



## Department of Information Technology

**COURSE CODE: DJS22ITL601**

**DATE: 4 May 2025**

**COURSE NAME: Software Engineering Laboratory**

**CLASS: T.Y.BTech**

### EXPERIMENT NO.4

**CO/LO** Analyze real world problem using software engineering principles.

**AIM / OBJECTIVE:** Draw Class diagram for the chosen problem statement

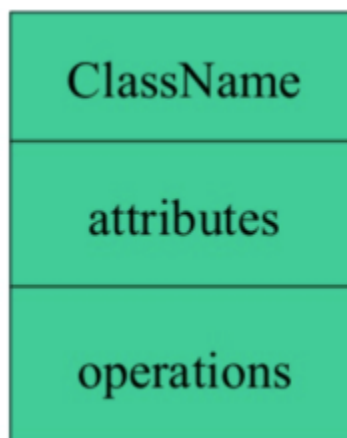
#### DESCRIPTION OF EXPERIMENT:

#### CLASS DIAGRAM

A class diagram is a type of UML (Unified Modeling Language) diagram that represents the static structure of a system by showing the classes, their attributes, methods, and relationships between them. It is one of the most commonly used types of UML diagrams and is typically used during the early stages of software development to help understand the requirements of the system being developed.

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations





Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## Department of Information Technology

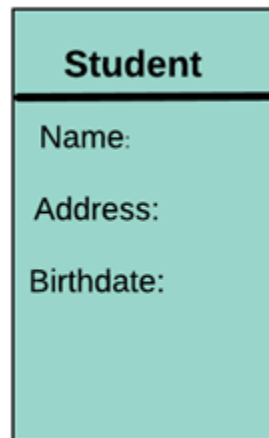
The name of the class is only needed in the graphical representation of the class. It appears in the topmost compartment. A class is the blueprint of an object which can share the same relationships, attributes, operations, & semantics. The class is rendered as a rectangle, including its name, attributes, and operations in sperate compartments.

Following rules must be taken care of while representing a class:

4. A class name should always start with a capital letter.
5. A class name should always be in the center of the first compartment.
6. A class name should always be written in **bold** format.
7. UML abstract class name should be written in italics format.

### Attributes:

An attribute is named property of a class which describes the object being modeled. In the class diagram, this component is placed just below the name-compartment.



A derived attribute is computed from other attributes. For example, an age of the student can be easily computed from his/her birth date.



Shri Vile Parle Kelavani Mandal's

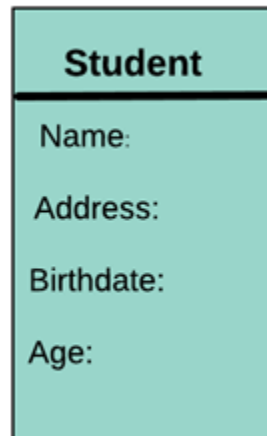
**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## Department of Information Technology



### Attributes characteristics

- The attributes are generally written along with the visibility factor.
- Public, private, protected and package are the four visibilities which are denoted by +, -, #, or ~ signs respectively.
- Visibility describes the accessibility of an attribute of a class.
- Attributes must have a meaningful name that describes the use of it in a class.

### Relationships

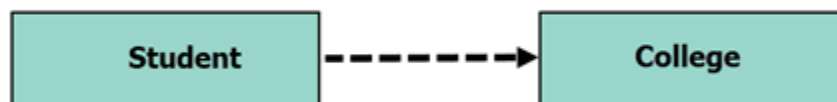
There are mainly three kinds of relationships in UML:

8. Dependencies
9. Generalizations
10. Associations

### Dependency

A dependency means the relation between two or more classes in which a change in one may force changes in the other. However, it will always create a weaker relationship. Dependency indicates that one class depends on another.

In the following UML class diagram examples, Student has a dependency on College





Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

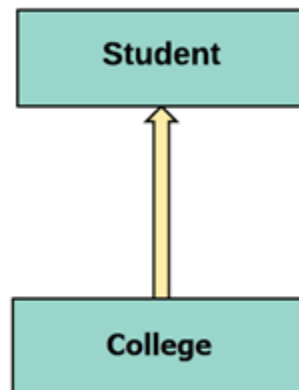
(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## Department of Information Technology

### Generalization:



A generalization helps to connect a subclass to its superclass. A sub-class is inherited from its superclass. Generalization relationship can't be used to model interface implementation. Class diagram allows inheriting from multiple superclasses.

In this example, the class Student is generalized from Person Class.

### Association:

This kind of relationship represents static relationships between classes A and B. For example; an employee works for an organization.

Here are some rules for Association:

- Association is mostly verb or a verb phrase or noun or noun phrase.
- It should be named to indicate the role played by the class attached at the end of the association path.
- Mandatory for reflexive associations

In this example, the relationship between student and college is shown which is studies.



Shri Vile Parle Kelavani Mandal's

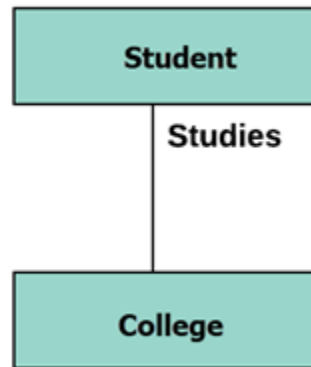
**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

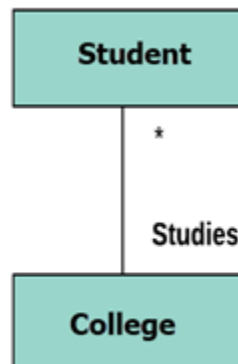
NAAC Accredited with "A" Grade (CGPA : 3.18)



## Department of Information Technology



### Multiplicity



A multiplicity is a factor associated with an attribute. It specifies how many instances of attributes are created when a class is initialized. If a multiplicity is not specified, by default one is considered as a default multiplicity.

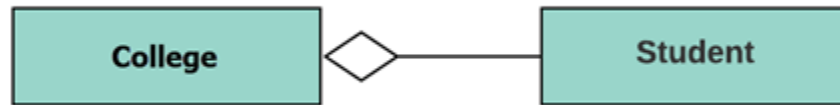
Let's say that there are 100 students in one college. The college can have multiple students.

### Aggregation

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



## Department of Information Technology



For example, the class college is made up of one or more student. In aggregation, the contained classes are never totally dependent on the lifecycle of the container. Here, the college class will remain even if the student is not available.

### Composition:



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

For example, if college is composed of classes student. The college could contain many students, while each student belongs to only one college. So, if college is not functioning all the students also removed.

### Aggregation vs. Composition

Aggregation	Composition
Aggregation indicates a relationship where the child can exist separately from their parent class. Example: Automobile (Parent) and Car (Child). So, If you delete the Automobile, the child Car still exist.	Composition display relationship where the child will never exist independent of the parent. Example: House (parent) and Room (child). Rooms will never separate into a House.

### Abstract Classes

It is a class with an operation prototype, but not the implementation. It is also possible to have an abstract class with no operations declared inside of it. An abstract is useful for identifying the functionalities across the classes. Let us consider an example of an abstract class. Suppose we have an abstract class called as a motion with a method or an operation declared inside of it. The method declared inside the abstract class is called a **move ()**.



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)

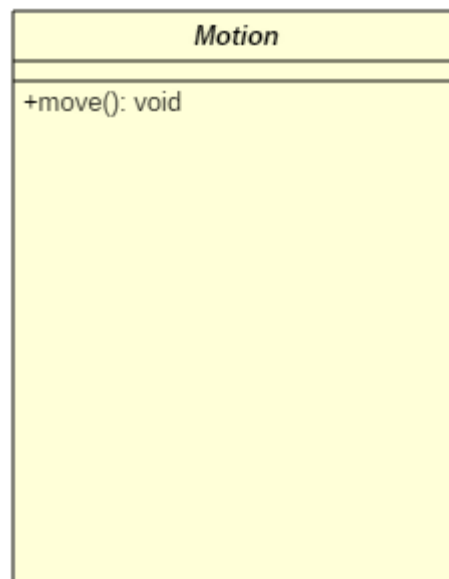


## Department of Information Technology

This abstract class method can be used by any object such as a car, an animal, robot, etc. for changing the current position. It is efficient to use this abstract class method with an object because no implementation is provided for the given function. We can use it in any way for multiple objects.

In UML, the abstract class has the same notation as that of the class. The only difference between a class and an abstract class is that the class name is strictly written in an italic font.

An abstract class cannot be initialized or instantiated.



Abstract Class Notation

In the above abstract class notation, there is the only a single abstract method which can be used by multiple objects of classes.

## QUESTIONS

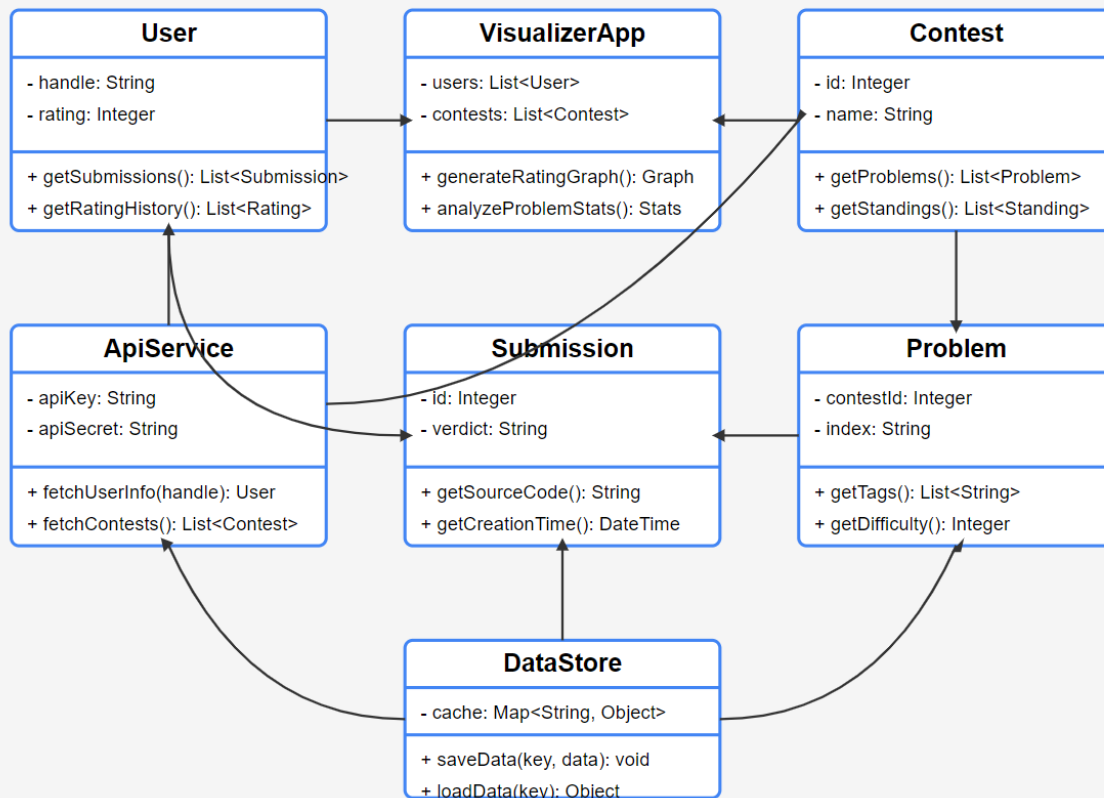
1. What is a UML class diagram, and what is its purpose?
2. How do you show dependencies between classes in a UML class diagram?

## OBSERVATIONS / DISCUSSION OF RESULT:



## Department of Information Technology

### Codeforces Visualizer Class Diagram



This section should interpret the outcome of the experiment. The observations can be visually represented using images, tables, graphs, etc. This section should answer the question "What do the result tell us?" Compare and interpret your results with expected behavior. Explain unexpected behavior, if any.

### CONCLUSION:

Base all conclusions on your actual results; describe the meaning of the experiment and the implications of your results.

### REFERENCES:

#### Website References:

1. [The UML 2 class diagram - IBM Developer](#)





Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## **Department of Information Technology**