UML Diagram

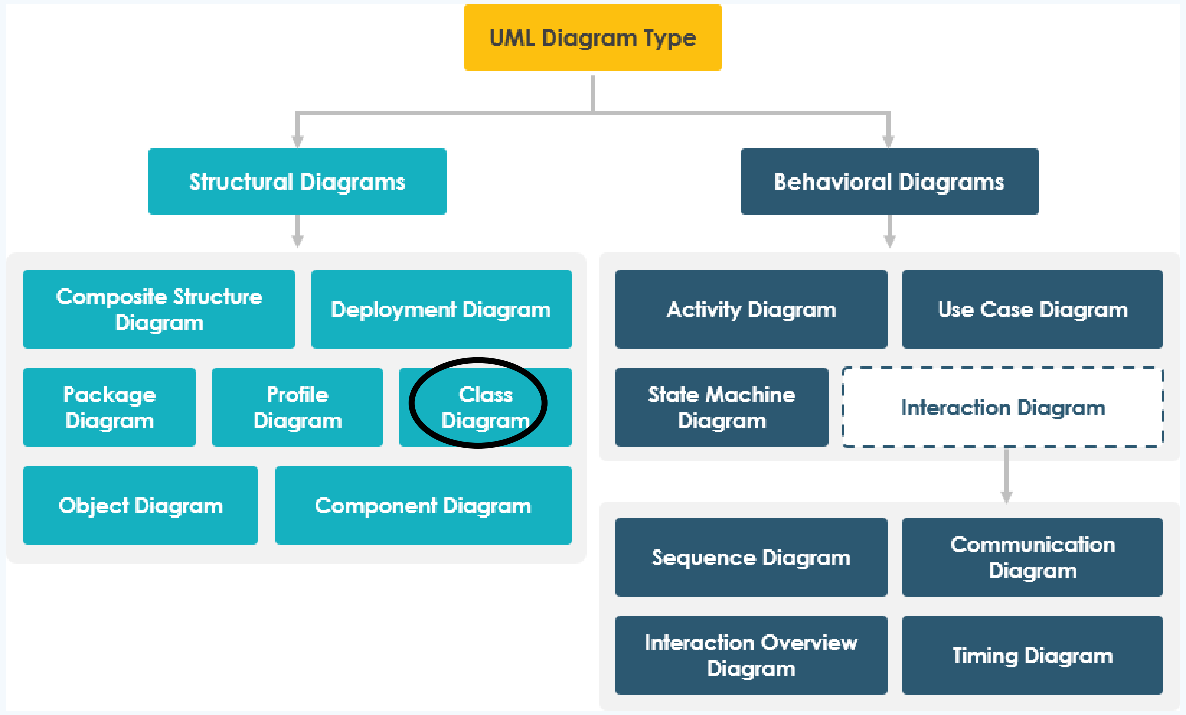
Prof.Anita Agrawal,
BITS-Pilani, k.k.Birla Goa Campus

UML: Unified Modeling Language

- UML is a modern approach to modeling and documenting software. In fact, it’s one of the most popular business process modeling techniques.
- It is a graphical language that is standard to the software industry for specifying, visualizing, constructing and documenting the artifacts of the software systems.
- **Types of UML:**
  - **Behavioral** UML diagram and
  - **Structural** UML diagram.

Types of UML Diagrams

- Structural UML Diagram: used to analyze and depict the structure of a system or process.
- Behavioral UML Diagram: used to describe the behavior of the system, its actors, and its building components.



Class Diagram

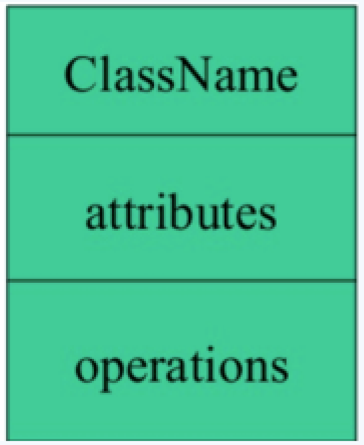
- It gives the static view of an application.
- It gives an overview of a software system by displaying classes, attributes, operations, and their relationships.
- **Class encapsulates all the relevant data of a particular object in a very systematic and clear way.**
- A class diagram is a collection of classes and interfaces

Benefits of class diagram

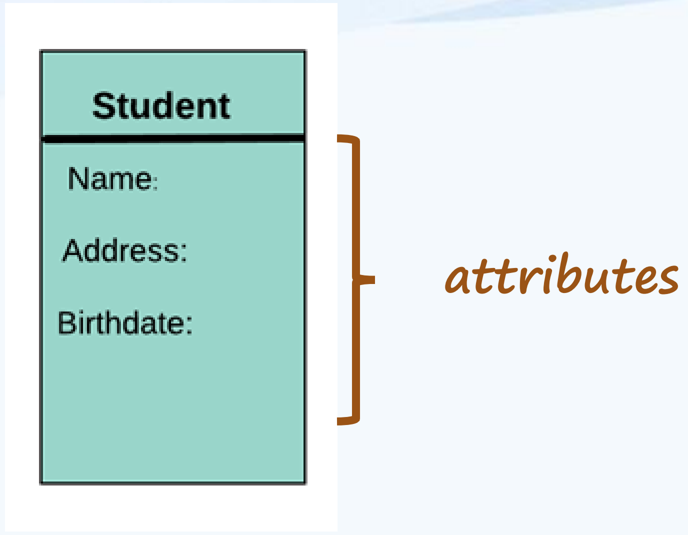
- Class Diagram illustrates data models for even very complex information systems
- It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time
- It helps for better understanding of general schematics of an application.
- Allows drawing detailed charts which highlight code required to be programmed
- It helps construct the code for the software application development.
- Helpful for developers and other stakeholders.

Essential elements of a UML class diagram

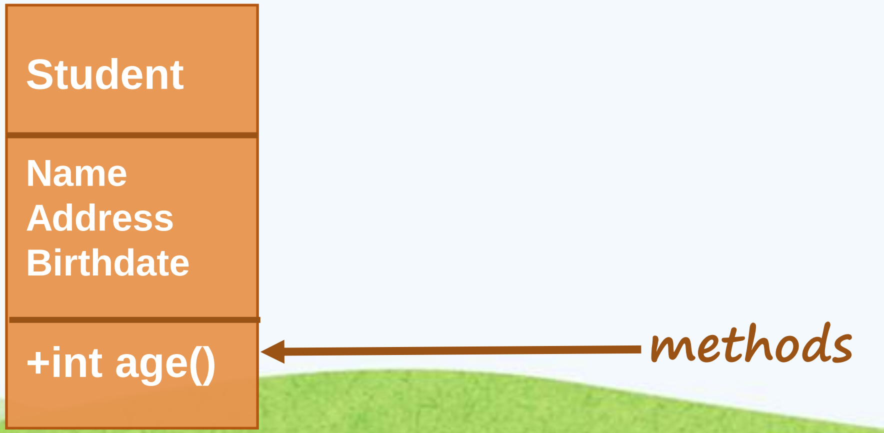
- Class Name
- Attributes
- Operations/Methods
- Relationships among objects



Attributes: Describe the variables used in a class



Operations: Describe the methods that are used in a class

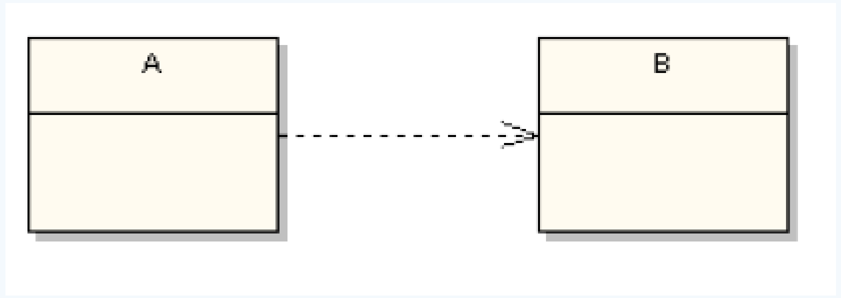


Relationships in class diagrams

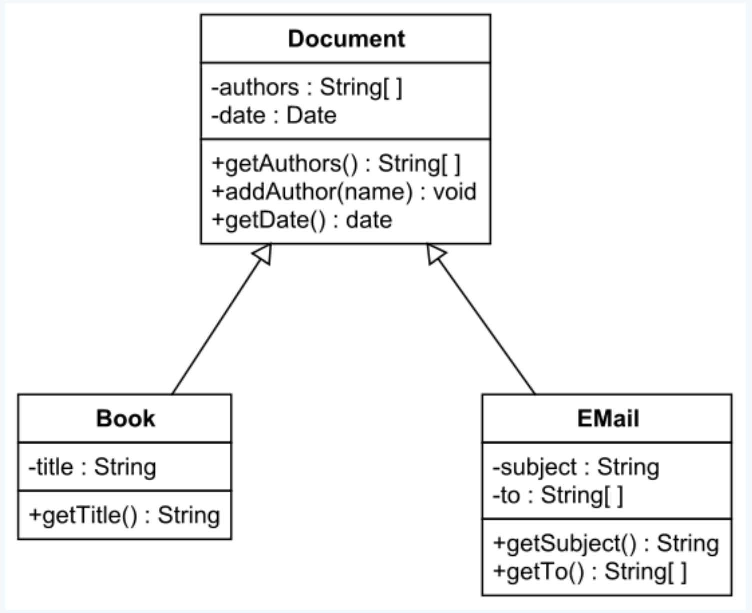
- Classes are interrelated to each other in specific ways.
  - There are mainly three kinds of relationships in UML:
    - Dependency
    - Generalization/Inheritance
    - Associations
- Relationships in class diagrams include different types of logical connections.

Relationships in class UML Diagram

- Dependency:
- An object of one class might use an object of another class in the code of a method. If the object is not stored as a class variable, then this is modeled as a dependency relationship.
- Exists between two classes if changes to the definition of one may cause changes to the other (but not the other way around).
- Class1 depends on Class2



Generalization/Inheritance



A class extends another class.

Example: The Book class might extend the Document class, which also might include the Email class. The Book and Email classes inherit the fields and methods of the Document class (possibly modifying the methods), but might add additional fields and methods.

Association

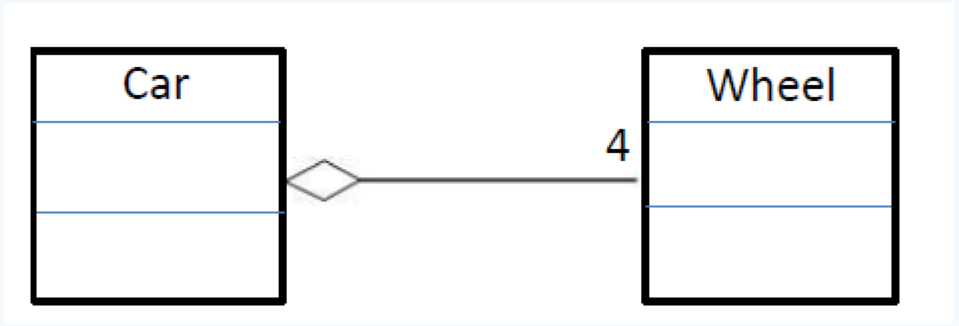
- Most often represent instance variables that hold references to other objects (**has-a** relationship)
- Two types:
  - Aggregation
  - Composition

Aggregation is a special type of association that models a whole- part relationship between aggregate and its parts.

The composition is a special type of aggregation which denotes strong ownership between two classes when one class is a part of another class.

Aggregation	*Composition*
Aggregation indicates a relationship where the child can exist separately from their parent class.	Composition display relationship where the child will never exist independent of the parent.

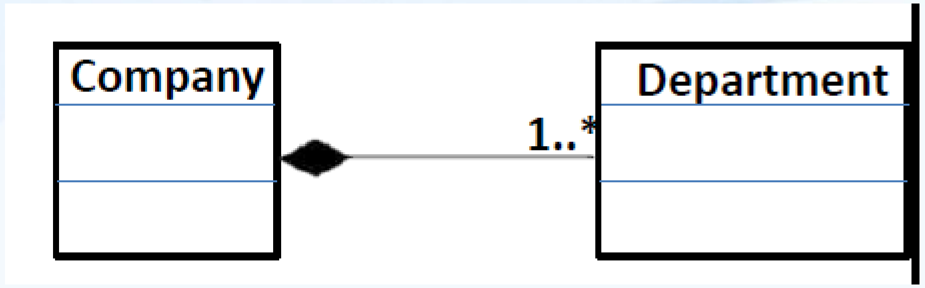
Aggregation



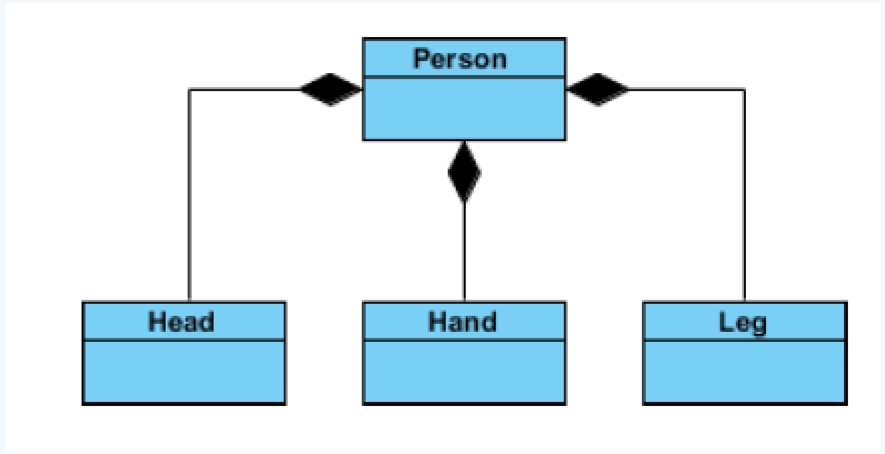
Real world example:
–*Car* as a **whole** entity and *Wheel* as **part** of the overall Car

does imply ownership, but shared
•Wheels can be removed from one car and can be attached to another car

Composition



Real world example: Company and departments



Human and Organs

Visibility and its symbols in UML

-	private
#	protected
+	public
~	package

BEST OF LUCK!!! ☺