CPSC 3200 Object-Oriented Development

Programming Assignment #5: Due Monday, November 9, 2020 before MIDNIGHT

P5 exercises your understanding of multiple inheritance and interfaces

For an acceptable P5 submission:

1. Reuse two inheritance hierarchies to define 'cross-product' functionality
2. Use the C# interface construct
3. Fulfill requirements as specified in steps 1-9 from P1

Part I: Class Design

Reuse the *dataFilters* from P3 -- *dataFilter, dataMod* and *dataCut*

Define a second class hierarchy of *beacon*s, where each *beacon* object:

1) May be on or off
2) May be charged or not
3) Emits a signal, if on and charged, which reduces its charge
4) Accepts an integer sequence to vary its signal

strobeBeacon is-a *beacon* that alternates its signal response, oscillating between negative and positive (or high and low). *strobeBeacon*s cannot be recharged.

quirkyBeacon is-a *beacon* that emits signals from which a discernible pattern is not evident.
*quirkyBeacon*s can be turned off and on only a limited number of times (variable across type but stable for an individual object).

Define and implement the classes need to mimic 'multiply inherited' *dataFilterBeacon* 'types' and thus reflect the 'cross-product' of both inheritance hierarchies. Composite type functionality must be represented, e.g. *dataFilterBeacon*, *dataModBeacon*, *dataCutBeacon*, *dataFilterStrobeBeacon*, *dataFilterQuirkyBeacon*, *...,* etc.

Many details are missing. You MUST make and DOCUMENT your design decisions!!
Do NOT tie your type definition to the Console.

Use Unit Testing to verify functionality of each new class (no need to retest the classes from P3).

Part II: Driver (P5.cs) -- External Perspective of Client – tests inheritance hierarchy design

The P5 driver must test the use of the 'multiply-inherited' types together.
Thus, it will differ from the unit tests which test each type separately.
Additionally:

1) Use at least one heterogeneous collection for testing functionality
2) Instantiate a variety of objects
3) Trigger a variety of mode changes