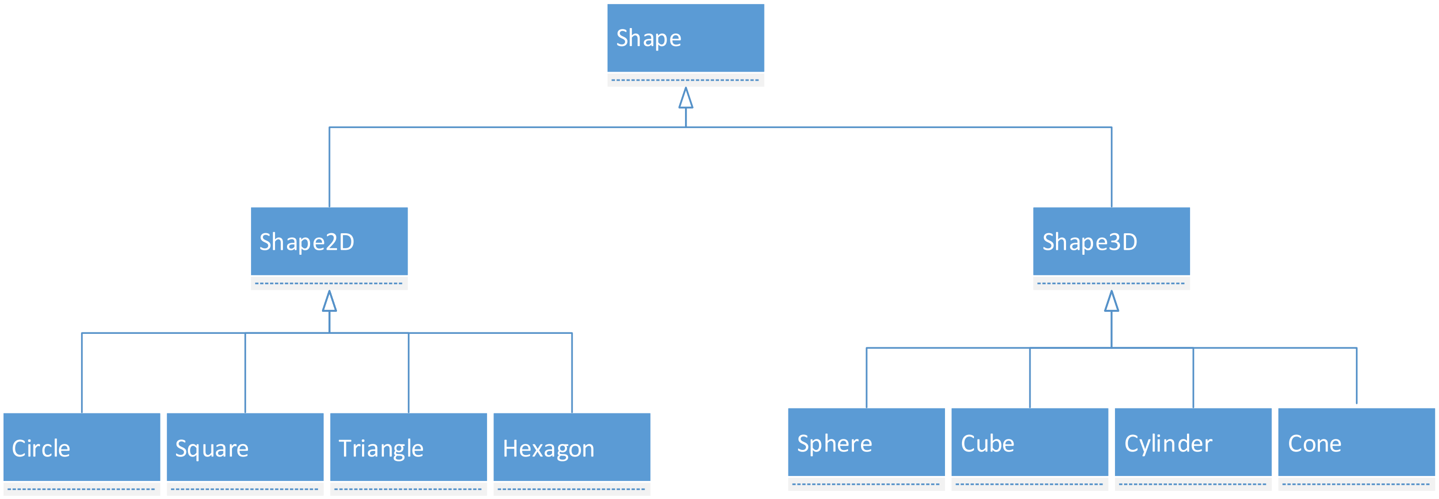
CS120: Programming 2

"Shape Hierarchy" Term Project

Due Date: Thursday November 10th, @ 2 pm

Description: Apply what you have learned in OOP, Inheritance, Polymorphism and Files I/O techniques to build the above shown **Shape** hierarchy. For all classes we need to store the following information:



1. All classes must have a ***color*** and ***name*** attributes, both of a string type and stored in the base class (**Shape**).
2. All **<u>2D shapes</u>** must have ***area*** and ***perimeter*** attributes, both of double type and stored in their base class (**Sahpe2D**).
3. All **<u>3D shapes</u>** must have ***surfaceArea*** and ***volume*** attributes, both of double type and stored in their base class (**Shape3D**),
4. **readData()**, **print()**, **printToFile(ofstream &output)**, **readFromFile(ifstream &input)** functions must be implemented in all classes (base and derived), taking into consideration they will be used for polymorphism,
5. **computeArea()** and **computePerimeter()** functions must be implementer in all **<u>2D shape</u>** classes (including their base class), taking into consideration their use in polymorphism,
6. **computeSurfaceArea()** and **computeVolume()** functions must be implemented in all **<u>3D shape</u>** classes (including their base class), taking into consideration their use in polymorphism,

7. For each of the derived classes, we need to store the information shown in the below table in double format:

Class	Attributes
Circle	**radius**
Square	**side**
Triangle	**base, height**
Hexagon	**side**
Sphere	**radius**
Cube	**side**
Cylinder	**radius, height**
Cone	**radius, height**

8. You need to find out the functions to compute ***area***, ***perimeter***, ***surfaceArea*** and ***volume*** attributes on your own.

Requirements

1. For every class, you need to build a **default/parametrized constructor**,
2. The default constructors set the objects' data members to 1.0,
3. Parametrized constructors are used to set objects' data members using the values passed to the objects during instantiation,
4. **readData()** function is used to read all the object's data members from the user,
5. **print()** function is used to print all object's data to the screen,
6. Your program must be able to write all objects' data to the binary file **"Shapes.dat"**, using the **printToFile** functions,
7. Your program should be able to read and append to the **"Shapes.dat"** file as needed, using the **readFromFile** functions,
8. Your main file must be able to store/retrieve data to/from output/input files depending on the user's choice,
9. It is not known how many objects will be added or their order of addition.

Teamwork

1. You need to divide yourselves into groups of 4 students each maximum,
2. Every group must choose a leader amongst themselves,
3. The leader is responsible (in addition to programming) of assigning tasks to every student in the group,
4. At the end of the project's due date, the leader is responsible of uploading all project's files to the Group's specified folder on the Blackboard system,
5. The leader must specify the tasks assigned to each member of the group in the final report.