**CIS255 Programming Assignment #1**

**OOP Encapsulation: Creating Your Own Class with C++**

Assigned: Week 3 Due: Week 4 Points: 40

Write a C++ application, and an additional class (with .cpp and .h files) to represent some real-world entity such as a technology item, an animal, a person, a vehicle, etc. Keep in mind that a class is a model in code of something real or imagined, which has attributes (member variables) and behaviors (member methods).

Note: This class will be used in future programs in which you will store objects in arrays, using arrays of objects and pointers, as well as searching and sorting.

The class will:

1. Create a total of 5 member variables for the class, selecting the appropriate data types for each field. For example, a class to represent a lamp might include color, price, height, numBulbs, batteryOperated. Each of these 5 variables need a data type.
2. Include at least three different constructor methods, in addition to the default constructor (0 argument constructor). The constructor is a function (method) which allocates memory and initialized the member variables specified in (a.).
3. Include getters/setters for to serve as as mutators and accessors for each variable. Name these appropriately such as setColor & getColor, setPrice & getPrice, setHeight & getHeight, setNumBulbs and getNumBulbs, and setBatteryOperated & getBatteryOperated.
4. Create a member function showValues() to display the values of an object in a vertically formatted manner.
5. Create a member function detailLine() to display the values of an object in a horizontally formatted manner.
6. Create a member function getValues() to interactively accept the values for an object.
7. Create at least 2 other member functions (methods) for the class that will perform some operation on the data (i.e. calculations or additional report/display of output). For example, turnLampOn, changeBulb, etc.

**Required Output: Generate output samples demonstrating each of the requirements (1-7). Utilize all your member functions, adequately testing them and showing their functionality (i.e. inputting values, outputting values (displaying them), performing calculations, etc.).**