Basic OOP Concepts

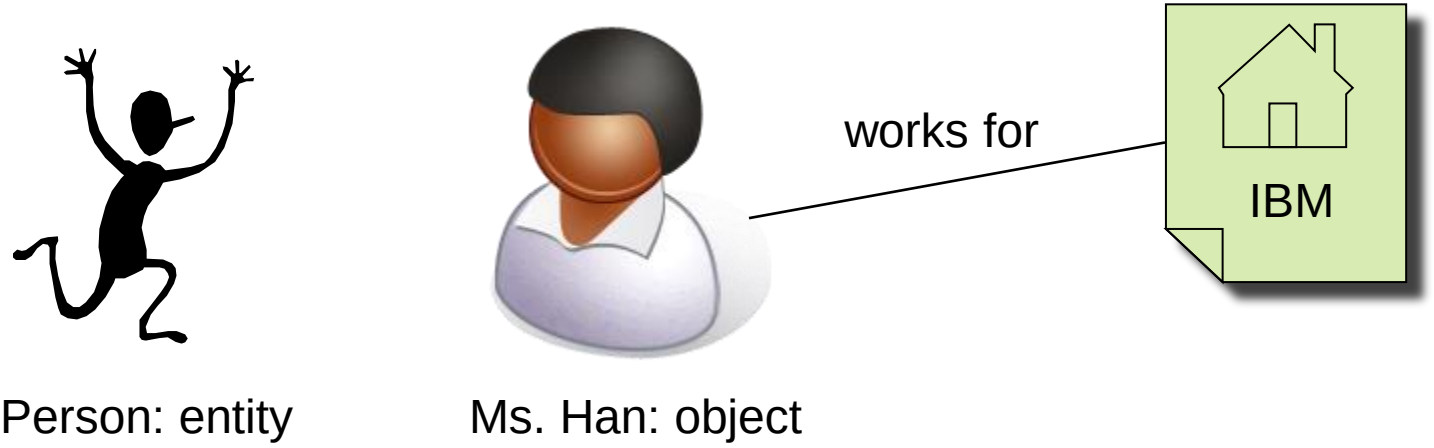
Nguyen Van Vu
nvu@fit.hcmus.edu.vn

Topics

- Object, link, class, attribute, operation, method, association
- Software reuse

Object

- An object is a discrete entity
  - an entity at a particular point in time
- An object has identity encapsulating state and behavior

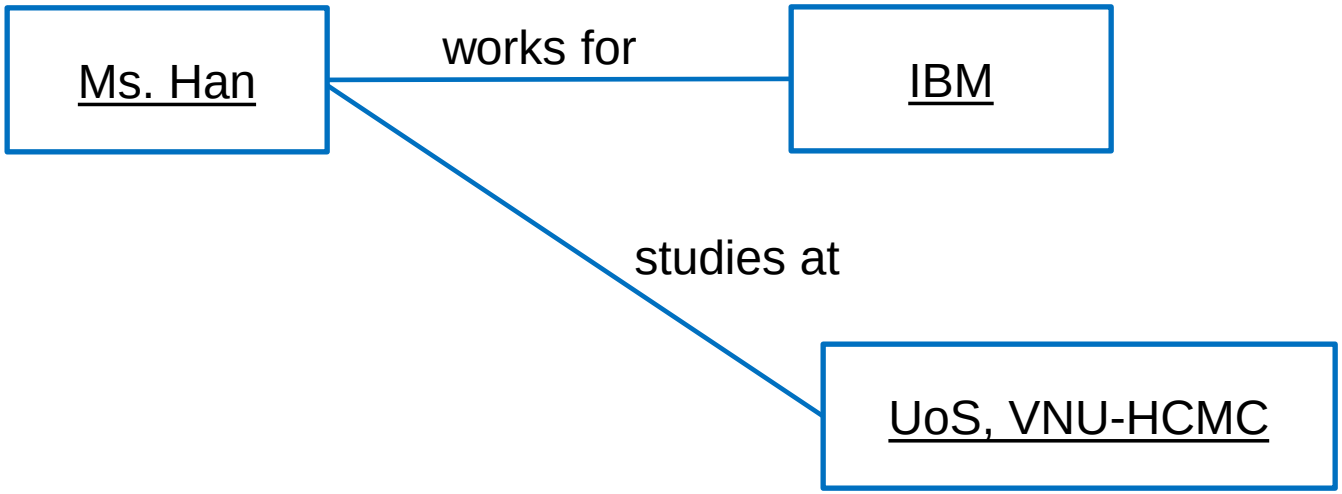


Person: entity

Ms. Han: object

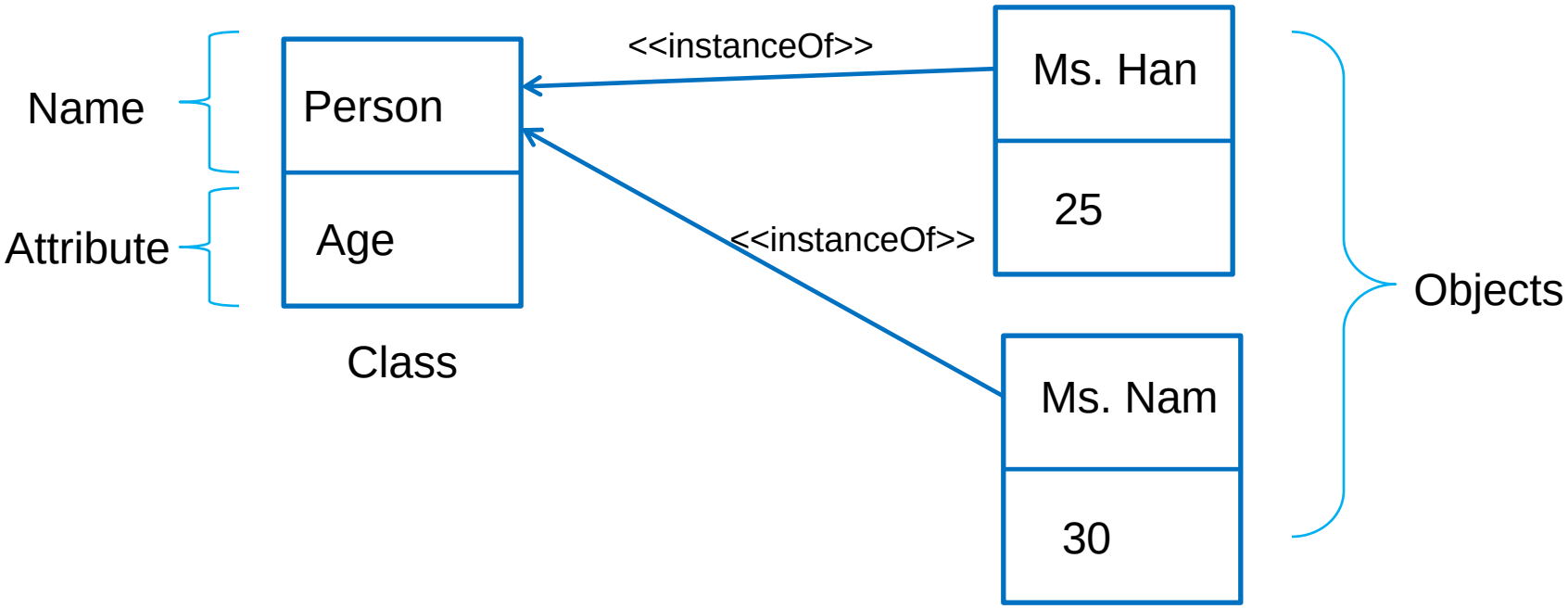
Link

- An object is connected with other objects via links
- Objects and a link together represent a fact



Class

- A class defines the set of common objects that can be created
- It's like a cake mold to create cakes (objects)



Attribute

- Attributes define data that characterize a class
  - An attribute is the description of a named slot for a specified type
  - Specifies values that an object holds

Operation and Method

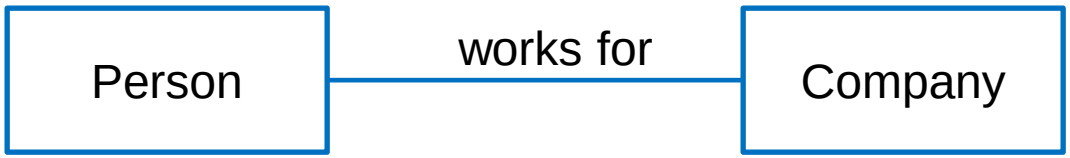
- An operation specifies a service that can be requested from objects of the class
- A method is an implementation of an operation

- Difference between *operation* and *method*?

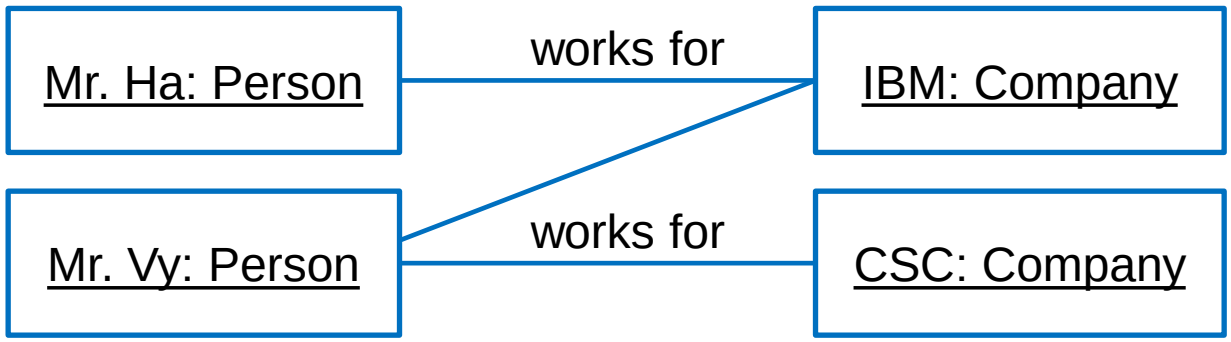
Specification vs. Implementation

Association

- An association represents possible links between objects of classes
- An object is an instance of a class → a link is like an instance of an association



Association



Links

Exercise

- Andrew is an online content creator. He usually shares his videos on social platforms such as Facebook Reels or Tiktok. When he posts a video, the video needs to have at least a name, a description.
- Barbara is a marketing specialist. She often views videos on social sharing platforms like Facebook Reels or Tiktok. This is a way for her to relax or find inspirations in her job. As a content viewer, she can “like” or “love” a video if it interests her. Also as a content viewer, she can follow certain content creator who made several videos which meet her taste

Exercise

- Bob is a farmer. He grows and produces a lot of delicious bananas, and he wanted to sell some. He went to an e-commerce website name amazon.com. He posts his bananas to sell.
- Alice is a software programmer who spends most of her time on computer. She does not have time for shopping, she decided to buy some bananas from amazon.com. She ended up buying the ones from Bob. Bob then shipped the product to Alice. One day later, she received it.

Exercise (cont'd)

- Using the concepts discussed to form
  - Object model
  - Class diagram

Topics

- Object, link, class, attribute, operation, method, association
- Software reuse

Software Reuse

- Software reuse means reusing work products within a system or multiple systems
  - source code
  - components
  - designs, architectures
  - ideas
  - documents, databases
  - ...

Software Reuse (cont'd)

- Benefits
  - reduce time to deliver
  - reduce cost to develop
  - higher quality
  - increase chance to succeed
  - reduce maintenance and evolution cost
  - improve performance, scalability, security, portability, maintainability, other '-ilities'
  - use personnel resources effectively
    - use right people for right work

Reuse Approaches

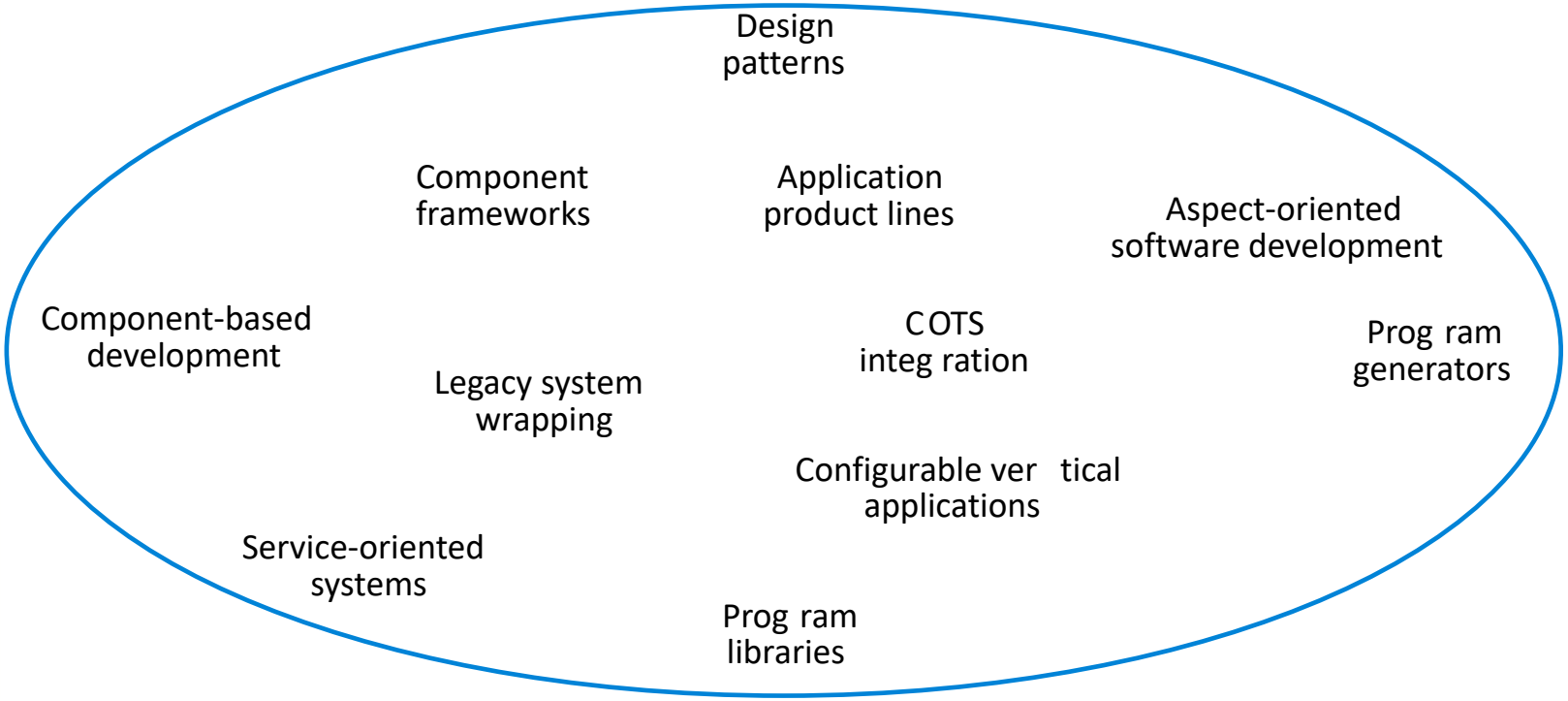
- Ad-hoc: copy-paste, copy-modify
- Function, procedure: these elements are defined and organized for reuse
  - Program libraries for reuse

- **Object-oriented approach**
  - Class, object, interface, inheritance, polymorphism, etc.
  - Class libraries
  - Design patterns

Reuse Approaches (cont'd)

- Component-based
  - Organizing classes into components
  - Components reused in multiple parts of a system or other systems
  - COTS development
  - Component frameworks
- Service-oriented
  - Service-oriented architecture
  - Web-services
- Software product-line
- Legacy system wrapping

Reuse Approaches (cont'd)



Procedural vs. OO Programming

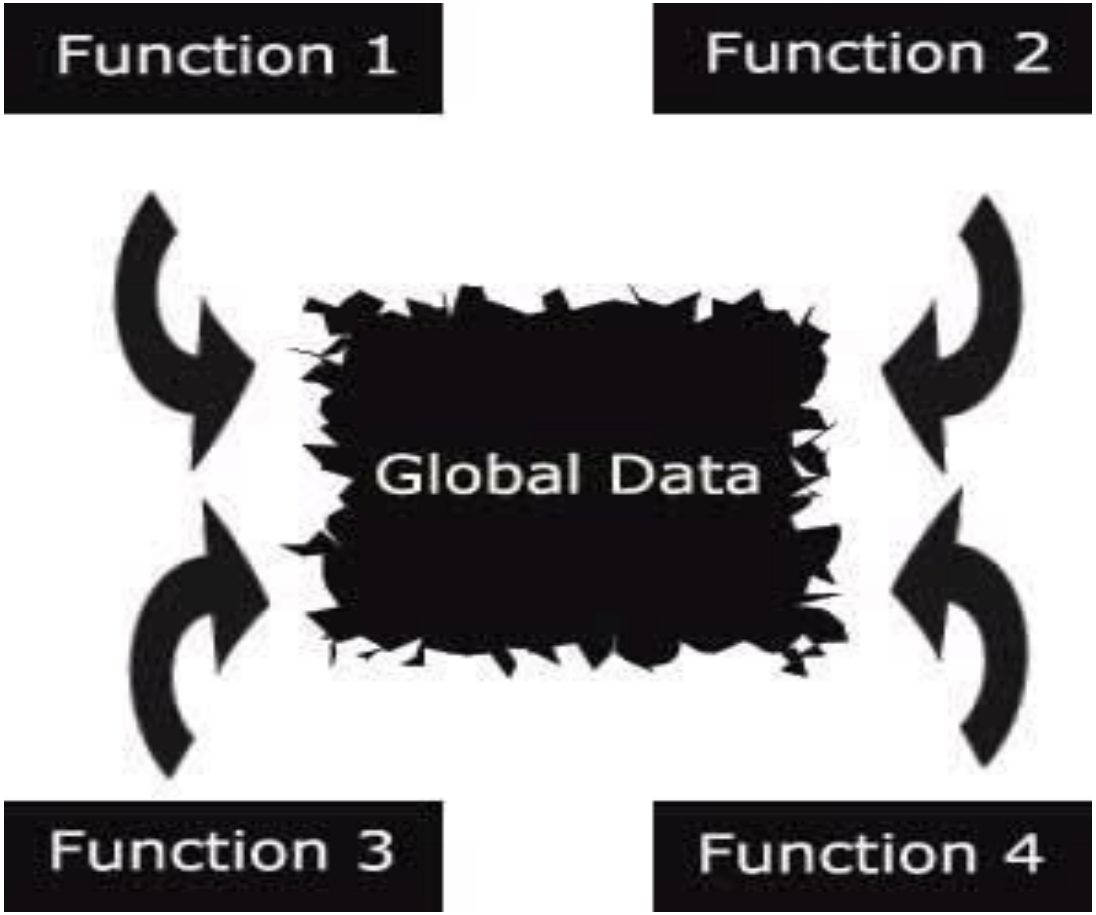
An *object* is an entity
that contains *both* data and behaviours

- In procedural programming:
  - Code is placed into totally distinct functions or procedures.
  - Data is placed into separate structures, and is manipulated by these functions or procedures.

Procedural vs. OO Programming (cont)

- In OO programming: the attributes and behaviours are contained within a single object
- In procedural programming: the attributes and behaviours are normally separated.

Why do we change from procedural to OO programming?



Why do we change from procedural to OO programming?

- In procedural programming:
  - Data is separated from the procedures.
  - Sometimes it is global → easy to modify data that is outside your scope
  - → This means that access to data is uncontrolled and unpredictable.
  - Having no control over the data → testing and debugging are much more difficult.

Why do we change from procedural to OO programming?

- Objects solve these problems by combining data and behaviours into a complete package.

- In a proper OO design: there is no global data.

Moving from Procedural to Object-Oriented Development

- Procedural programming separates the data of the program from the operations.

- Example: if you want to send information across a network, only the relevant data is sent.

➔ handshaking agreement must be in place between the client and server to transmit the data.

Moving from Procedural to Object-Oriented Development

- In OO programming, when an object is transported across a network, the entire object, including the data and behaviours, goes with it.

Exercise

- Define the attributes and behaviours for the object *student*.
- Define the attributes and behaviours for the object *date*.