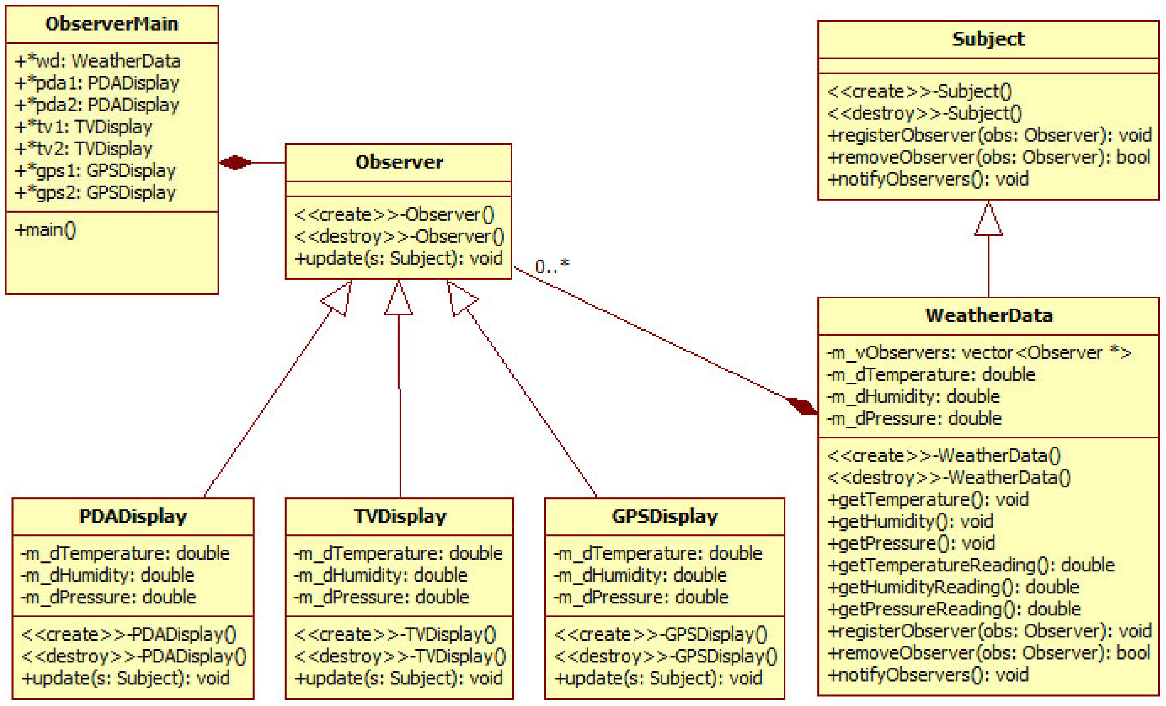
Observer Design Pattern

GoF Statement:	**Defines a one-to-many dependency between objects so that when one object changes state, all of its dependents are notified and updated automatically.**
Category:	**Behavioral**
UML Diagram:	



Description of the Demonstration:

The WeatherData class, a sub-class of Subject, maintains a vector of pointers to instances of sub-classes of Observer. The subclasses are PDADisplay, TVDisplay, and GPSDisplay. At startup in main() instances of each subclass of Observer are created and passed to the instance of WeatherData::registerObserver. The demonstration loop then runs for 15 seconds. Each second the instance of WeatherData is told to collect all of its weather information then a call is made to WeatherData::notifyObservers. The WeatherData object then calls Observer::update on each of the Observers it has registered. Each Observer can then make calls back to WeatherData to get the data they need to display. At any time an Observer can call WeatherData::removeObserver to unsubscribe from the data feed or call WeatherData::registerObserver to resubscribe to the data feed.