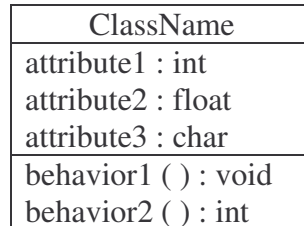


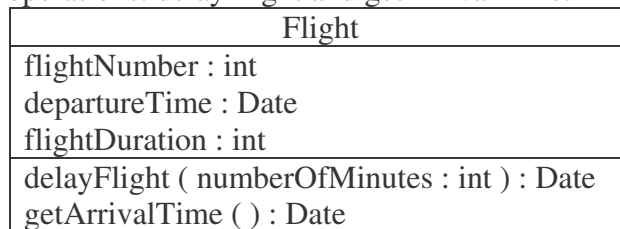
## UML Class Diagram basics

The class diagram shows each class's internal structure along with the relationship that the class has to other classes. The UML representation of a class – a class diagram – is a rectangle containing three compartments stacked vertically. The top compartment shows the class's name. The middle compartment lists the class's attributes. The bottom compartment lists the class's behaviors.



A generic class diagram showing a single class

Let's look at a real-world example from the commercial airline industry. Below, we have an airline flight modeled as a UML class on a class diagram. As we can see, the name is Flight, and in the middle compartment we see that the Flight class has three attributes: flightNumber, departureTime, and flightDuration. In the bottom section we see that the Flight class has two operations: delayFlight and getArrivalTime.



Class diagram of the airline class Flight

### The class's attribute list

The attribute section of a class (the middle compartment) lists each of the class's attributes on a separate line. Each attribute line uses the following format:

name : attribute type

For example,

flightNumber : int

We can also set a default value for an attribute:

name : attribute type = default value

For example,

flightDuration : Minutes = 60

Flight
flightNumber : int departureTime : Date flightDuration : int = 60
delayFlight ( numberOfMinutes : int ) : Date getArrivalTime ( ) : Date

**A Flight class diagram showing the flightDuration attribute's value defaulted to 60 minutes**

### The class's behavior list

The class's behaviors are documented in the third (lowest) compartment of the class diagram's rectangle. Like attributes, the operations of a class are displayed in a list format, with each behavior on its own line. Behaviors are documented using the following notation:

name(parameter list) : type of value returned

For example:

store\_sides (s1 : float, s2 : float) : void

### Public/Private

Classes allow us to protect data. For our purposes, all attributes will be private (unable to be directly accessed by a calling function) and all behaviors (methods) will be public. In a UML class diagram, we indicate private with a "-" and public with a "+" (see below):

Flight
-flightNumber : int -departureTime : Date -flightDuration : int = 60
+delayFlight ( numberOfMinutes : int ) : Date +getArrivalTime ( ) : Date

\*\*\*\*\* In Lab Discussion \*\*\*\*\*

Functions (behaviors of an object) already "know" about the attributes in an object. For example, if you use a function "store\_sides" to set the sides attributes in an object, the other functions "know" about those values and can use or change the values. So, if a class had a function calc\_area, you would not have to pass that function any information. It uses the values already stored in object to calculate and return the area.