CLASS DIAGRAM

Prof. Sudip Misra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur
http://cse.iitkgp.ac.in/~smisra/

What is Class Diagram?

- A picture of the classes in an Object Oriented system, their fields and methods, and connections between the classes that interact or inherit from each other
- A general template that we use to create specific instances or objects in the application domain
- Represents a kind of person, place, or thing about which the system will need to capture and store information
- Abstractions that specify the attributes and behaviors of a set of objects

What is Object?

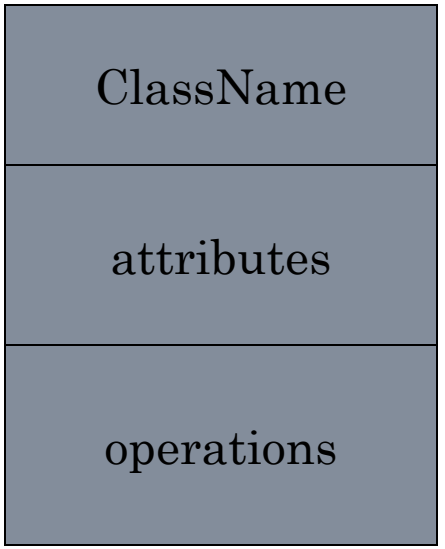
- Entities that encapsulate state and behavior
- Each object has an identity
- It can be referred individually
- It is distinguishable from other objects

Why Class Diagram?

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

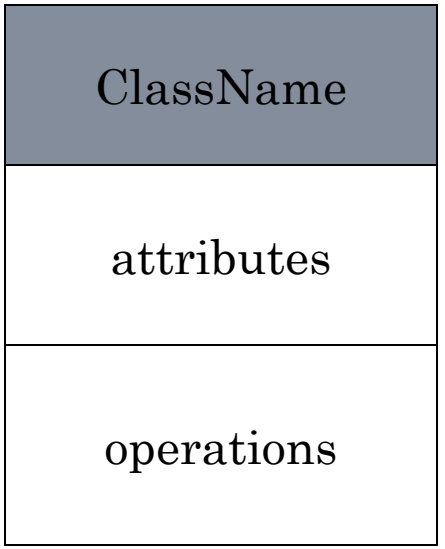
CLASSES

- A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics.
- Graphically, a class is rendered as a rectangle, usually including its name, attributes, and operations in separate, designated compartments.



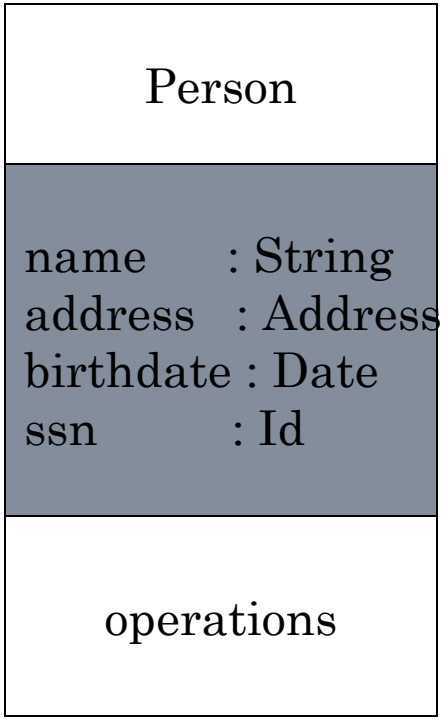
CLASSES

- The name of the class is the only required tag in the graphical representation of a class. It always appears in the top-most compartment.



Class Attribute

- An *attribute* is a named property of a class that describes the object being modeled. In the class diagram, attributes appear in the second compartment just below the name-compartment.



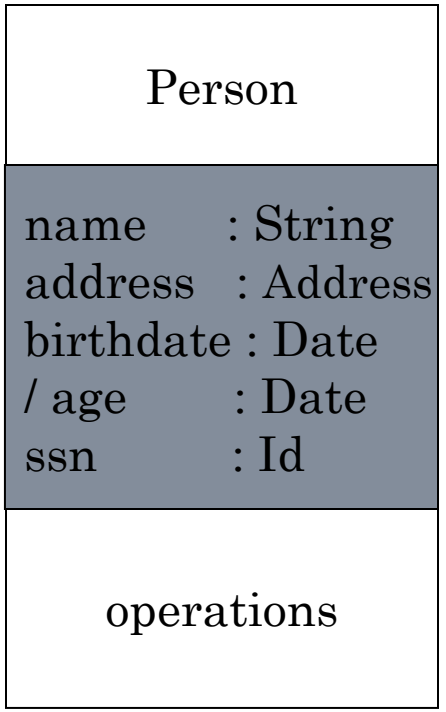
Class Attribute (Contd.)

- Attributes are usually listed in the form:

  attributeName : Type

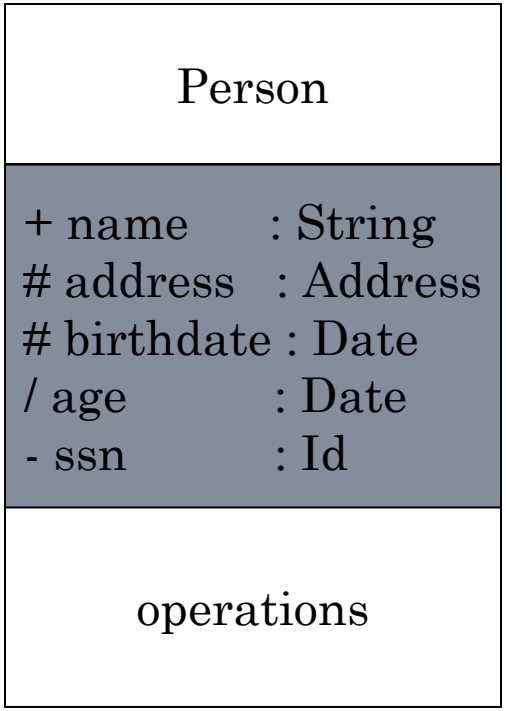
- A derived attribute is one that can be computed from other attributes, but doesn't actually exist. For example, a Person's age can be computed from his birth date. A derived attribute is designated by a preceding '/' as in:

  / age : Date



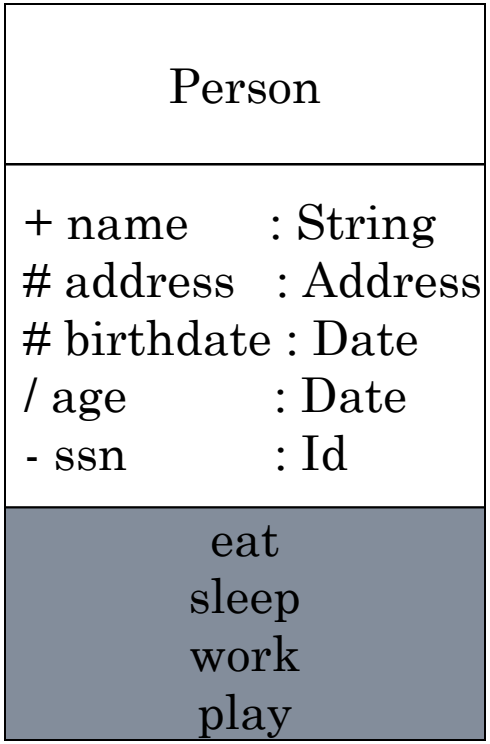
Class Attribute (Contd.)

- Attributes can be:
  - + public
  - # protected
  - - private
  - / derived

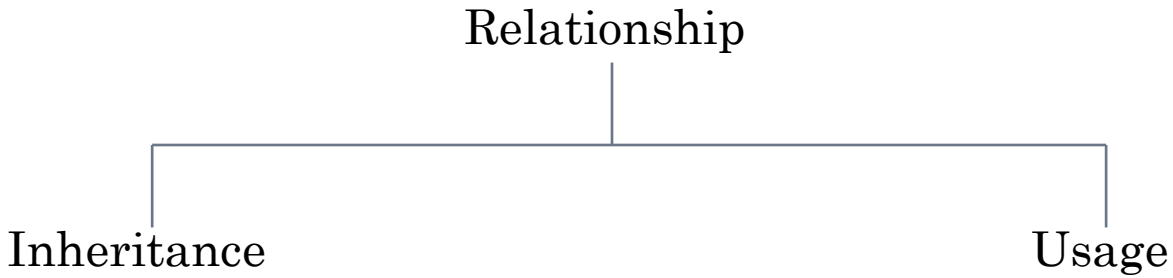


Class Operations

- *Operations* describe the class behavior and appear in the third compartment.



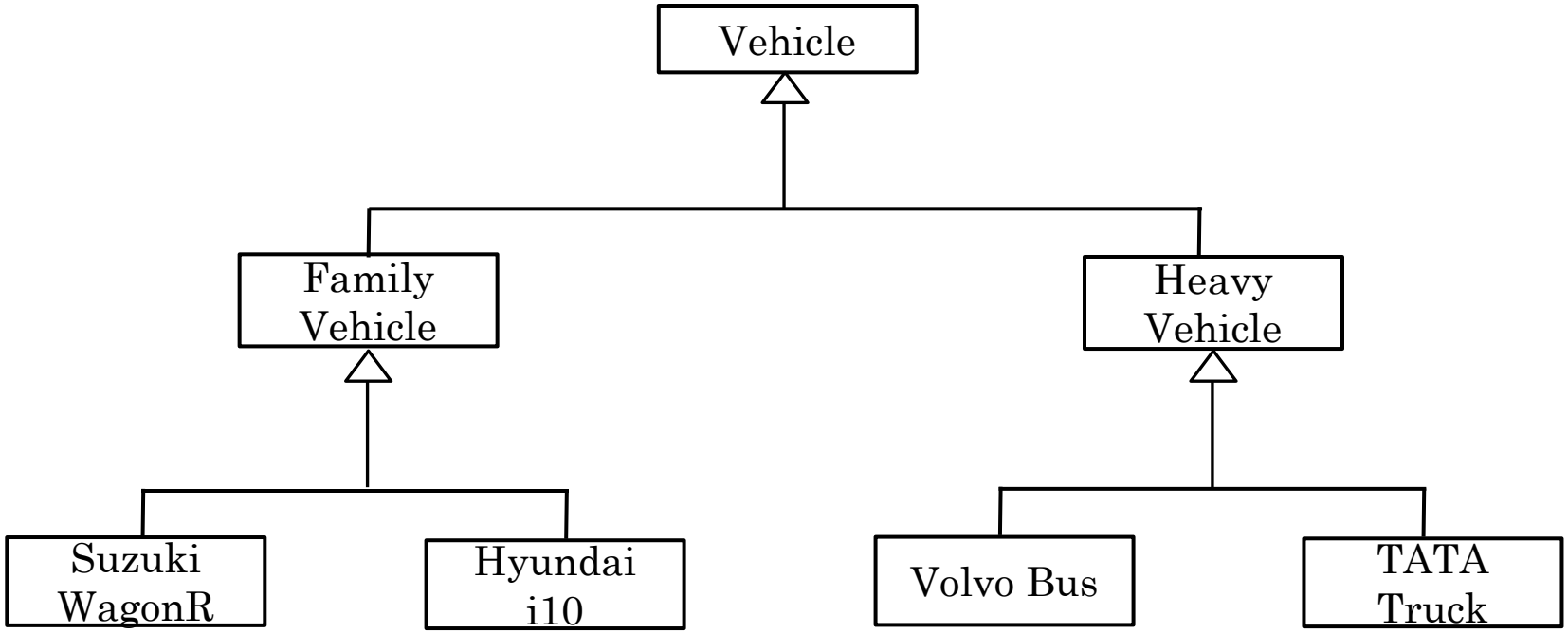
Relationship between Class



- Inheritance: Generalization
- Usage: Association
  - Aggregation
  - Composition

GENERALIZATION

- Enables the analyst to create classes that inherit attributes and operations of other classes



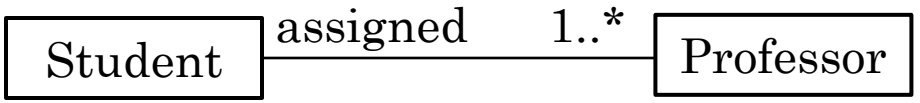
Association

- Associational (usage) relationships
  - Multiplicity

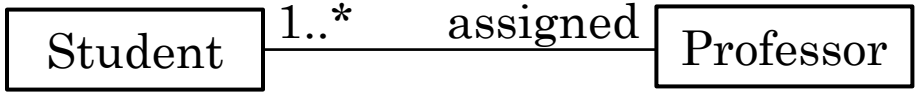
Symbol	Meaning
*	0, 1, or more
1	1 exactly
1..*	1 or more
2..4	between 2 and 4, inclusive
3..*	3 or more

Association (Contd.)

- Example: Multiplicity
  - A student can be assigned one or multiple professors

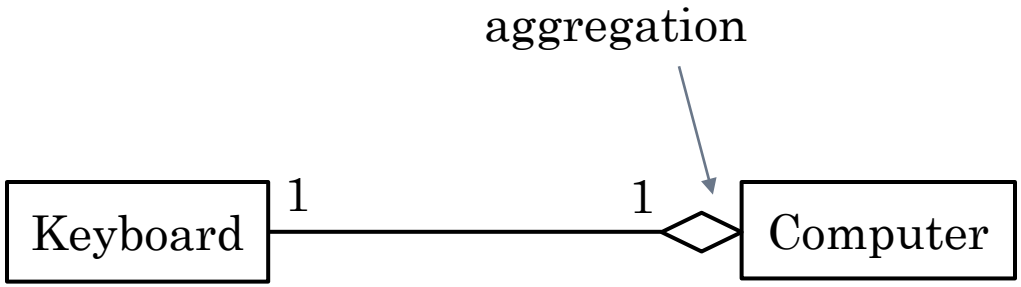


  - A professor can be assigned one or multiple student



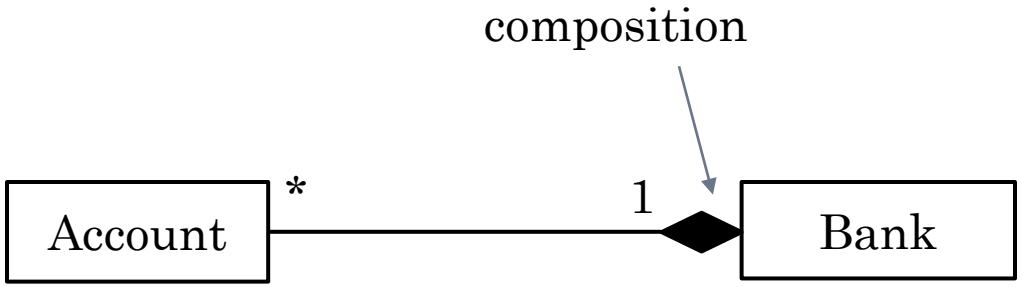
AGGREGATION

- "is part of"
  - symbolized by a clear white diamond



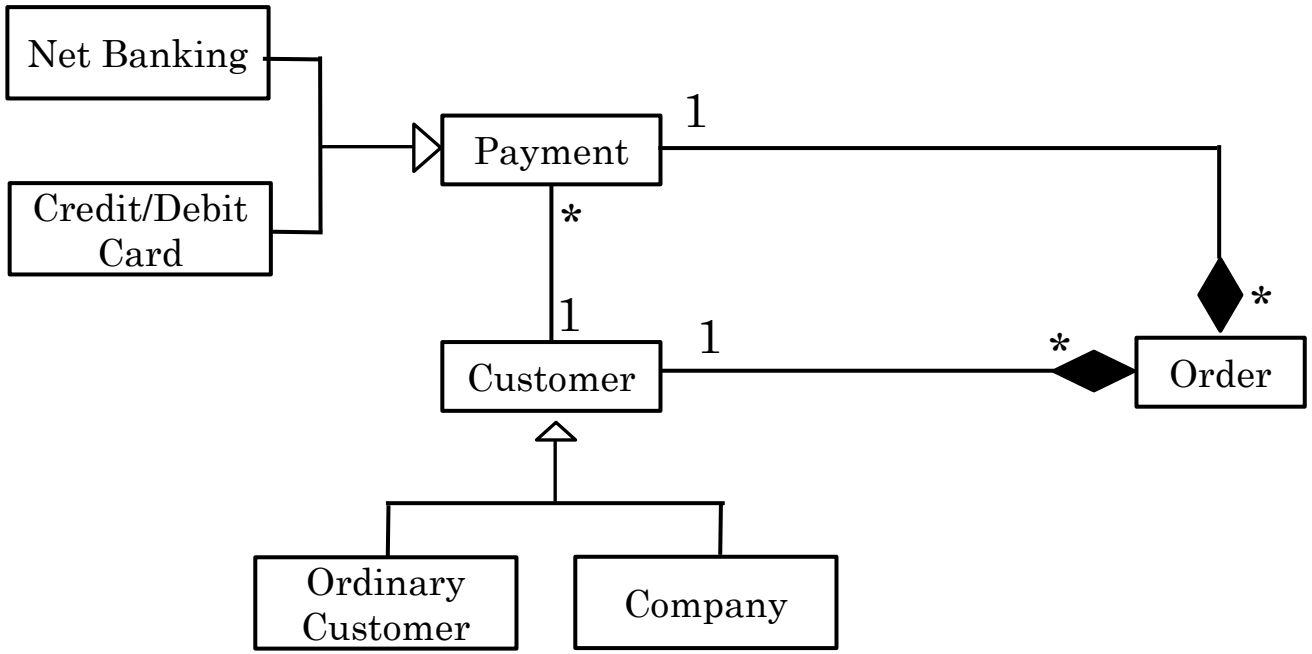
Composition

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



Example: Online Purchase

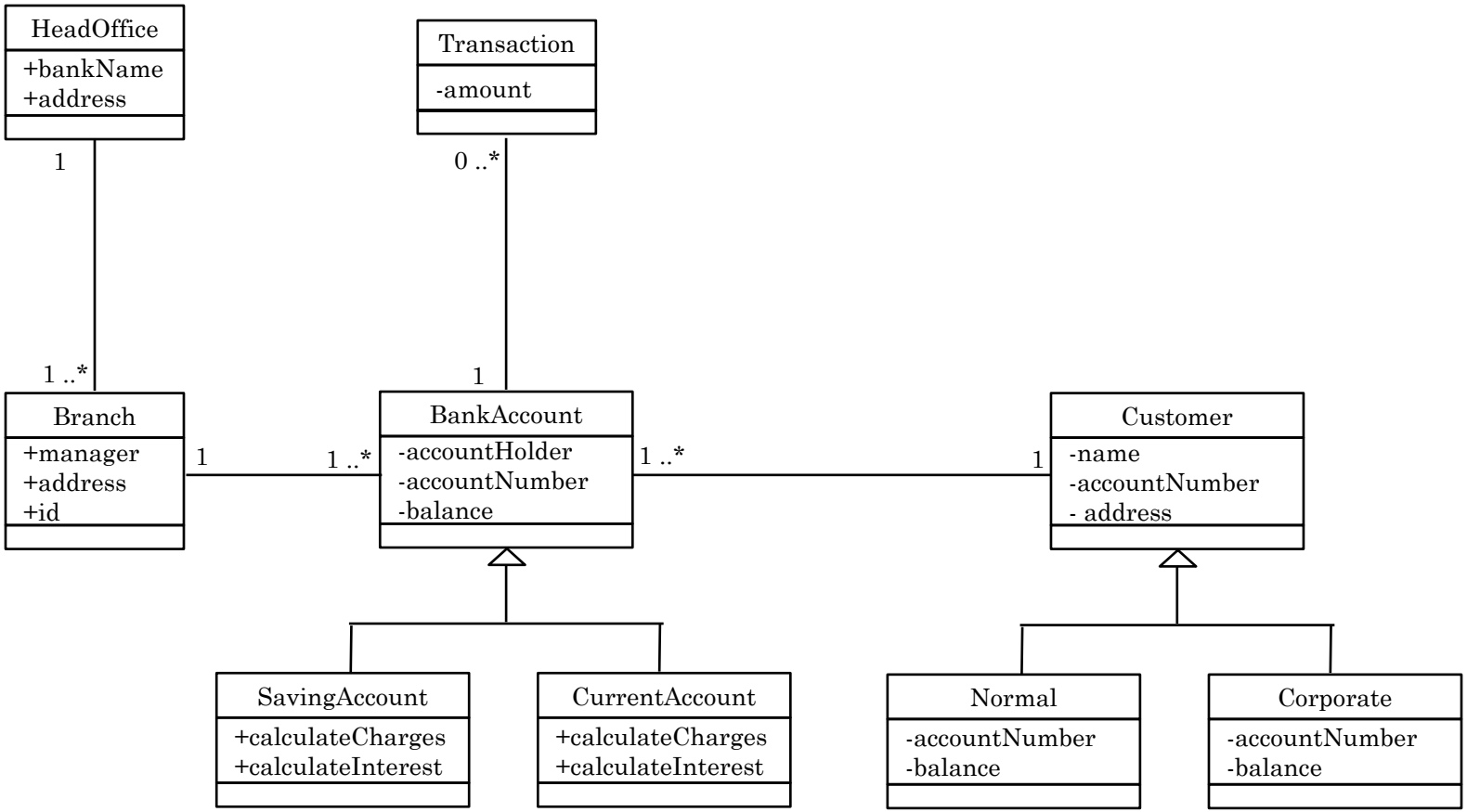
- Payment can be done through net banking and credit/debit card
- Customer may be a company or an ordinary person



Assignment: Banking System

- Possible classes
  - Head office
  - Branch
  - Bank account
- Try to think few more
- Use generalization and association wherever necessary

Solution: Banking System



THANK YOU