CS5002NI

Software Engineering

Recap

- Use Case Diagram
- Sequence Diagram
- Collaboration Diagram

Agendas

- Class Diagram
- Purpose of Class Diagram
- Notations of Class Diagram
- Relationships used in Class Diagram

Class Diagram

- Structure diagram for designing and modeling software
- Models software in a high level abstraction and without having to look at the source code
- Corresponds with classes in the source code

Class Diagram

- Shows names and attributes of a class, connection between the classes and sometimes also the methods of the class

Purpose of a Class Diagram

- Provides a sense of orientation, detailed insight into the structure of the system
- Quick overview of the synergy happening among different system elements as well as their properties and relationships

Purpose of a Class Diagram

- Based on the principle of Object Orientation and can be implemented in various phases
  - Analysis -> domain model
  - Design -> model software
  - Implementation -> generate source code

Purpose of a Class Diagram

- Static view of the elements
- Blueprint for a building or a piece of machinery, see the parts used to make it and how they are assembled
- Cannot see how they behave when they are set in motion

Elements of a Class Diagram

- A set of classes
- A set of relationships between classes

Classes

- A description of a group of objects all with similar roles in the system, which consists of a structural and behavioural feature

Classes - Structural Features

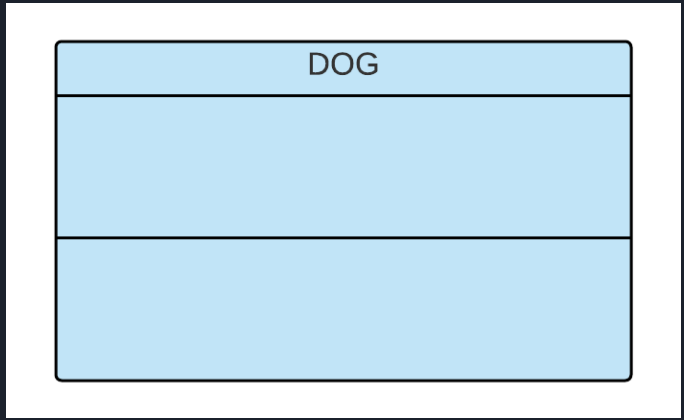
- Define what objects of the class knows
- Attributes of a class

Classes - Behavioural Features

- Define what objects of the class can do
- Methods of a class

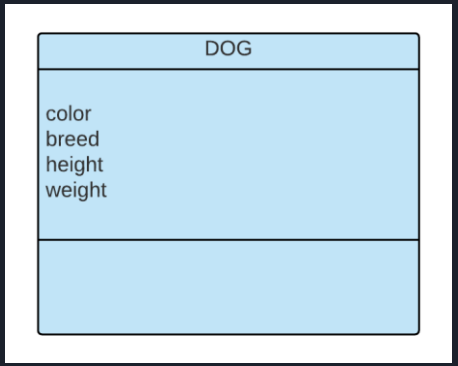
Class - Representation

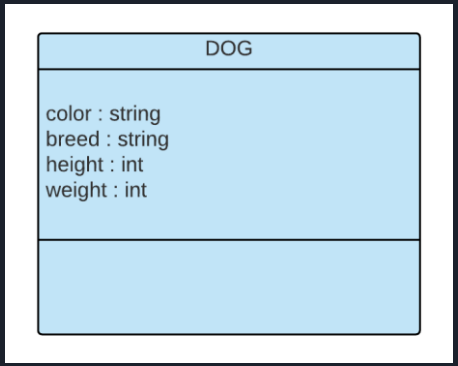
- A box separated into three parts
- Class name in first partition

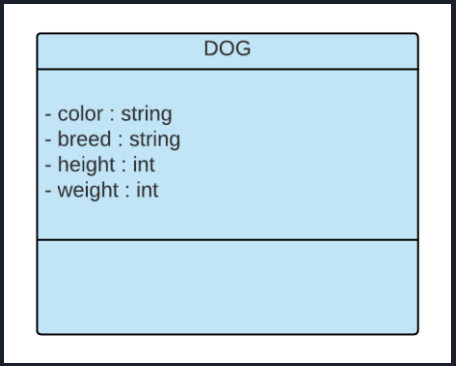


Class - Representation

- Attribute name in second partition

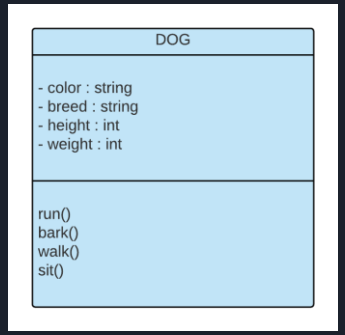


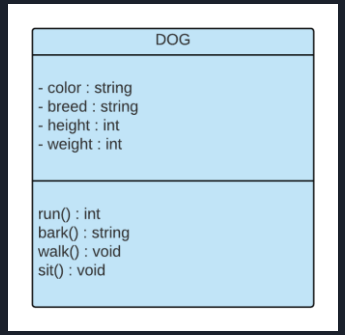


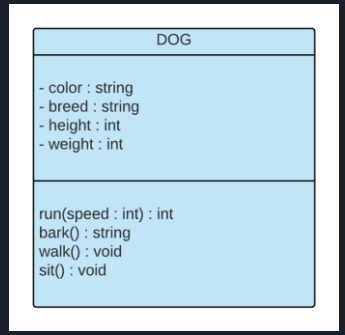


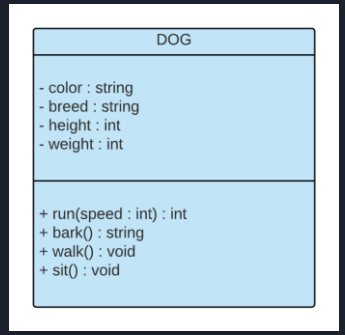
Class - Representation

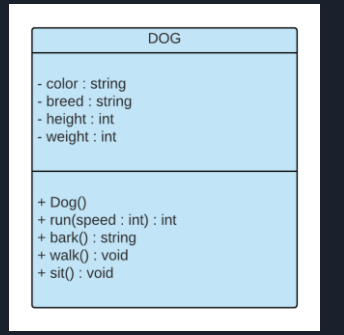
- Method name in third partition











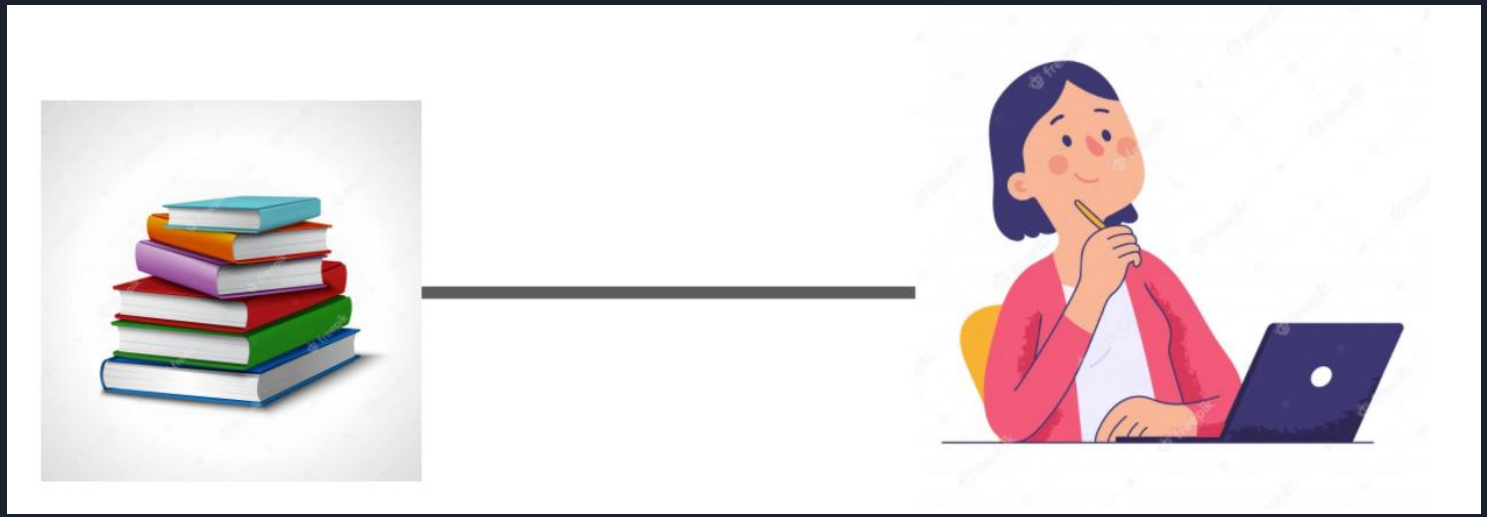
Relationships

- Association
- Inheritance
- Aggregation
- Composition
- Dependency

Association

- Any relationship between classes
- Represented as a solid line connecting two classes

Association

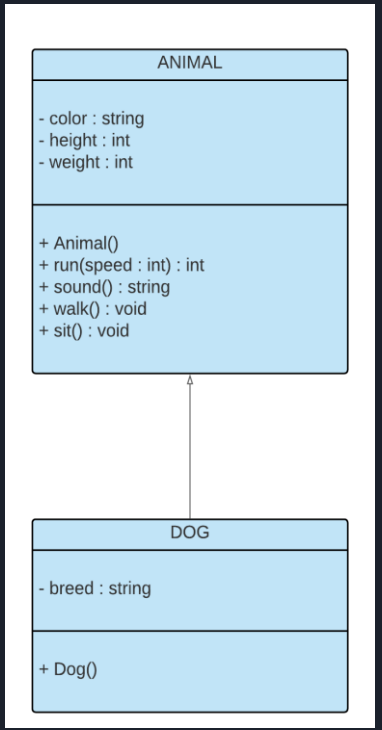


Inheritance

- Generalization
- Parent-child relationship
- Described by a solid line with a hollow arrowhead that points from the child to the parent class

Inheritance

- Child class inherits properties from of the parent class



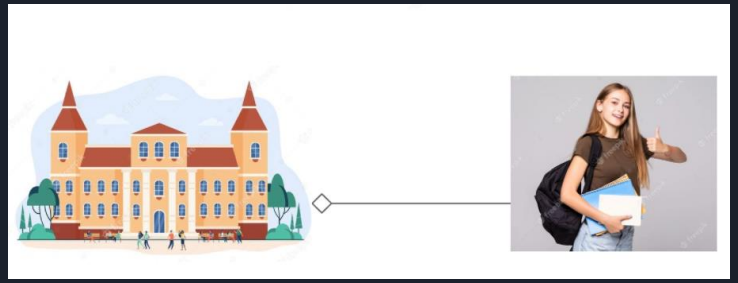
Aggregation

- Represents ‘has a’ relationship
- More specific than a association relationship as it defines a part-whole or a part-of relationship

Aggregation

Aggregation

- Denoted by a solid line with an unfilled diamond.



Aggregation

- Child can exist independently of its parent

Composition

- Subset of aggregation
- Represents a whole-part relationship, portrays the dependency between the parent and the child
- If one is deleted, the other one is discarded

Composition

- Solid line with a filled diamond

Dependency

- Semantic relationship between two or more classes
- A change in one class causes changes in others
- Weakest relationship

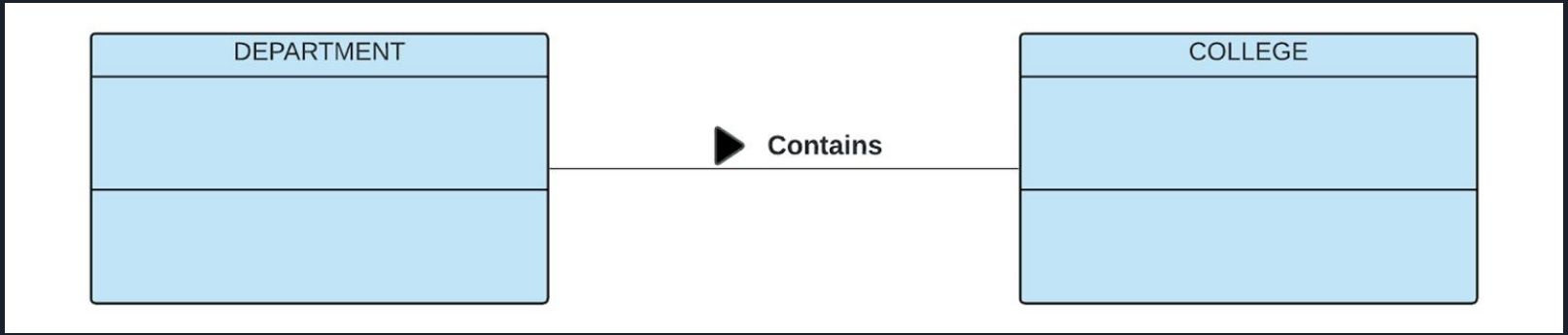
Dependency

- Denoted by a dashed line with an arrow towards the dependent class

Class Diagram - Roles

- Directional purpose of a class
- Written at the end of a relationship line, and describe the purpose played by the class in the relationship
- Represented by a verb prefixed by a filled triangle

Class Diagram - Roles



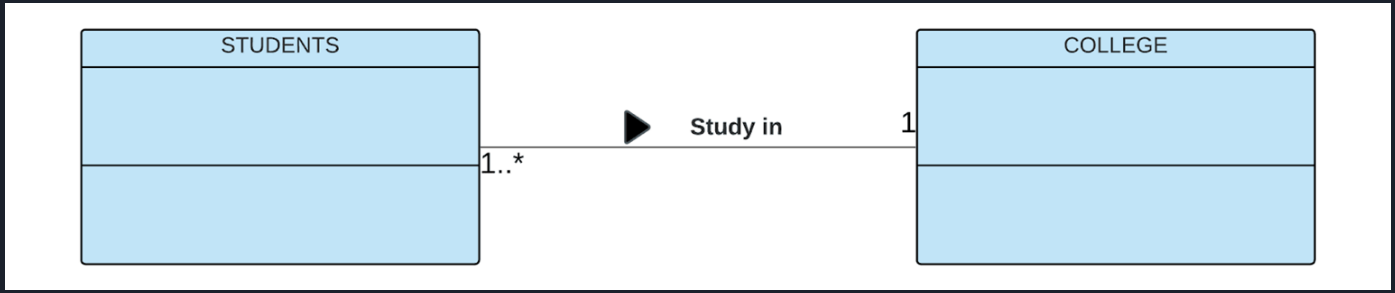
Class Diagram - Multiplicity

- Factor associated with an attribute, meaning it specifies how many instances of attributes are created when a class is initialized

Class Diagram - Multiplicity

- They can expressed in the following ways:
  - Exactly one - 1
  - Zero or one - 0..1
  - Many - 0..* or *
  - One to many - 1..*
  - exact number of instances - 1, 3, 6
- Default multiplicity is exact one

Class Diagram - Multiplicity



THANK YOU!