CSE 3421

UML Class Diagram

SPRING 2022

MD. RAFI-UR-RASHID

LECTURER, DEPT. OF CSE, UIU

Why This Lesson?

It is a crucial task in OOP

What is a UML class diagram?

- A UML class diagram is a picture of the classes in an OO system
  - their fields and methods
  - connections between the classes that interact or inherit from each other

- Not represented in a UML class diagram:
  - details of how the classes interact with each other
  - algorithmic details; how a particular behavior is implemented

Class attributes (fields, instance variables)

visibility name : type [count] = default_value

- Visibility
  - \+ public
  - # protected
  - - private
  - ~ package (default)
  - / derived
- Method listed as name: return type
- Parameters listed as name: type
- Omit return type on constructors and when return type is void

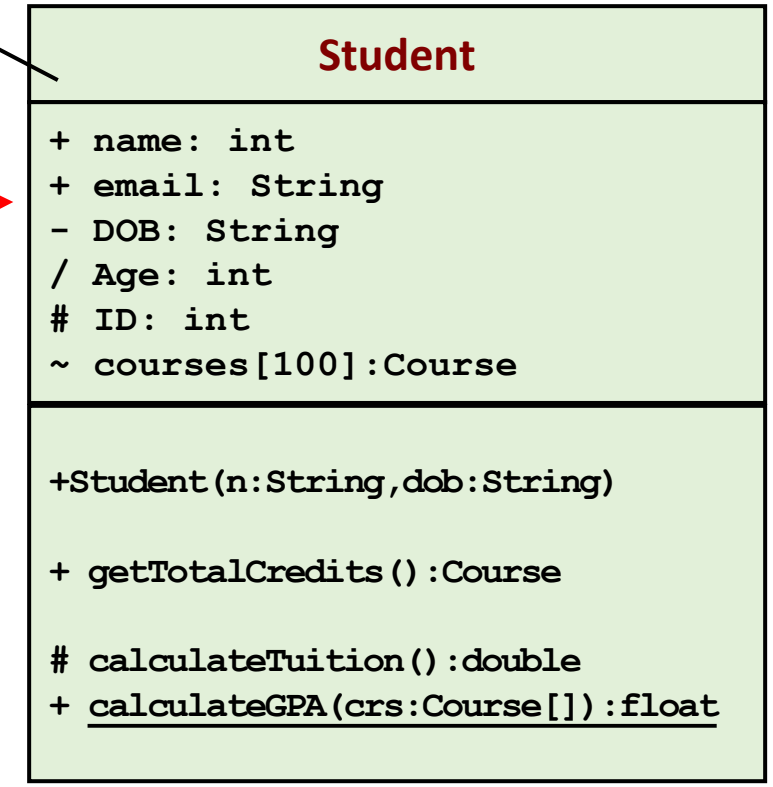
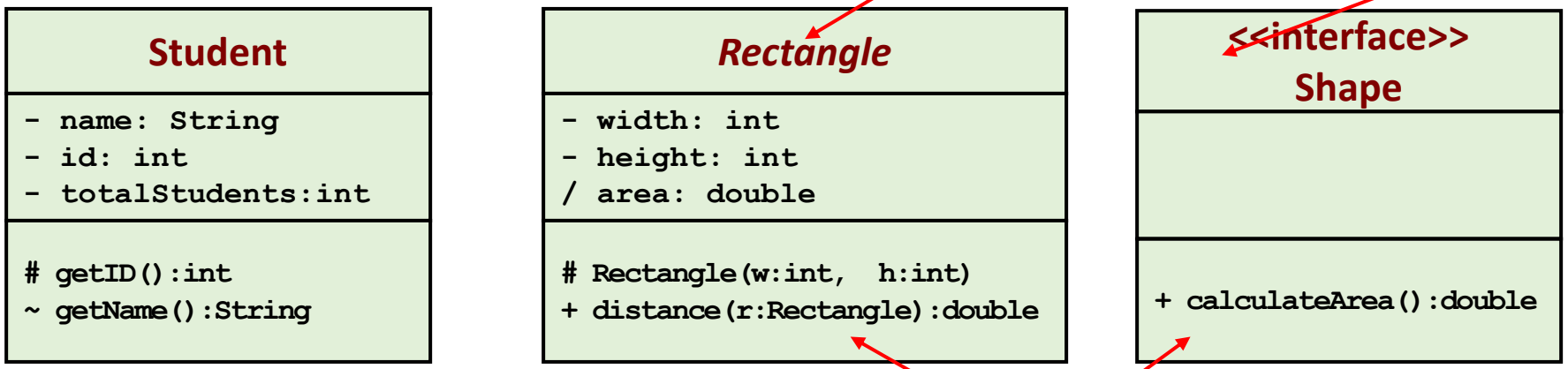


Diagram of a single class

Class name on top
- write «interface» on top of interfaces' names
- use *italics* for an abstract class name



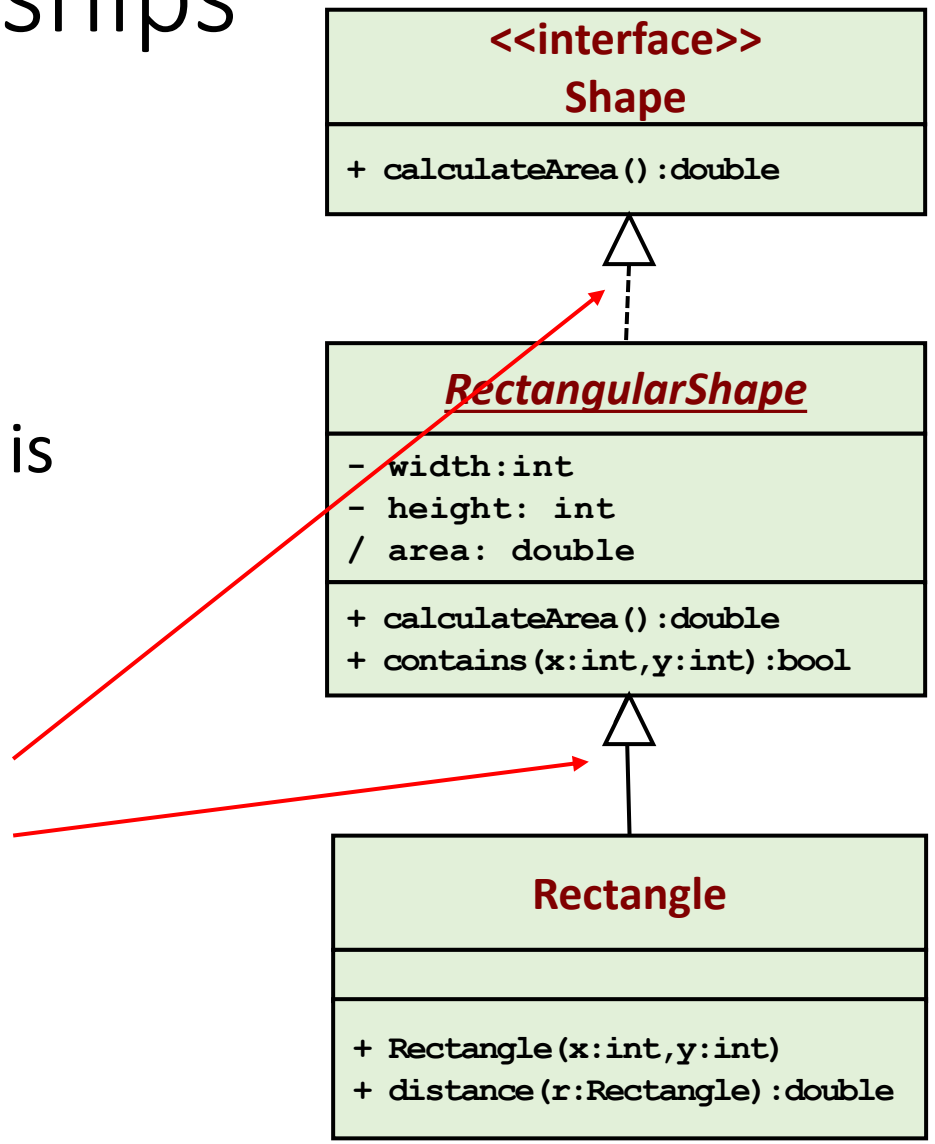
Operations/ methods
- may omit trivial (get/set) methods
- but don't omit any methods from an interface!
- should not include inherited methods

Relationships between class

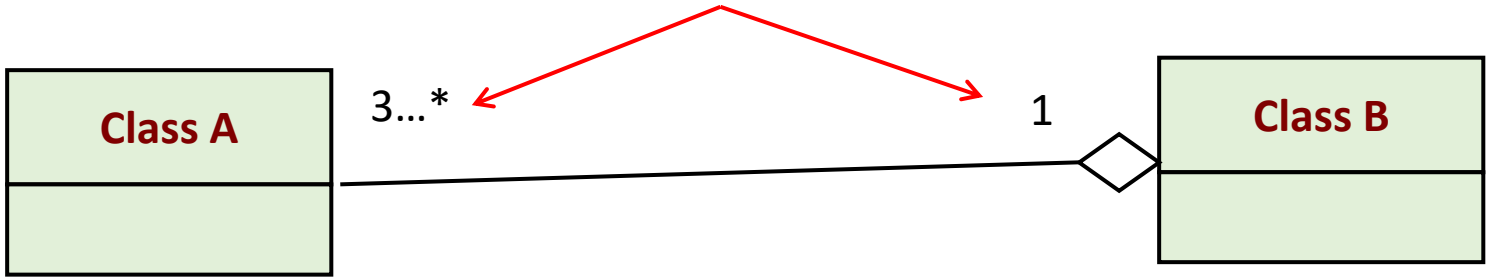
- **Generalization**: an inheritance relationship
  - inheritance between classes
  - interface implementation
- A**ssociation**: a usage relationship
  - dependency
  - aggregation
  - composition

Generalization relationships

- Hierarchies drawn top-down
- Arrows point upward to parent
- Line/arrow styles indicate if parent is a(n):
  - **class**: solid line, black arrow
  - **interface**: dashed line, white arrow
  - **abstract class**: solid line, white arrow



Association (usage) relationships



Association multiplicities

One to one

- Each car has exactly one engine
- Each engine belongs to exactly one car

One to many

- Each book has many pages
- Each page belongs to exactly one book

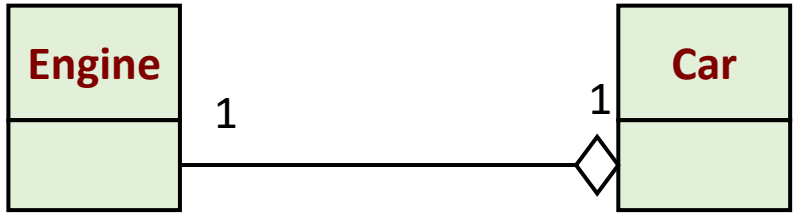
Multiplicity (how many are used)

- * (zero or more)
- 1 (exactly one)
- 2..4 (between 2 and 4, inclusive)
- 3..* (3 or more, * may be omitted)

Association types

Aggregation: “is part of”

- symbolized by a clear white diamond



Composition: “is entirely made of”

- stronger version of aggregation
- the parts live and die with the whole
- symbolized by a black diamond

Dependency: “uses temporarily”

- symbolized by dotted line
- often is an implementation detail, not an intrinsic part of the object's state

Thank You