COEN 160 **OO Analysis, Design and Programming** **Winter 2017**

Assign 2 (125 + 20 EC) **Due: (16th Feb 10 PM)**

In this assignment, you will learn to

- Use Composition and Inheritance to achieve reuse

Question 1 (125 + 20 extra points)

This question tests your understanding of the concepts of inheritance and containment (as techniques of reuse) as they are applied to common artifacts that we are familiar with. In addition, the question allows you to be creative in applying the concepts to create additional artifacts of interest.

a) Firstly, you are required to define a few object types (classes) to represent a number of basic concepts: A **Rectangle**, a **Ball**, a **Picture**, a **Frame** and a **Block.**

Some of the important properties of the basic classes for the above concepts are given below. You are free to add any other properties that you may find applicable. Please note that not all properties may need to be stored as data members; some may be computed in methods.

Rectangle
Dimensions (Length, Width), Area
Ball
Dimensions (Radius), Volume
Picture
Dimensions (Length, Width), Title, Artist
Frame
Dimensions (Length, Width), Area, Material
Block
Dimensions **(***Height, Depth, Length)*

Note: You are free to define any other basic classes (similar to the ones above) of your choice.
(50 pts)

b) After you define the above basic classes, you must create some new classes via inheritance and composition of the basic classes above. For example, some of the new classes can be a **Square**, a **Box**, a **FramedPicture** and so on.

Points:

- You are required to define at least 5 new classes (using the basic classes via containment and inheritance) **(50 points)**
- You are free to create additional classes (up to 2) for an additional **20 pts**.

For each of the new classes you create (from the basic classes), define the following:

- Data Members
- Constructors
- Setters and getters (as applicable)
- A *toString()* method to show the contents of the object
- Other methods relevant to that particular class.

  Note: *Make sure that some of the basic constraints (for example, the dimensions of a Frame object should be able to accommodate a Picture object, in a FramedPicture class)*

c) Define a **Question1Tester** class with main(), where you create instances of the classes you have defined and call the methods to show they work correctly. **(15 pts)**

d) Draw an UML class diagram (using an UML tool) showing the classes and relationships (containment and inheritance) among them. **(10 pts)**

Point distribution:

a) 50 pts
b) 50 pts + a possible 20 extra points
c) 15 pts
d) 10 pts