E-JUST
الجامعة المصرية اليابانية للعلوم والتكنولوجيا
エジプト日本科学 技術大学

CSE 323 – Software Engineering

Lab 2 -Introduction Unified Modelling Language (UML)

Objectives

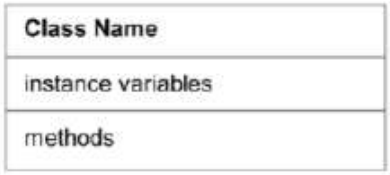
The students will be introduced to the Unified Modelling Language as a standard language independent modelling tool.

Part I – Introduction to UML

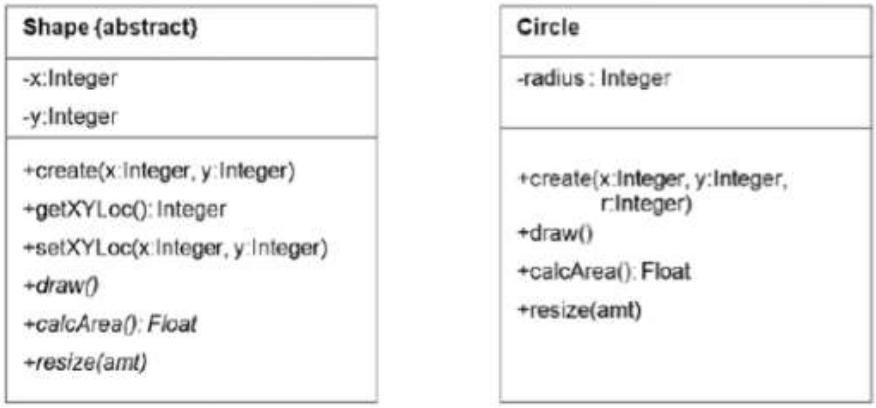
UML ("Unified Modeling Language") is a standardized language-independent, graphical modeling language for specifying an object-oriented design.

Class diagrams in UML are used to express the static aspects of an object-oriented design. Other diagrams, called **interaction diagrams**, are used to represent the sequence of method calls between objects during program execution.

A class is denoted in UML in three parts: a class name, a set of class attributes (instance variables), and a set of methods, as given below.



Initialization methods like `__init__` in Python are named create() in UML. The types of attributes (instance variables) and the return type of methods is indicated by :<type name>, for any given type (for example, :Integer). The + and − symbols are used to specify if a given member of a class has either public (+) or private (−) access.



UML Class Diagrams for `Shape` and `Circle` Classes

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. UML abstract class name should be written in italics format.

Relationships:

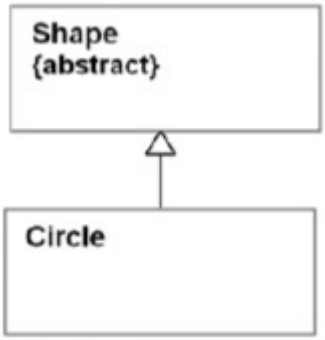
Associations

An **association** between two classes, denoted by a connecting solid line (and a possible arrow-head) indicates that methods of one class call methods of the other.

Association in UML

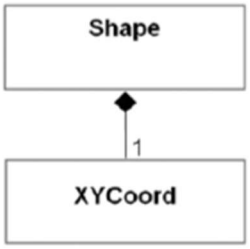
Sub-classes:

Subclass relationships in UML are indicated by use of a solid line with a closed arrow head from a subclass to a superclass.



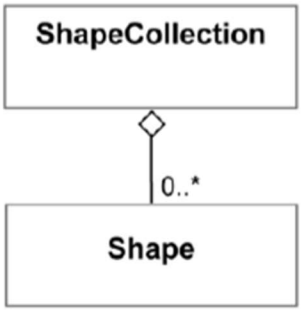
Composition vs Aggregation:

Composition is a "part of" relationship between classes denoted by a filled diamond head in UML. **Aggregation** is a "grouping" relationship, denoted by an unfilled diamond head.




```
def class XYCoord:

    def __init__(self, x, y):
        self.__xCoord = x
        self.__yCoord = y

    def getXY(self):
       return (self.__x, self.__y)

    def setXY(self, x, y):
        self.__xCoord = x
        self.__yCoord = y
```

Composition of Classes in UML

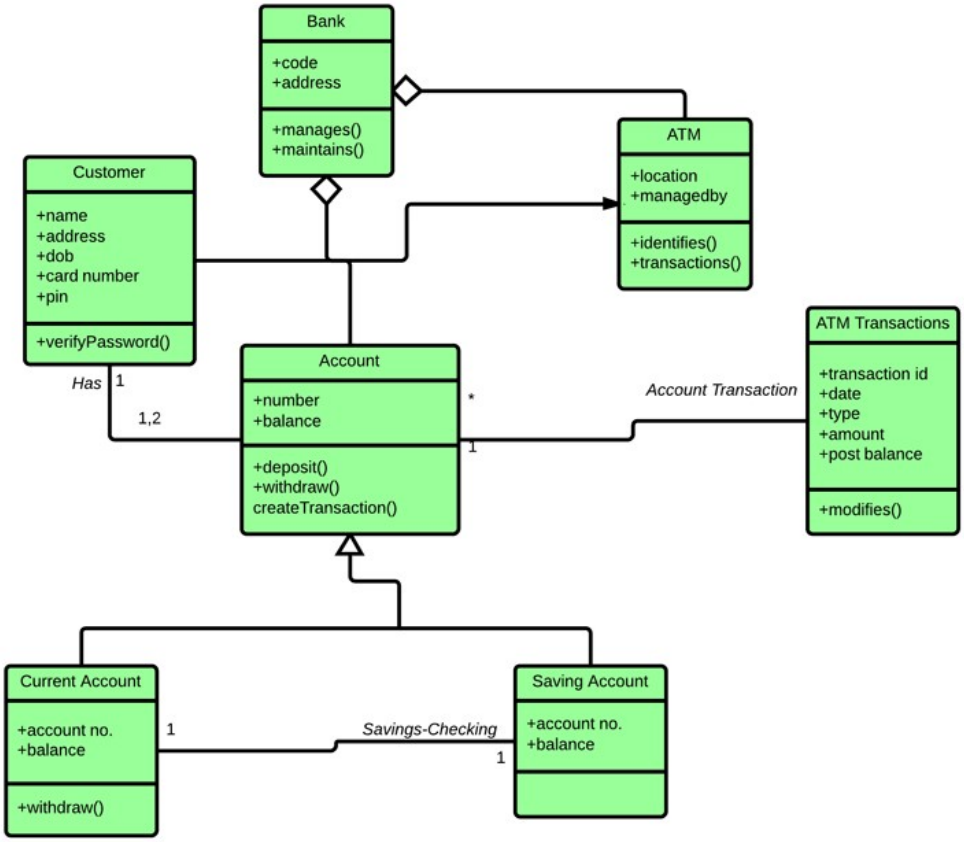



```
def class ShapeCollection:

    def __init__(self):
        self.__collection = []

    def add(self, shape):
        self.__collection.append(shape)

    def remove(self, shape):
        self.__collection.remove(shape)
```

Aggregation of Classes in UML

Example of Class Design

The following is an example of UML design for ATM class design:



source: https://www.guru99.com/uml-class-diagram.html

Part II – Exercise

2.1. Answer the following questions:

1. Which of the following is true of UML?
   (a) UML is a specification language for designing Python programs
   (b) UML is a specification language that can be used for designing programs in various programming languages

2. In UML, class diagrams are used to express the _____ aspects of a design, and _____ are used to denote the dynamic aspects

3. In UML, an association between two classes indicates that
   (a) The two classes have a common superclass
   (b) Objects of each of the two class types are created at the same time
   (c) Methods of one of the classes make calls to methods of the other

4. Multiplicity in UML indicates
   (a) How many objects of a given class type exist
   (b) How many objects of one given class there are in relation to another
   (c) How many subclasses of a given class there may be

5. Composition in UML indicates,
   (a) A "part of" relationship
   (b) A grouping of objects

6. Aggregation in UML indicates,
   (a) A "part of" relationship
   (b) A grouping of objects

2.2. Create UML diagram for the following problem

The problem is to develop an object-oriented UML modelling for a program capable of maintaining reservations for a vehicle rental agency. The agency rents out three types of vehicles—cars, vans, and moving trucks. The program should allow users to check for available vehicles, request rental charges by vehicle type, get the cost of renting a particular type vehicle for a specified period of time, and make/cancel reservations.

Requirements:

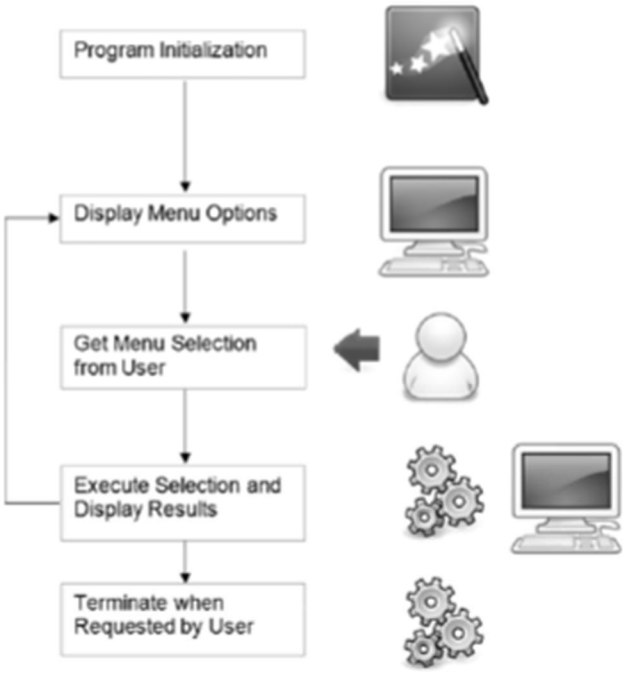
The program must maintain a group of specific model vehicles for the following vehicle categories: cars, vans, and (moving) trucks with the following characteristics:

Cars: make/model, miles-per-gallon, num of passengers, num of doors, VIN
Vans: make/model, miles-per-gallon, number of passengers, VIN
Trucks: miles-per-gallon, length, number of rooms, VIN

The program must be able to display the specific vehicles available for rent by vehicle type.

The program must display the cost associated with a given type vehicle including daily, weekend and weekly rate, insurance cost, mileage charge, and number of free miles. It must also allow the user to determine the cost of a particular vehicle, for a given period of time, an estimated number of miles, and the cost of optional insurance.

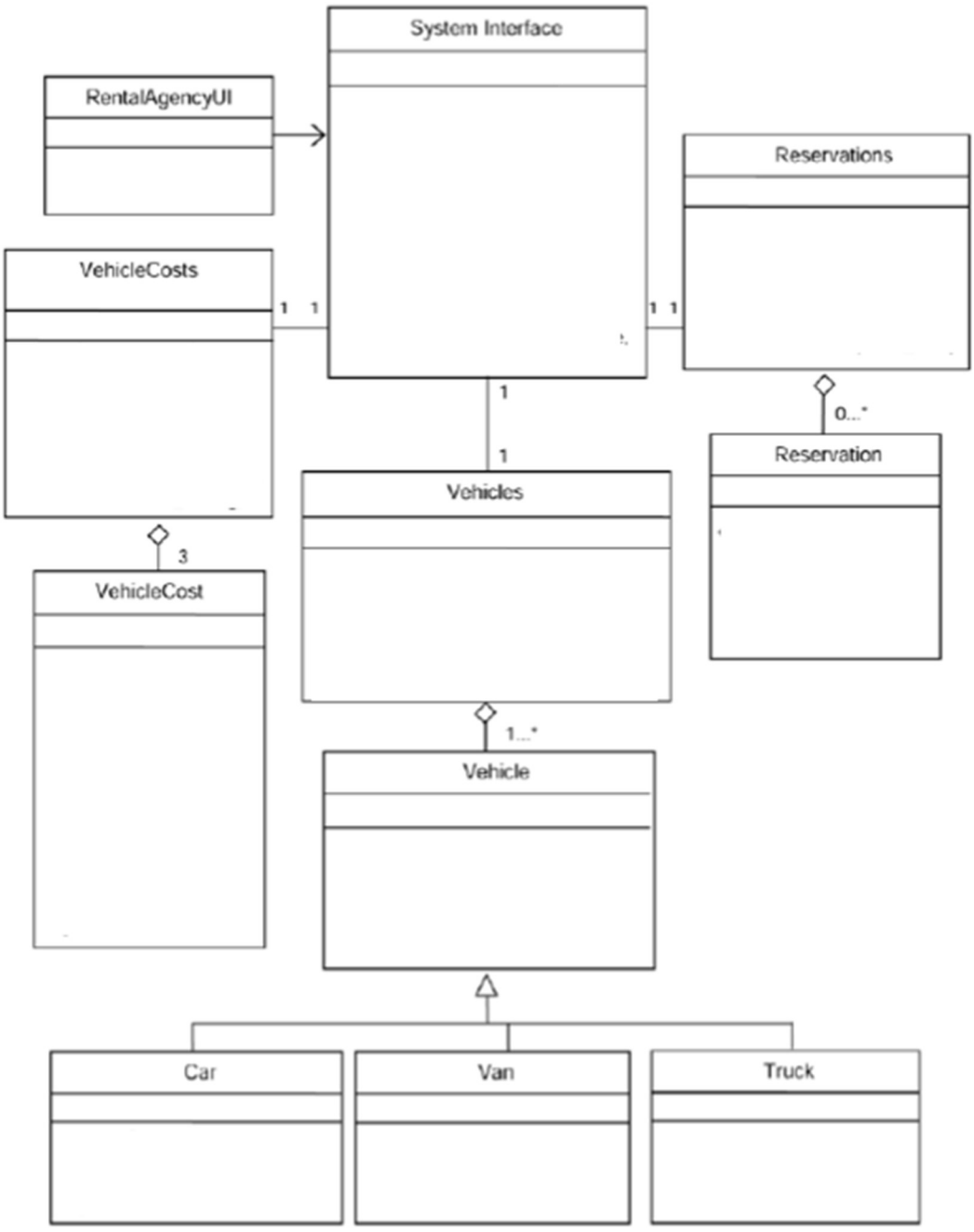
The program must be able to allow a particular vehicle to be reserved and cancelled.



Overall Design of the Vehicle Rental Agency Program

You are requested to deliver UML diagram for the program:

- Use the following as your starting point.
- Create a report for your submission with proper description of the problem
- Bonus: Use Modelio or any other software modelling environment to create your UML diagram.
  https://www.modelio.org/downloads/download-modelio.html


System Interface
RentalAgencyUI
Reservations
VehicleCosts
1
1
1
1
1
0...*
Reservation
Vehicles
3
VehicleCost
1...*
Vehicle
Car
Van
Truck