**Practical no :- 06**

**AIM :** Modeling UML class diagram

**Introduction**

Classes are the structural units in object oriented system design approach, so it is essential to know all the relationships that exist between the classes, in a system. All objects in a system are also interacting to each other by means of passing messages from one object to another. Sequence diagram shows these interactions with time ordering of the messages.

UML Class Diagram

A UML class diagram procedure involves defining the classes within a system, their attributes and methods, and the relationships between them. It's a visual representation of the static structure of a system, showing how classes are organized and interact.

**Structural and Behavioral aspects**

Developing a software system in object oriented approach is very much dependent on understanding the problem. Some aspects and the respective models are used to describe problems and in context of those aspects the respective models give a clear idea regarding the problem to a designer. For developer, structural and behavioral aspects are two key aspects to see through a problem to design a solution for the same.

**Class diagram**

It is a graphical representation for describing a system in context of its static construction.

**Elements in class diagram**

Class diagram contains the system classes with its data members, operations and relationships between classes.

**Class**

A set of objects containing similar data members and member functions is described by a class. In UML syntax, class is identified by solid outline rectangle with three compartments which contain

* **Class name** A class is uniquely identified in a system by its name. A textual string is taken as class name. It lies in the first compartment in class rectangle.
* **Attributes** Property shared by all instances of a class. It lies in the second compartment in class rectangle.
* **Operations** An execution of an action can be performed for any object of a class. It lies in the last compartment in class rectangle.

**Example**

To build a structural model for an Educational Organization, ‘Course’ can be treated as a class which contains attributes ‘courseName’ & ‘courseID’ with the operations ‘addCourse()’ & ‘removeCourse()’ allowed to be performed for any object to that class.

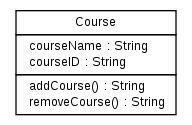
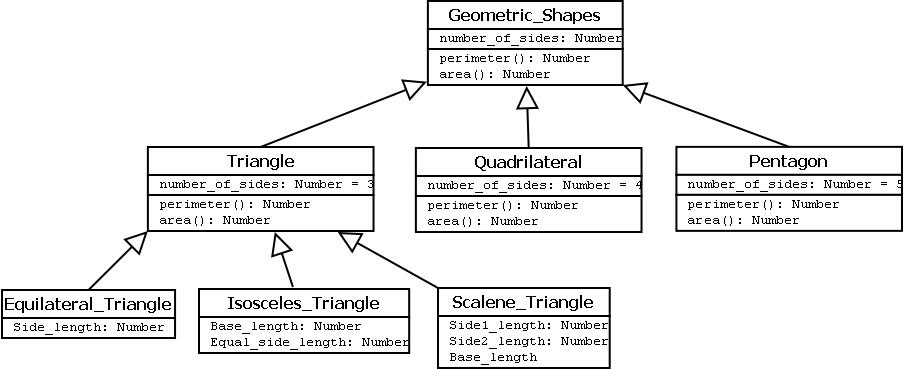


Figure - 1

* **Generalization/Specialization** It describes how one class is derived from another class. Derived class inherits the properties of its parent class.

**Example**



Procedure

1. **Define the Scope and Purpose:** Clearly identify the system or subsystem you're modeling and its purpose.
2. **Identify Classes:** Determine the main entities and concepts within the system that will be represented as classes.
3. **Add Attributes:** For each class, list the data fields (attributes) that describe its properties.
4. **Add Methods:** Identify the operations (methods) that the class can perform.
5. **Show Relationships:** Use lines and arrowheads to represent the connections between classes, indicating associations, inheritance, aggregation, etc.
6. **Add Multiplicity:** If necessary, use notation to specify the number of instances of a related class that can be associated with another class.
7. **Review, Refine, Share, Iterate:** Continuously refine the diagram as your understanding of the system evolves.