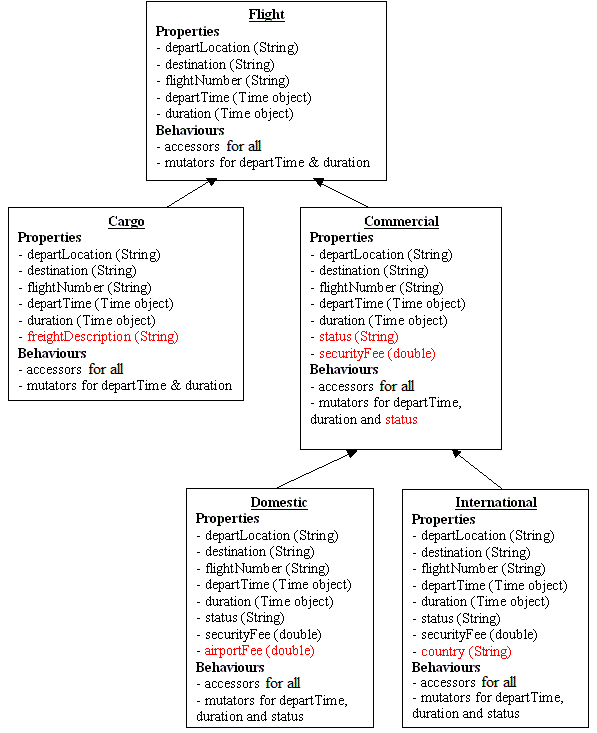
**COSC 121 - Assignment #1**

You will generate five classes related to flight information records for *CS Airlines*. You will define each of the classes in the hierarchy of classes shown below. Note that the highlighted members (indicated in **red**, bold and italic) are the members that make each class unique from its parent class.



A Flight represents a generic record of information related to one of CS Airlines' flights. CS Airlines has 3 types of flights - *cargo*, *domestic* commercial, and *international* commercial. In general, the information associated with any one of these types of flights is similar in that all flight records contain:

* A departure location (departLocation)
* A destination location (destination)
* A flight number (flightNumber)
* A departure time (departTime)
* A flight duration (duration)

A *Cargo* flight record is a unique type of *Flight* record in that a *Cargo* flight record contains a description of the freight (freightDescription) as well.

A *Commercial* flight record is a unique type of *Flight* record in that a *Commercial* flight record contains status information (status).The status of a flight is generally one of "*On Time*", "*Delayed*", or "*Cancelled*".

A *Domestic* flight record is a unique kind of *Commercial* flight record in that a *Domestic* flight record also contains an airport fee amount (airportFee).

An *International* flight record is a unique kind of *Commercial* flight record in that an *International* flight record also contains the name of the destination country (country).

I have provided you with the Flight class. You can download it from here: [Flight.java](file:///D:\cosc%20121%20-%20prep\Alan\aW2014\W2013-WebsiteFinal\labs\Flight.java.txt). Your are not permitted to modify the Flight.java file in any way. Even though you must use my Flight class, I strongly encourage you to first sketch out on paper how you would define the Flight class yourself, and then compare it to mine. You will notice that this class uses the services of another class called Time. You can download this class from here: [Time.java](file:///D:\cosc%20121%20-%20prep\Alan\aW2014\W2013-WebsiteFinal\labs\Time.java.txt).

Study the Time class to ensure that you understand what it does. It does not do anything too complicated. It is just a class to represent a time in the form of *HH*:*MM* where *HH* is the hour and *MM* is the minute.

Your job is to define four classes - Cargo, Commercial, Domestic, and International. You must define these classes in four separate files. You may **not** modify Flight class. Flight.java and Time.java should also belong to your Eclipse project called, *Flight*.

Here are some additional requirements for this set of classes.

* Name your data members using the names (exactly) indicated in the hierarchy diagram. Pay attention to upper and lower case letters.
* Name your accessor methods by following exactly the convention we have been using in this course and CoSc 111. For example, the accessor method for freightDescription in the Cargo class should be named getFreightDescription. Put the word get in front of the name of the variable with the first letter of the variable's name changed to upper case. **If you don't follow this convention, my driver program won't compile, and as a result you will lose marks**.
* You must define (override) toString() methods for your classes that textually represent the objects exactly as shown in the sample output presented below.
* Note that the Flight class creates only the first part of the flight number. The classes that you create should modify the flight number appropriately. I.e. add "-C", "-D" or "-I" to the end of the flight number depending on whether the flight is Cargo, Domestic or International, respectively.
* Be sure to define properties and methods in the appropriate class and be sure to take advantage of code that is already defined.
* You may download this initial driver program to test your classes - [CSAirlines.java](file:///D:\cosc%20121%20-%20prep\Alan\aW2014\W2013-WebsiteFinal\labs\CSAirlines.java.txt). This program does not exercise any of your accessor methods, so you will probably want to add code to make sure they work.
* Here is the way this program should run when accessing your flight classes:

Flight CS001-C departs Vancouver for Pearson at 08:30   
Freight: Hockey Sticks

Flight CS002-C departs Calgary for Hamilton at 09:22   
Freight: Diamonds

Flight CS003-D departs Vancouver for Victoria at 10:11   
Status: Cancelled

Flight CS004-D departs Kelowna for Victoria at 12:55   
Status: On Time

Flight CS005-I departs Vancouver for Las Vegas at 17:00   
Status: On Time   
Country: United States

Flight CS006-I departs Vancouver for Paris at 20:50   
Status: Delayed   
Country: France

* Note that CSAirlines.java creates Domestic and International objects without specifying a value for status, as it is reasonable to assume that any Flight object would be *On Time* when it is first created. Be sure to create your constructors as the CSAirlines class requires.

**Submitting your work**

You can now submit your **Cargo.java**, **Commercial.java**, **Domestic.java**, and **International.java** files on Moodle. This assignment is curtesy of Alan Kennedy.