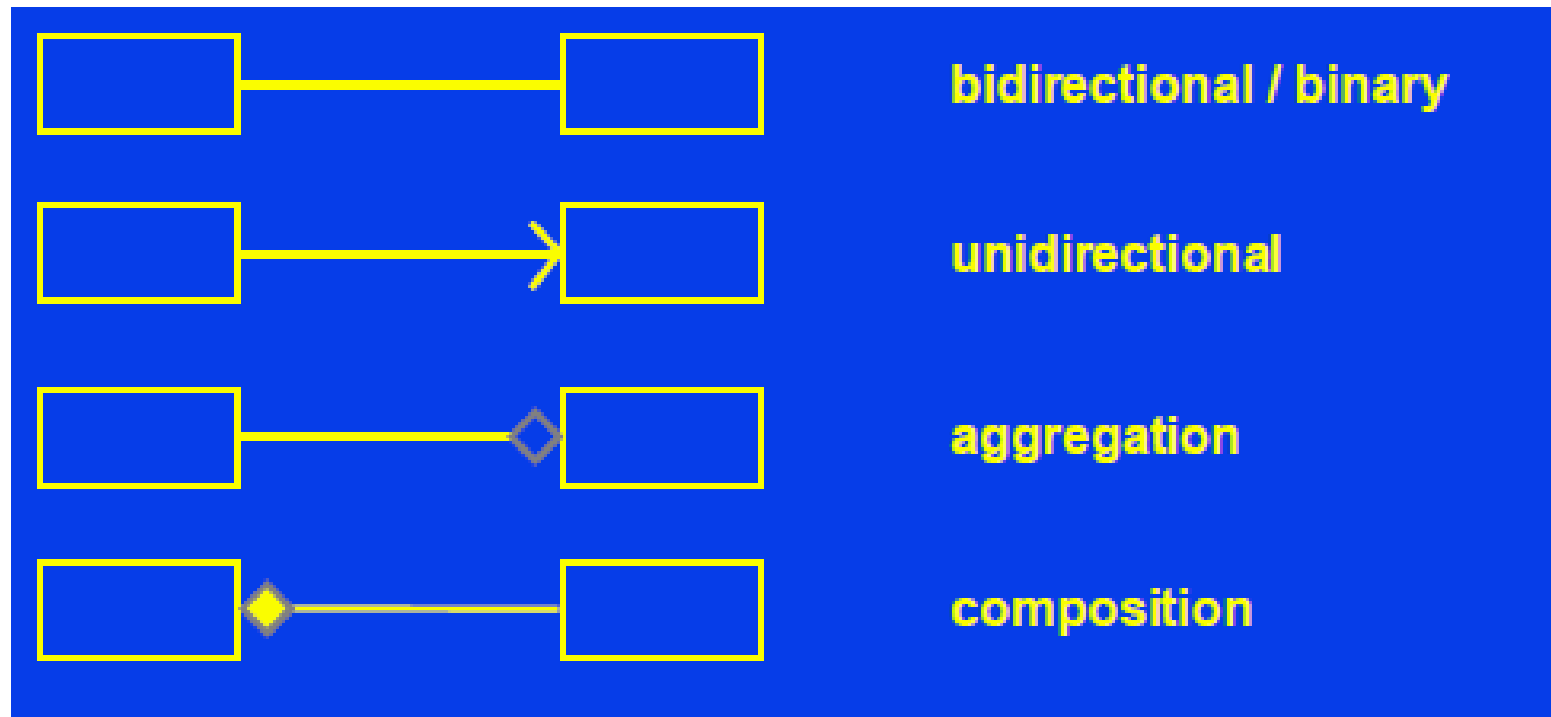
Aggregation

- If the association conveys the information that one object is part of another object (“has-a”), but their lifetimes are independent (they could exist independently).
- Aggregation is unidirectional

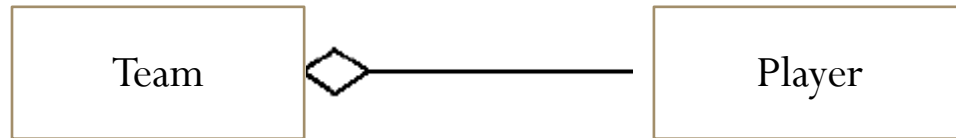
Composition

- When an object is contained in another object, and it can exist only as long as the container exists and it only exists for the benefit of the container.

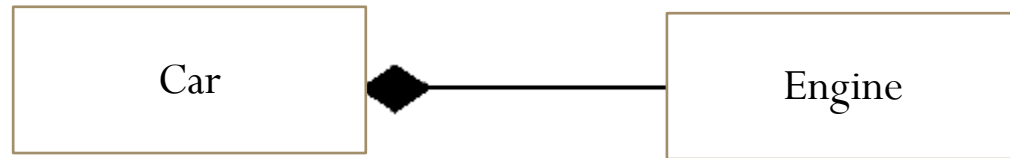
Associations in UML



aggregation

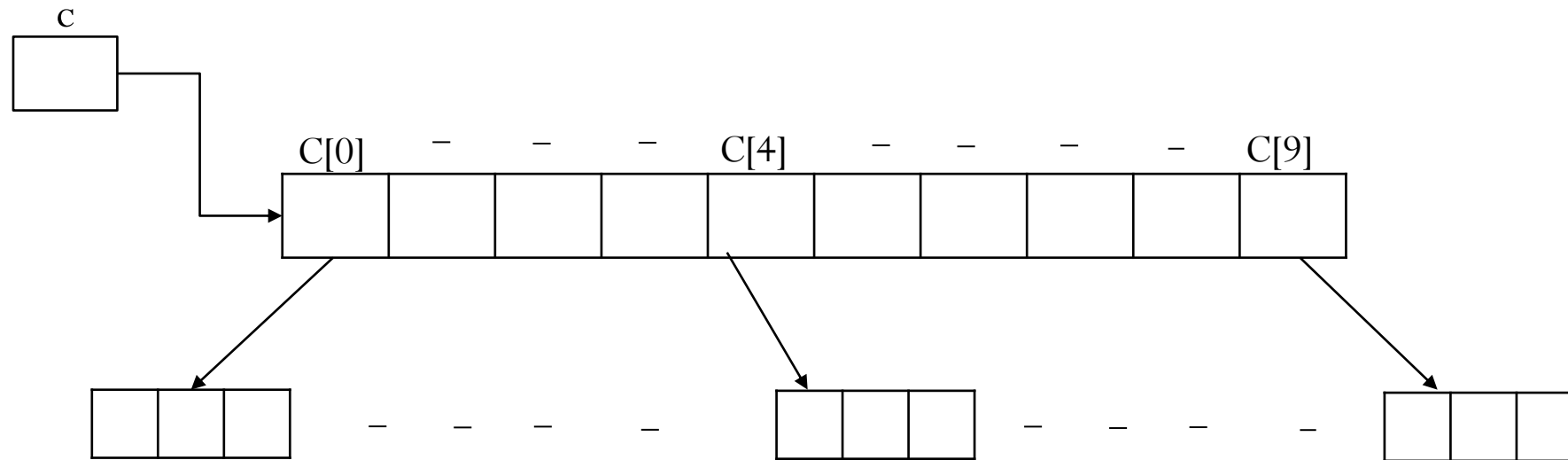


Composition



Array of objects

- `Circle[] c=new Circle[10];`



```
for(i=0;i<10;i++) c[i]= new Circle(.....);
```

Inheritance/Héritage

- Generalization allows:
 - To share the most general knowledge in **super-classes** (*base classes*)
 - To define increasingly specific information by defining specialized classes named *subclasses* (or *derived classes*)
- The inheritance is a concept attached to the classes which support generalization

Inheritance = build a class from other (others)

Inheritance

- How does it work?
 - A subclass inherits the attributes and the methods of its super-class
 - Specialization is done by:
 - Enhancement (new attributes, new operations)
 - Redefinition/Substitution (attributes, operations)
- The inheritance relation is:
 - Transitive
 - Realized with an aim of re-use
 - Can be simple or multiple

