ECE 1410 Polymorphic Points, Circles, and Cylinders Requirements

Modify the previous assignment, in which you developed Point, Circle, and Cylinder classes, in the following ways:

- Add an abstract base class named Shape (define it in shape.h).

- Point inherits from Shape. Circle continues to inherit from Point and Cylinder continues to inherit from Circle.

- Point, Cylinder, and Circle must continue to support the insertion operator (<<), but it must also support polymorphism (i.e. work with pointers to Shapes). See the example below. **HINT: define the insertion operator ONLY in Shape, but have it call a virtual function that is redefined in each inherited class.**

The following main.cpp file:

```
#include <iostream>
#include "shape.h"
#include "point.h"
#include "circle.h"
#include "cylinder.h"

using namespace std;

int main()
{
      Shape *s[3];

      Point p(4,4);
      Circle c(5,5,5);
      Cylinder y (6, 6, 6, 6);

      s[0] = &p;
      s[1] = &c;
      s[2] = &y;

      cout << *(s[0]) << endl << endl;
      cout << *(s[1]) << endl << endl;
      cout << *(s[2]) << endl << endl;

      return 0;
}
```

produces this output:

```
Point at (4, 4)

Circle with center = (5, 5); Radius = 5; Area = 78.5397

Cylinder with center = (6, 6); Radius = 6; Height = 6; Volume = 678.583
```

Submit 7 files (shape.h, point.h, point.cpp, circle.h, circle.cpp, cylinder.h, and cylinder.cpp) combined as a single zip file. The graders will compile your source with their main.cpp to test program functionality.