```cpp
/*  Program: A5P3 - Shapes
    Author: Tom Stutler
    Last Date Modified: 5/7/2015

    The intent of this program is to demonstrate knowledge
    of virtual functions in c++ and practice using inherited
    classes again.
*/

#include <iostream>
#include <cmath>

using namespace std;

const float PI = 3.14159265359;
const int MAX_SHAPES = 20;

class Shape
{
protected:
    float volume, perim, area;

public:
    Shape () : volume(0), perim(0), area(0) {}

    virtual void Display() = 0;
    virtual void GetDimensions() = 0;
    virtual float Perimeter() = 0;
    virtual float Area() = 0;
    virtual float Volume() = 0;
};

class Rectangle : public Shape
{
private:
    float length, width;

public:
    Rectangle () : Shape(), length(0), width(0) {GetDimensions();}

    void Display ()
    {
        cout << "\nShape = Rectangle "
             << " Length = " << length
             << " Width  = " << width
             << endl;
    }
    void GetDimensions ()
    {
        cout << "Input for Rectangle - enter length and width: ";
        cin >> length >> width;
    }
    float Perimeter ()
    {
        perim = 2*length + 2*width;
        cout << "Perimeter = " << perim << endl;
    }
    float Area ()
    {
        area = length*width;
        cout << "Area = " << area << endl;
    }
    float Volume () {}
};

class Circle : public Shape
```

```
67  {
68  private:
69      float r;
70
71  public:
72      Circle () : Shape(), r(0) {GetDimensions();}
73
74      void Display ()
75      {
76          cout << "\nShape = Circle  Radius = " << r << endl;
77      }
78      void GetDimensions ()
79      {
80          cout << "Input for Circle - enter radius: ";
81          cin >> r;
82      }
83      float Perimeter ()
84      {
85          perim = 2*PI*r;
86          cout << "Perimeter = " << perim << endl;
87      }
88      float Area ()
89      {
90          area = PI*pow(r,2);
91          cout << "Area = " << area << endl;
92      }
93      float Volume () {}
94  };
95
96  class Triangle : public Shape
97  {
98  private:
99      float a, b, c;
100
101 public:
102     Triangle () : Shape(), a(0), b(0), c(0) {GetDimensions();}
103
104     void Display ()
105     {
106         cout << "\nShape = Triangle  Side lengths = "
107              << a << " "  << b << " " << c << endl;
108     }
109     void GetDimensions ()
110     {
111         cout << "Input for Triangle - enter side 1, side 2, side 3: ";
112         cin >> a >> b >> c;
113     }
114     float Perimeter ()
115     {
116         perim = a+b+c;
117         cout << "Perimeter = " << perim << endl;
118     }
119     float Area ()
120     {
121         float p = perim/2.0;
122         area = sqrt(p*(p-a)*(p-b)*(p-c));
123         cout << "Area = " << area << endl;
124     }
125     float Volume () {}
126 };
127
128 class Box : public Shape
129 {
130 private:
131     float l, w, h;
132
```

```
public:
    Box () : Shape(), l(0), w(0), h(0) {GetDimensions();}

    void Display ()
    {
        cout << "\nShape = Box  Length = " << l
             << " Width = " << w
             << " Height = " << h << endl;
    }
    void GetDimensions ()
    {
        cout << "Input for Box - enter length, width, and height: ";
        cin >> l >> w >> h;
    }
    float Volume ()
    {
        volume = l*w*h;
        cout << "Volume = " << volume << endl;
    }
    float Area ()
    {
        area = 2*l*w + 2*w*h + 2*l*h;
        cout << "Surface area = " << area << endl;
    }
    float Perimeter () {}
};

class Can : public Shape
{
private:
    float r, h;

public:
    Can () : Shape(), r(0), h(0) {GetDimensions();}

    void Display ()
    {
        cout << "\nShape = Can  Radius = " << r
             << " Height = " << h << endl;
    }
    void GetDimensions ()
    {
        cout << "Input for Can - enter radius and height: ";
        cin >> r >> h;
    }
    float Volume ()
    {
        volume = PI*pow(r,2)*h;
        cout << "Voume = " << volume;
    }
    float Area ()
    {
        area = 2*PI*pow(r,2) + 2*PI*r*h;
        cout << "Surface area = " << area << endl;
    }
    float Perimeter () {}
};

class Cone : public Shape
{
private:
    float r, h;

public:
    Cone () : Shape(), r(0), h(0) {GetDimensions();}

```

```
199      void Display ()
200      {
201          cout << "\nShape = Cone  Radius = " << r
202               << " Height = " << h << endl;
203      }
204      void GetDimensions ()
205      {
206          cout << "Input for cone - enter radius and height: ";
207          cin >> r >> h;
208      }
209      float Volume ()
210      {
211          volume = (PI*pow(r,2)*h)/3;
212          cout << "Voume = " << volume << endl;
213      }
214      float Area ()
215      {
216          area = PI*pow(r,2) + PI*r*sqrt(pow(h,2)+pow(r,2));
217          cout << "Surface area = " << area << endl;
218      }
219      float Perimeter () {}
220  };
221
222  class Ball : public Shape
223  {
224  private:
225      float r;
226
227  public:
228      Ball () : Shape(), r(0) {GetDimensions();}
229
230      void Display ()
231      {
232          cout << "\nShape = Ball  Radius = " << r << endl;
233      }
234      void GetDimensions ()
235      {
236          cout << "Input for ball - enter radius: ";
237          cin >> r;
238      }
239      float Volume ()
240      {
241          volume = (4*PI*pow(r,3))/3;
242          cout << "Voume = " << volume << endl;
243      }
244      float Area ()
245      {
246          area = 4*PI*pow(r,2);
247          cout << "Surface area = " << area << endl;
248      }
249      float Perimeter () {}
250  };
251
252  int main()
253  {
254      Shape* pArray[MAX_SHAPES];
255      int selShape, numOfShapes=0;
256      char repeat;
257      Rectangle *rect;
258      Circle *circ;
259      Triangle *tri;
260      Box *bo;
261      Can *ca;
262      Cone *co;
263      Ball *ba;
264
```

```
265     do
266     {
267         cout << "It is time to enter your shape selection and dimensions.\n"
268              << "Enter the number of the shape type:\n"
269              << "   1 - Rectangle\n"
270              << "   2 - Circle\n"
271              << "   3 - Triangle\n"
272              << "   4 - Box\n"
273              << "   5 - Can\n"
274              << "   6 - Cone\n"
275              << "   7 - Ball\n"
276              << "=> ";
277         cin >> selShape;
278
279         switch (selShape)
280         {
281         case 1:
282             rect = new Rectangle;
283             pArray[numOfShapes] = rect;
284             numOfShapes++;
285             break;
286
287         case 2:
288             circ = new Circle;
289             pArray[numOfShapes] = circ;
290             numOfShapes++;
291             break;
292
293         case 3:
294             tri = new Triangle;
295             pArray[numOfShapes] = tri;
296             numOfShapes++;
297             break;
298
299         case 4:
300             bo = new Box;
301             pArray[numOfShapes] = bo;
302             numOfShapes++;
303             break;
304
305         case 5:
306             ca = new Can;
307             pArray[numOfShapes] = ca;
308             numOfShapes++;
309             break;
310
311         case 6:
312             co = new Cone;
313             pArray[numOfShapes] = co;
314             numOfShapes++;
315             break;
316
317         case 7:
318             ba = new Ball;
319             pArray[numOfShapes] = ba;
320             numOfShapes++;
321             break;
322
323         default:
324             cout << "Invalid - reenter\n";
325             break;
326         }
327
328         do
329         {
330             cout << "\nSelect another shape? (y or n): ";
```

```
            cin >> repeat;
            repeat = toupper(repeat);

        } while (repeat!='Y' && repeat!='N');

    } while(repeat=='Y' && numOfShapes<MAX_SHAPES);

    if (numOfShapes==MAX_SHAPES) {
        cout << "Max number of shapes reached.\n";
    }

    cout << "Next, a display of your selected shapes, dimensions, and other data\n";
    for (int i=0; i<=numOfShapes; i++) {
        pArray[i]->Display();
        pArray[i]->Area();
        pArray[i]->Perimeter();
        pArray[i]->Volume();
    }

    delete [] pArray;
    delete [] rect;
    delete [] circ;
    delete [] tri;
    delete [] bo;
    delete [] ca;
    delete [] co;
    delete [] ba;

    cout << "\n Thank you, come again!\n";
}
```