Lab report no: 04

Name of the lab report: Class diagram.

Group members:

- **IT-17015**
- **IT-17032**
- **IT-17045**
- **IT-17057**
- **IT-17060**

Objectives:

- Learn basic about class diagram
- Learn class diagram notations
- Learn how to draw an class diagram.

Theory:

What is an class diagram?

UML CLASS DIAGRAM gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments. Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

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 highlights code required to be programmed

- Helpful for developers and other stakeholders.

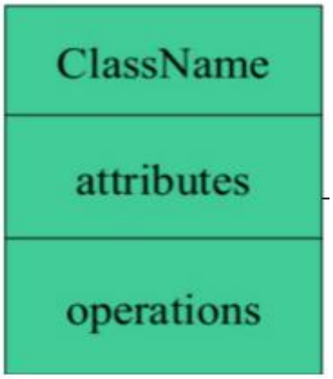
Essential elements of A UML class diagram

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations

Class Name:

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.

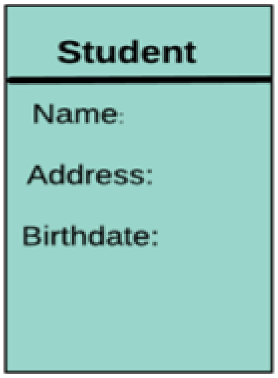


Following rules must be taken care of while representing a class:

1. A class name should always start with a capital letter.
2. A class name should always be in the center of the first compartment.
3. A class name should always be written in bold format.
4. An 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.



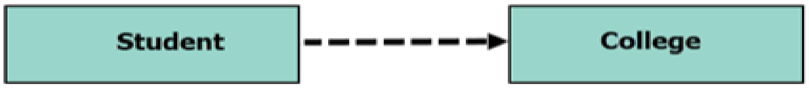
Relationships

There are mainly three kinds of relationships in UML:

1. Dependencies
2. Generalizations
3. Associations

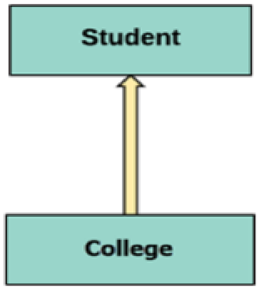
Dependency

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 example, Student has a dependency on College.



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.

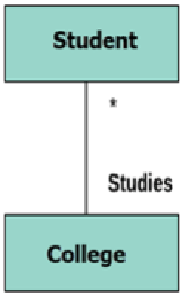


Association:

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

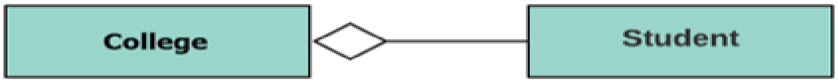
Multiplicity

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.



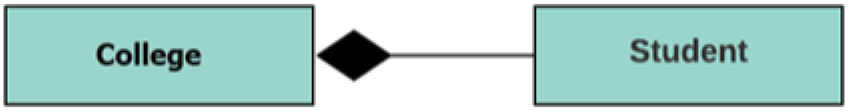
Aggregation

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

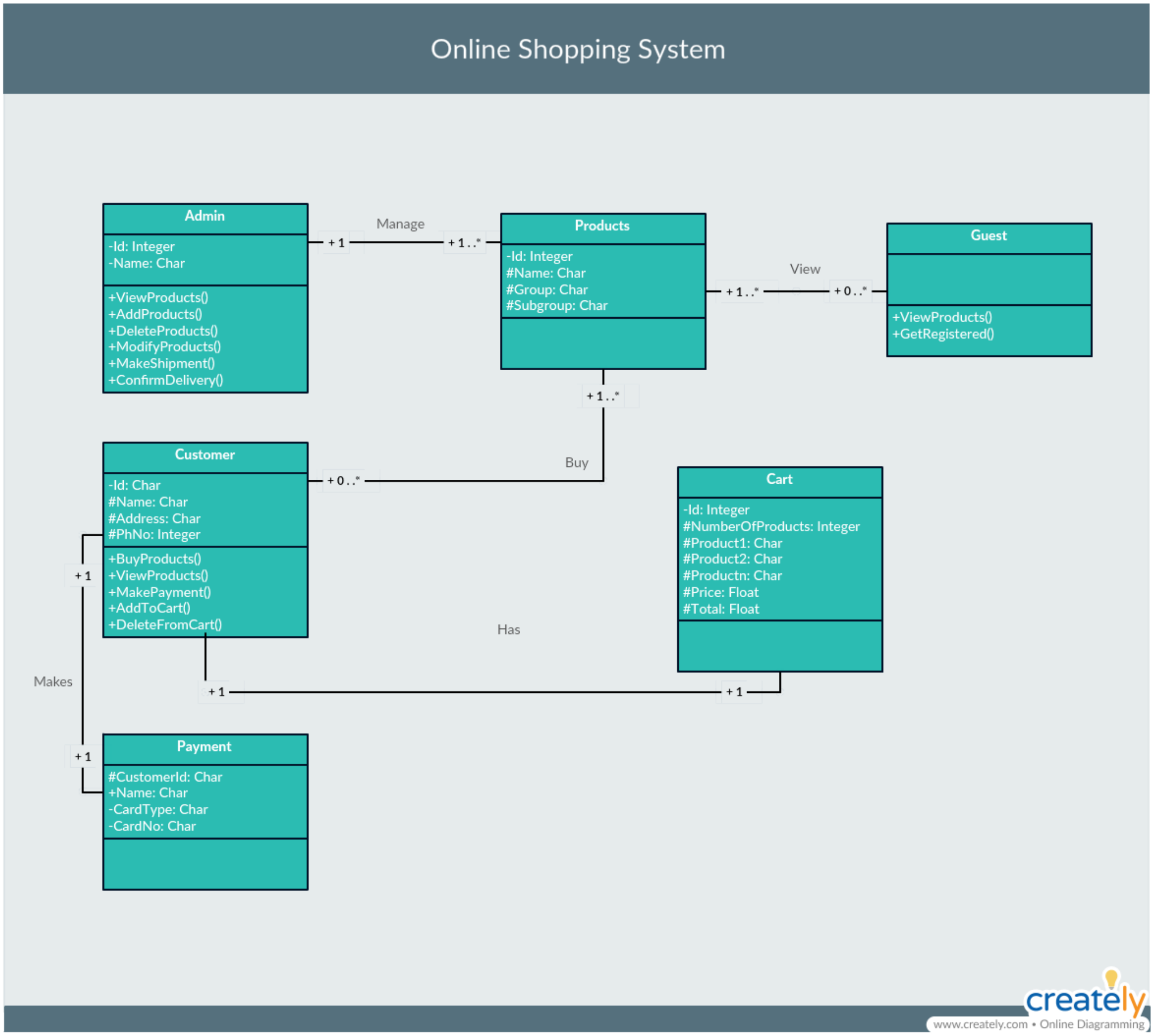


Composition

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



Example of class Diagram:



Our group story:

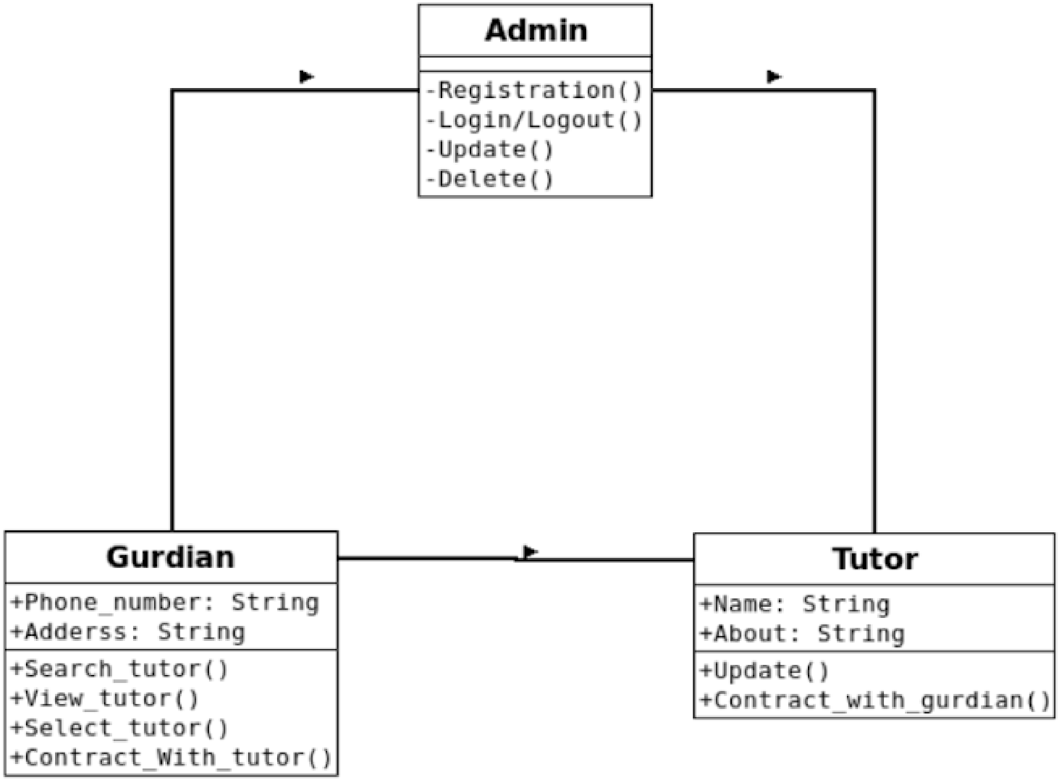
X is a student of Y university. He belongs to a middle class family. When he at first came to university, he faces a lot of trouble specifically financial problems. So he collects tuition from his varsity's senior brother. After getting the tuition somehow he maintains his cost. But he thinks that the problem doesn't over. He notices that when students at first came to university they are also facing the same problems who are also belonging to middle class family, they are also facing a lot of financial problems. So try to do something about this problem. At third year second semester his teacher teaches them a course about software engineering. In this course he learns laraval software development framework and decides to develop a software on tuition management system using laravel framework. To develop such a system he first want to draw the following diagram of the system.

- Activity diagram.

- Use Case diagram.
- Sequence diagram.
- Class diagram.

In this lab we will draw a class diagram based on that story

Class diagram of our group story:



Conclusion:

This was an interesting lab. I learned many things from this lab. I don't know nothing about class diagram before this lab work but now I know about class diagram, the main components of class diagram and finally how to draw a class diagram.