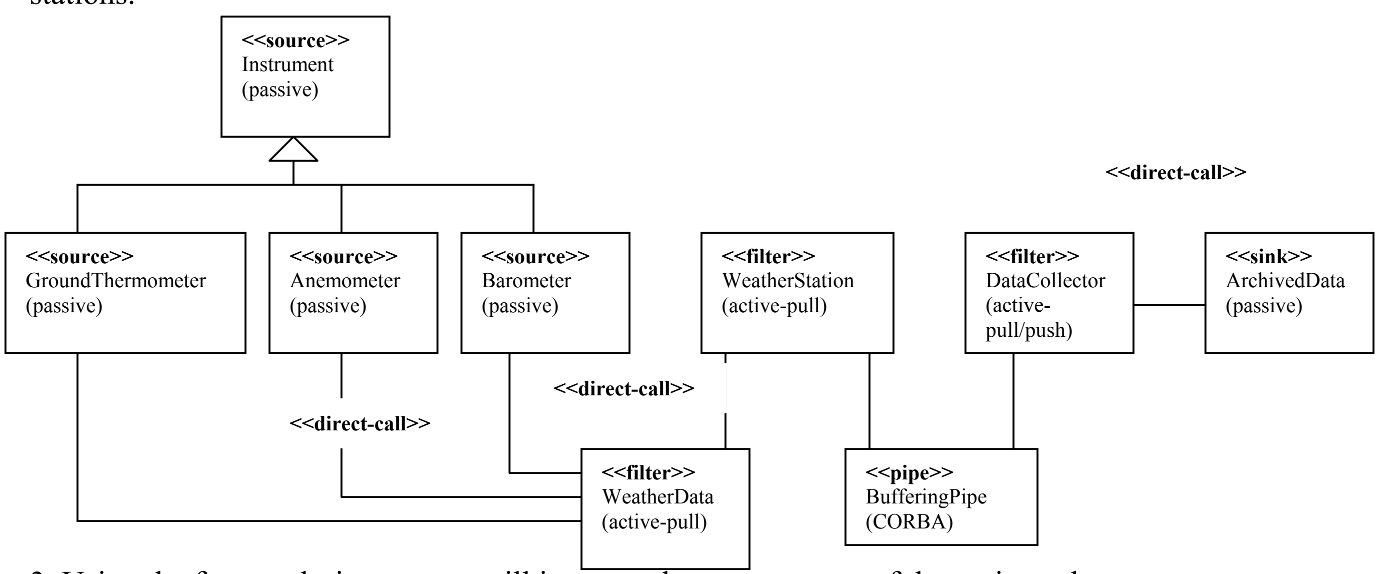
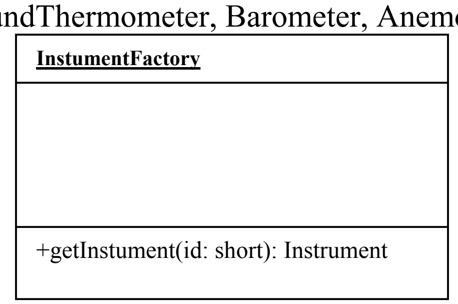
Lab4: Solutions

1. *Extensibility, maintainability,* and *portability* are among key requirements of the system. It is expected that the product requirements may change during development or operation. The physical characteristics, the kind and the number of the instruments involved may change.
An important design issue in that case consists of making easy removal and addition of components. That can be achieved using the pipes-and-filter pattern.
In order to handle the changes in the instruments, the factory design pattern is a mechanism that can introduce the desired level of decoupling.

2.The pipes-and-filter pattern can be applied to the given class structure in various ways. The instruments must be designed as data source, active or passive. WeatherData class may be designed either as a filter or as a pipe. WeatherStation and DataCollector must represent active filters. ArchivedData must be designed as a passive data sink. According to the requirements, the area computer, through DataCollector class, polls weather stations. So DataCollector is an active filter based on push/pull mechanism. Since WeatherStation is also an active filter, a synchronization buffer may be included here; the buffer may be implemented later using an Interprocess Communication Mechanism such as CORBA. The DataCollector may also pull data directly from weather stations.



3. Using the factory design pattern will improve the management of the various classes corresponding to weather instruments and to anticipate changes in the class structure. The factory design pattern is applied by integrating a factory class named *InstrumentFactory* in the class diagram that will be used to create instances of instruments. It provides the *getInstrument()* method, which is in charge of calling the appropriate constructors (GroundThermometer, Barometer, Anemometer).



4. The difference between the two grouping proposed is the allocation of class WeatherData.

So we only need to compute coupling metrics related to this class:

Grouping 2:

Coupling (station, sensor) = CBO(WeatherData, WeatherStation)= 1

Grouping 1:

Coupling(station,sensor) = CBO(WeatherData,Anemometer) + CBO(WeatherData, GroundThermometer) + CBO(Barometer) = 1+1+1=3.

The best grouping is the one that provides the lowest coupling between modules. We need of course to evaluate also cohesion and if necessary make a trade-off.