CS 1410 Introduction to Computer Science – CS2
Section 1: MWF 10:30 a.m. – 11:20 a.m.
Section 2: MWF 1:00 p.m. – 1:50 p.m.
Instructor: Xiaojun Qi
Assignment #5

Given: Thursday, Feb. 20, 2014
Due: 11:59 p.m. Saturday, March 1, 2014
Total Points: 35 points

This assignment is a bit more abstract than your previous assignments. In other words, there isn't some specific example problem you'll be solving, instead, you will be writing a set of derived classes based upon a description. You are responsible for **making smart choices** about how to design the classes. Your assignment will be graded in a large part according to the design decisions you make.

In this assignment, you will create a set of classes that describe various forms of geometry and generate a simple report for each geometric object.

You are to create seven classes:

1. A base class named `Geometry`
2. A derived class named `Rectangle`
3. A derived class named `Circle`
4. A derived class named `Box` (i.e., a box is a 3D rectangle)
5. A derived class named `Sphere` (i.e., a sphere is a 3D circle)
6. Two derived classes of your choice that represent a 2D geometry object and its corresponding 3D geometry object, respectively.

The `Geometry` class **must be an abstract base class**. It will have the following features:

- Constructor and destructor code as you decide is appropriate.
- A '`char *`' data member called '`Name`' that stores a named description of the object.
- A '`char *`' data member called '`Type`' that stores a description of the type for the object. That is, if the object is a `Box`, the `Type` member would be "Box". If the object is a `Sphere`, the `Type` member would be "Sphere".
  Note: You cannot use the `string` data type for `Name` and `Type`. In other words, **it is required that you must use a '`char *`' for both of these two data members. This requirement means that you need to explicitly write a copy constructor, an overloading assignment operator, and a destructor.**
- The class will have two methods, `GetName`() and `GetType`(), that return the '`char *`' for their respective data members.
- You **may not** write `SetName` and `SetType` member functions since all the information related to an object is set by an appropriate constructor.
- The class will have two methods, `ComputeVolume()` and `ComputeSurface()` that will, when called using the appropriate object, return the volume or surface area for that object, respectively.

- Volume for a sphere is: (4/3)PI*radius^3
- Surface area for a sphere is: 4*PI*radius^2
- Volume for a circle is: 0
- Surface area for a circle is: PI*radius^2

Each of the derived classes will have the following features:

- Define the `Type` of the object. That is, the object itself will set the value in the `Type` variable. **Remember, there is no `SetType` function!**
- Provide the appropriate volume and surface area computations.
- Provide storage for unique data values.
- **No SetX or GetX functions may be added. Here, X indicates the data member that the derived class is able to access.**

At last, you are to write a driver program to demonstrate the use of these objects. Your driver program will have the following features:

- A function called report that accepts a Geometry pointer and generates a simple report about the object passed into the function. The prototype will look like:
  - `void report(Geometry *obj)`
- The following is a section from the sample output (**Figure 1**):

```
----- Geometry Report -----
Type: Box
Name: Box 1
Volume: 6
Surface Area: 22
```

- You will create an array of Geometry object pointers which dynamically allocate six pointers associated with each of the six derived classes, respectively. **You will then call the `report` function for each object to summarize its geometric information as shown in Figure 1. Don't forget to clean up!** Here is a short code segment of what the main function in your driver code will look like:

```
void report(Geometry *obj) ;
int main()
{
    Geometry *array[ITEM_COUNT] ;

    array[0] = new Box("Box 1", 1, 2, 3) ;  //Its output is shown in Figure 1.
```

You are responsible for deciding what constructors to make, what parameters (if any) they should accept, and what the appropriate visibility for each function and data member should be. You should be responsible for writing up the report function and call the report function to report the geometric information of each object stored in the array. Your program should also report the following information:

1) The geometry report of the six objects in the ascending order based on the volumes of the objects.
2) The geometry report of the six objects in the descending order based on the surfaces of the objects.
3) Statistics (e.g., average volume, average surface, median volume, median surface) of the six objects.

You should be responsible for writing up the driver to test the correctness of the copy constructor, overloaded assignment operator, and destructor for each derived class. Don't try to write this whole program at once, start small, and then keep building.