CSc 322 - Software Engineering

Lab – Fall 2017

Teaching Assistant: Isatou Sanneh

Email: isanneh00@citymail.cuny.edu

UML Class Diagrams

October 13th, 2017

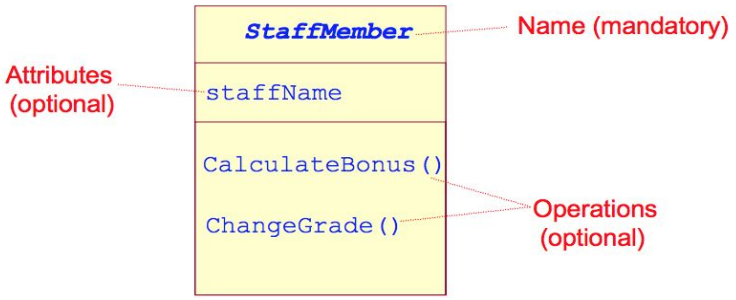
Classes are the structural units in object oriented system design approach, so it is essential to know all the relationships that exist between the classes in a system. They are used to show the different objects in a system, their attributes, their operations and the relationships among them.

Class diagram

Elements in a 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 a 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/row of the class rectangle
- *Attributes:* Property shared by all instances of a class. It lies in the second compartment/row of the class rectangle
- *Operations:* An execution of an action can be performed for any object **in** a class. It lies in the last compartment/row of the 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 in that class

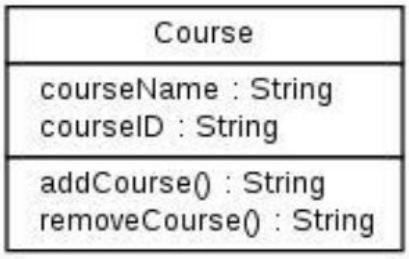


Figure 1: Simple Class Diagram for a Course

- *Generalization/Specialization:* It describes how one class is derived from another class. A derived class inherits the properties of its parent class
  Example: Geometric_Shapes is the class that describes how many sides a particular shape has. Triangle, Quadrilateral, and Pentagon are the classes that inherit the property of the Geometric_ Shapes class. So the relations among these classes are generalization. Now Equilateral_ Triangle, Isosceles_Triangle and Scalene_Triangle classes inherit the properties of Triangle class as each one of them has three sides. Therefore, these are specializations of the Triangle class.

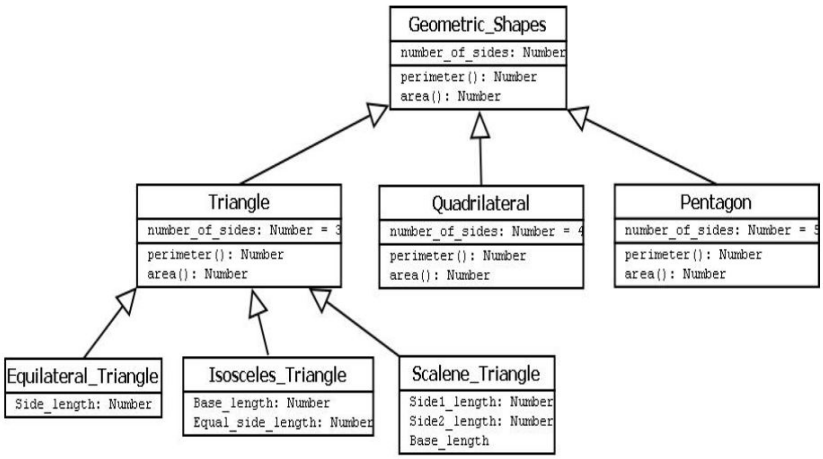


Figure 2: Generalization/Specialization: Shapes

Relationships

Existing relationships in a system describe legitimate connections between the classes in that system

- *Association*
  It is used to show any logical connection or relationship between classes.
  An association between two classes is depicted by a straight line connecting the two class boxes.
  Example: In the structure model for a system of an organization, an employee (instance of 'Employee' class) is always assigned to a particular department

(instance of 'Department' class) and the association can be shown by a line connecting the respective classes

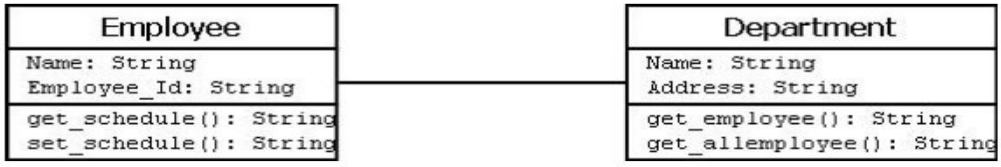


Figure 3: Relationship (Association): Employee-Department

- *Aggregation*
  It is a special form of association that depicts the relationship between a single class (child) and an aggregate class (parent) made up of a collection of the same single classes.
  Aggregation is a special type of association used to model a "whole to its parts" relationship. In basic aggregation relationships, the lifecycle of a *part* class is independent from the *whole* class's lifecycle.
  In aggregation, the child class is not dependent on the lifecycle of the parent class. That is, a child class will not be destroyed when its parent class is destroyed.
  An aggregation can be depicted by drawing a line from the parent class to the child class with a diamond shape near the parent class.
  Example: For a supermarket in a city, each branch runs some of the departments they have. So, the relation among the classes 'Branch' and 'Department' can be designed as an aggregation.
  In UML, it can be shown as in the figure below

Figure 4: Relationship (Aggregation): Branch-Department

Example: The class "library" is made up of one or more books, among other materials. In aggregation, the child classes are not strongly dependent on the lifecycle of the parent. Therefore, books will remain so even when the library is dissolved.

Figure 5: Relationship (Aggregation): Library-Books

- *Composition*
  The composition relationship is similar to the aggregation relationship with the following exception: the life cycle of child class is dependent on the life cycle of the parent class. That is, the child class will be destroyed when the parent class is Destroyed.
  A composition can be depicted by using a directional line connecting the two classes, with a filled diamond shape adjacent to the parent class and the directional arrow to the child class
  Example: Let us consider a shopping mall that has several branches in different locations in a city. The existence of branches completely depends on the shopping mall - if the shopping mall doesn't exist, its branches will no longer exist in the city. This relation can be described as composition and can be shown as below

Figure 5: Relationship (Composition): Branch-Shopping Mall

- *Multiplicity*
  It describes how many number of instances of one class is related to the number of instances of another class in an association.
  Notations for different types of multiplicity
  # Single instance – 1
  # Zero or one instance – 0..1
  # Zero or more instance – 0..*

One or more instance – 1..*
Specific range (say, 2 to 6) – 2..6

Example: One vehicle may have two or more wheels

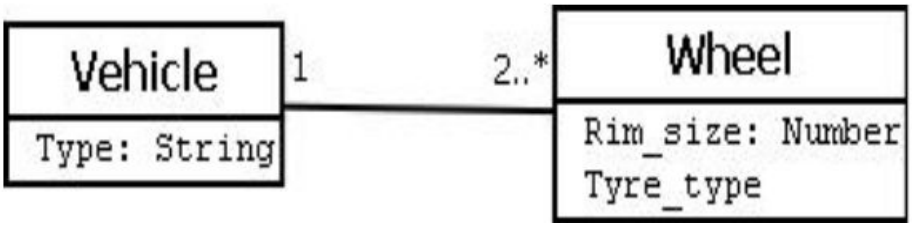
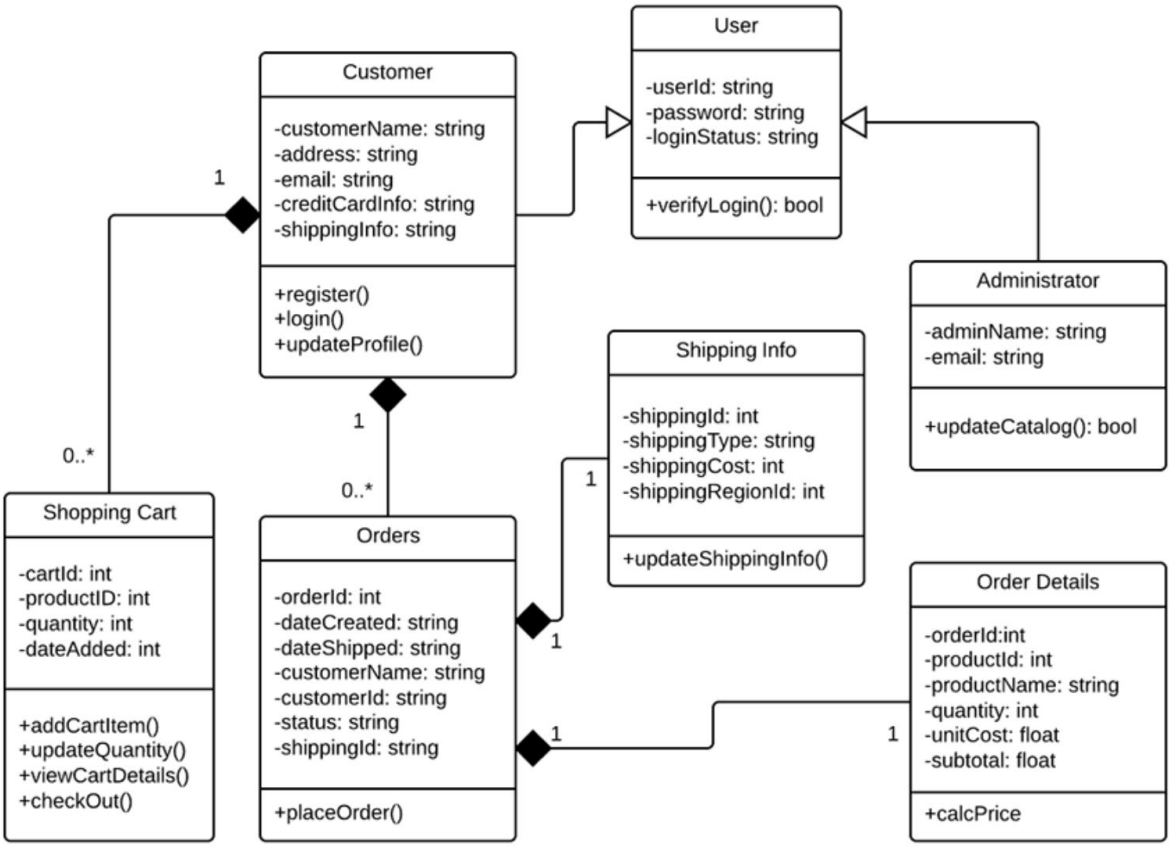


Figure 6: Multiplicity

Online Shopping Class Diagram



Practice Tasks

Part 1 Draw a class diagram for a University course application described below

For each course, maintain a list of the students on that course and the lecturer who has been assigned to teach that course. Allow options such as adding and removing of students from the course, assigning a teacher, getting a list of students currently assigned to a course, and the teacher currently assigned to teach the course. Teachers are modelled as Lecturer objects. A lecturer may teach more than one course. Each Lecturer object also maintains a list of the Courses that it teaches. Similarly, a course is attended by zero or more students, and a student may attend multiple courses.

Part 2 Class Diagram of a Web Browser

A web browser is a software that is used to access a resource (web page) available on the World Wide Web and identified by a URL. Whenever a user types in the URL of a web page in the browser's address bar and clicks the "Go" button, the browser sends a HTTP request to the concerned web server. If the requested resource is available and accessible, the web server sends back a HTTP response to the requesting web browser. In case of any error, a HTTP response is sent indicating the error.

When the web browser receives a HTTP response, it displays the web page to the user. In very simple terms, a web browser can be thought of consisting the following sub-components: rendering engine and browser control.

Once a HTTP response has been obtained from the server, the rendering engine decides the layout of the contents and actually displays the requested page. This is done keeping in mind the different HTML elements that are present in the page, and corresponding CSS rules, if any.

The browser control provides facilities like navigating across pages (by following hyperlinks), reloads a page, and handles other events related to the window display, for example, resizing the browser window.

Instruction: Represent classes with their state and behavior. Identify and represent association, aggregation, composition, and inheritance of classes as needed.

Part 3 Draw a class diagram for your team project (Coding Turk System)

******