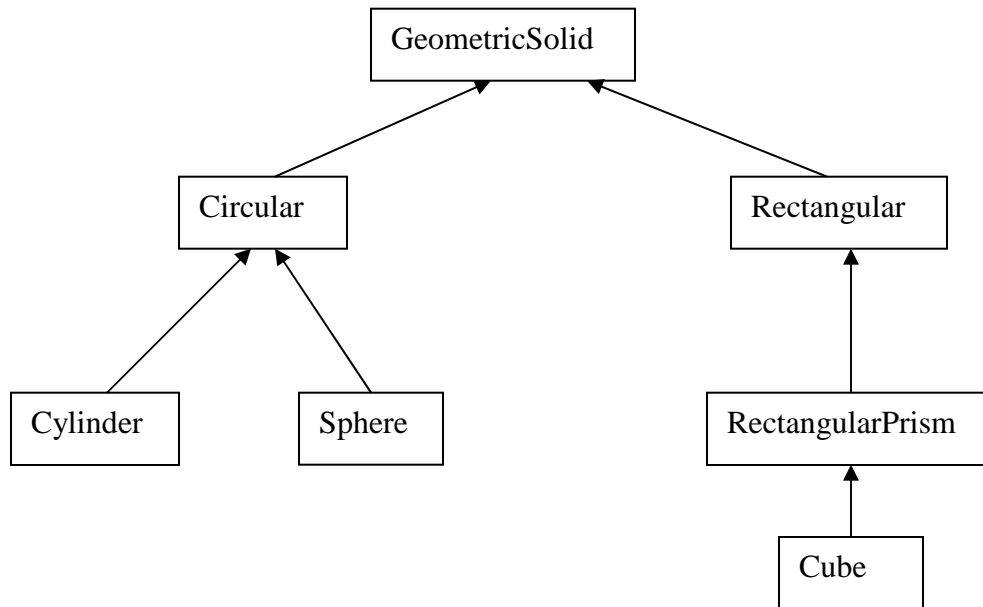


## Assignment 2



Create the inheritance hierarchy illustrated above. The GeometricSolid, Circular, and Rectangular classes should be abstract. Create three pure virtual functions in the GeometricSolid class and define/declare them in the hierarchy as necessary:

1. A name() function that returns the class type as a const char\*, for all classes.
2. A volume() function that returns the volume of the solid.
3. A surface\_area() function that returns the surface\_area of the solid.

The Circular class should have a data member that represents a radius.

The Rectangular class should have three data members, representing length, width, and height.

Input data for this assignment will be provided in a 50 record randomly ordered text file, **ass2data**. The format for this file will be a two-byte character code (“cy”, “sp”, “re”, or “cu”) indicating the type of solid, followed by 1, 2, or 3 doubles representing requisite data members of the class. Specifically the data format will appear as:

<u>GeometricSolid</u>	<u>Code</u>	<u># of doubles</u>	<u>Representing</u>
Cylinder	cy	2	radius and height
Sphere	sp	1	radius
RetangularPrism	re	3	length, width, and height
Cube	cu	1	side

Input file: **ass2data**

```
cu    2.19966
sp    8.15059
cy    2.62835  0.782609
cu    39.6149
re    8.46099  8.44898  0.47349
sp    1.2963
...
```

Your program needs to perform the following tasks:

1. Read in the data file, correctly processing each particular type of GeometricSolid.
2. Using a pointer to a GeometricSolid, and making use of polymorphism, allocate memory dynamically for a specific type of GeometricSolid. Using polymorphism print the type, surface area and volume for each record.
3. The output should be formatted such that any surface area or volume that is greater than 10000, should be displayed in scientific notation with 3 significant digits of precision in the mantissa. Surface area or volumes less than 10000 should be displayed in fixed notation with 4 decimal place accuracy. The output should match that which is partially displayed below.

Program Output

```
cube           29.0310           10.6431
sphere         834.8106           2268.0663
cylinder       56.3300             16.9848
cube          9416.0418           6.22e+004
rectangular prism 158.9869           33.8483
sphere        21.1164             9.1244
...
```

Your main() might contain a few lines similar to:

```
Solid* ptrSolid;           // declare a pointer to a Geometric Solid

for (unsigned i = 0; i < 50; ++i) {
    fin >> code;
    if (!strcmp(code, "cy")) {
        fin >> d1 >> d2;
        ptrSolid = new Cylinder(d1, d2);
    }
    else ...
    ...
}
cout << left << setw(18) << ptrSolid->name() << ...
cout << ... << ptrSolid->surfaceArea();
...
```

You may need the following formulas for this assignment:

<b>Solid</b>	<b>Volume</b>	<b>Surface Area</b>
Rectangular Solid	$V = l w h$	$A = 2(lw + lh + wh)$
Sphere	$V = 4/3\pi r^3$	$A = 4\pi r^2$
Cylinder	$V = \pi r^2 h$	$A = 2\pi r(h + r)$
Cube	$V = s^3$	$A = 6s^2$