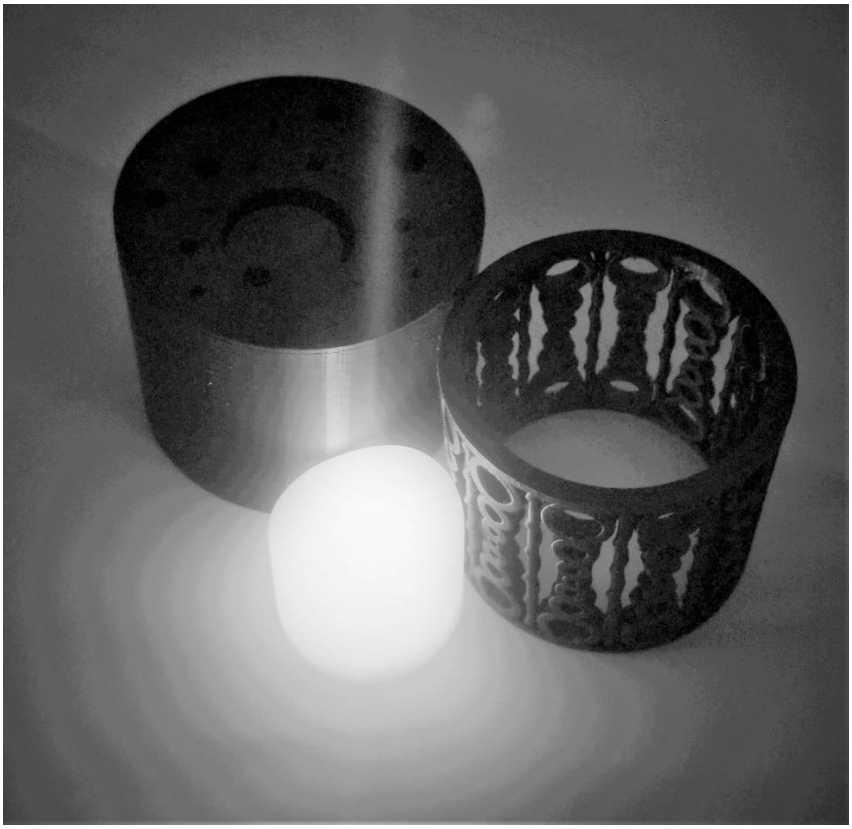
Digital Manufacturing

Uni: dy2421 | David Yi

Late Hours: 0 Used, 96 Hours Remaining

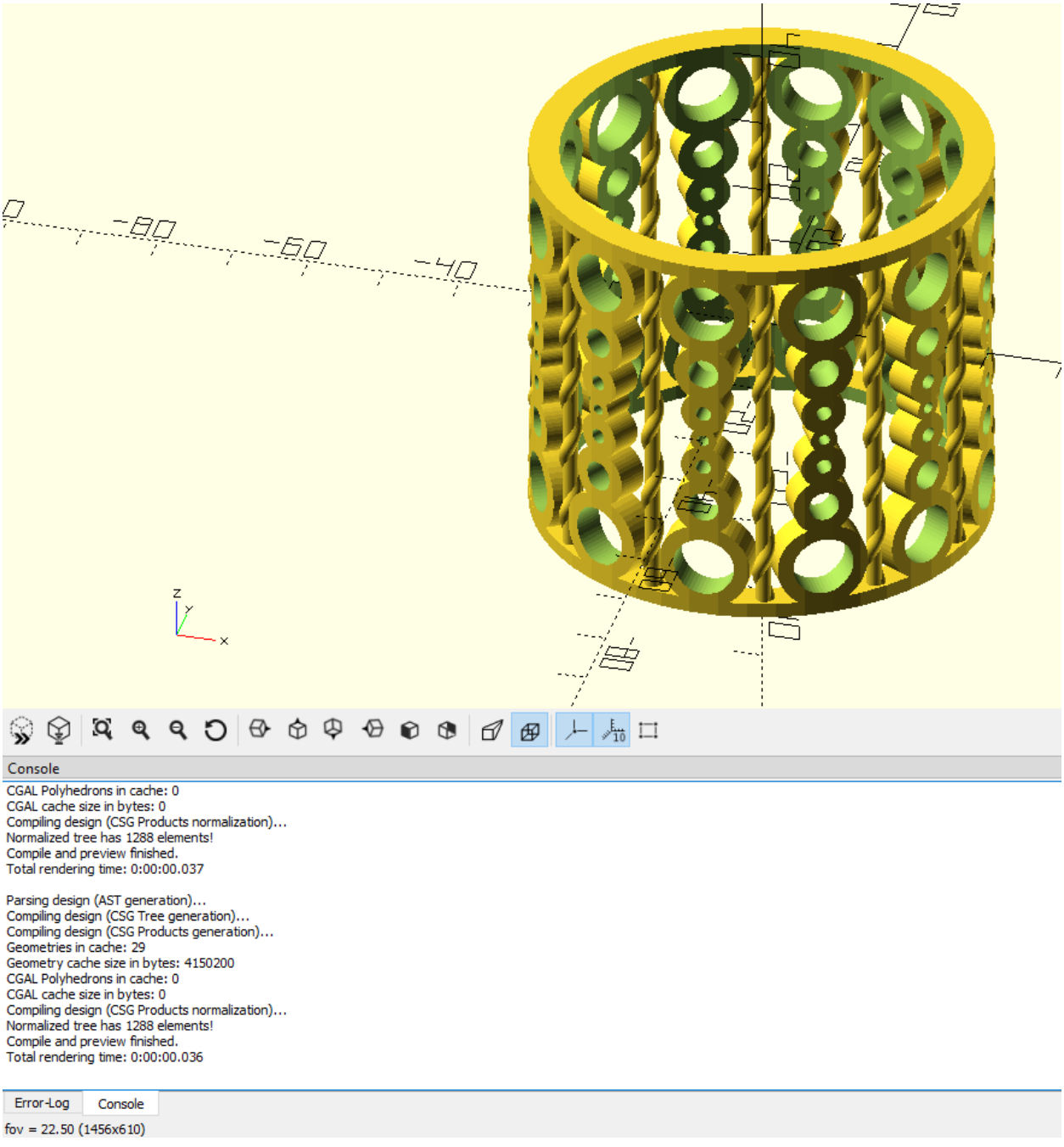
Tea Light Lamp Shade Two Piece Collection



This piece is designed to be used either separately or together, the exterior stacking on top of the interior lamp shade. Video will demonstrate more clearly.

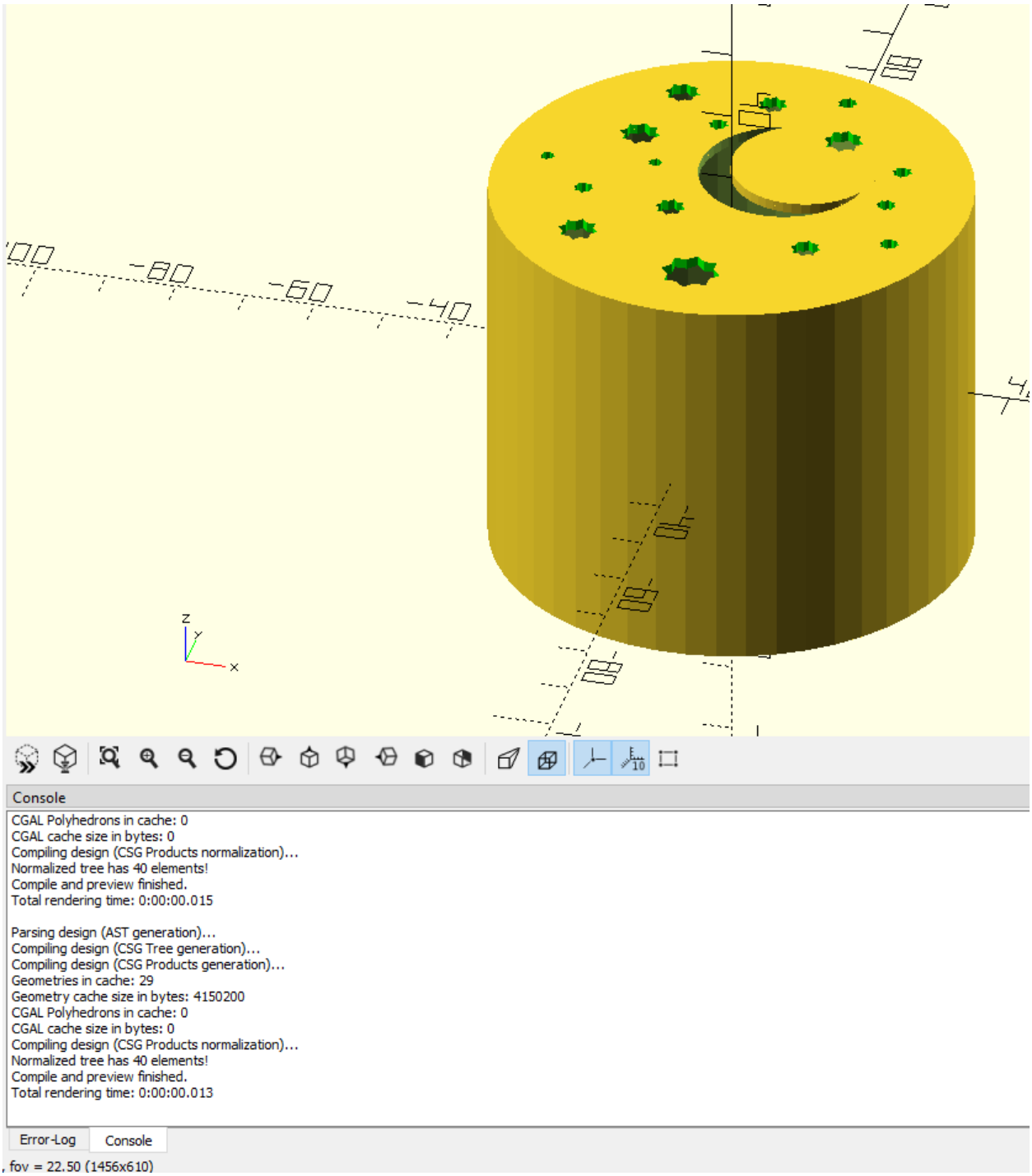
Section 1

Rendering of Lamp Shade Interior in OpenSCAD



Number of Elements: 1280
Geometries: 29
Cache size in bytes: 4150200

Rendering of Lamp Shade Exterior in OpenSCAD



Number of elements: 40
Geometries: 29
Cache size in bytes: 4105200

Section 2

The Tea Light Candle Lamp Shade Interior

There are three main shapes that are used for this CAD: the cylinder, the line and the spiral. The main functions that I used were difference, intersection, translate, and rotate. The main body was created by using the difference function. The rings that go around are created by using the intersection function. The rows were created by using the rotate and the translate function.

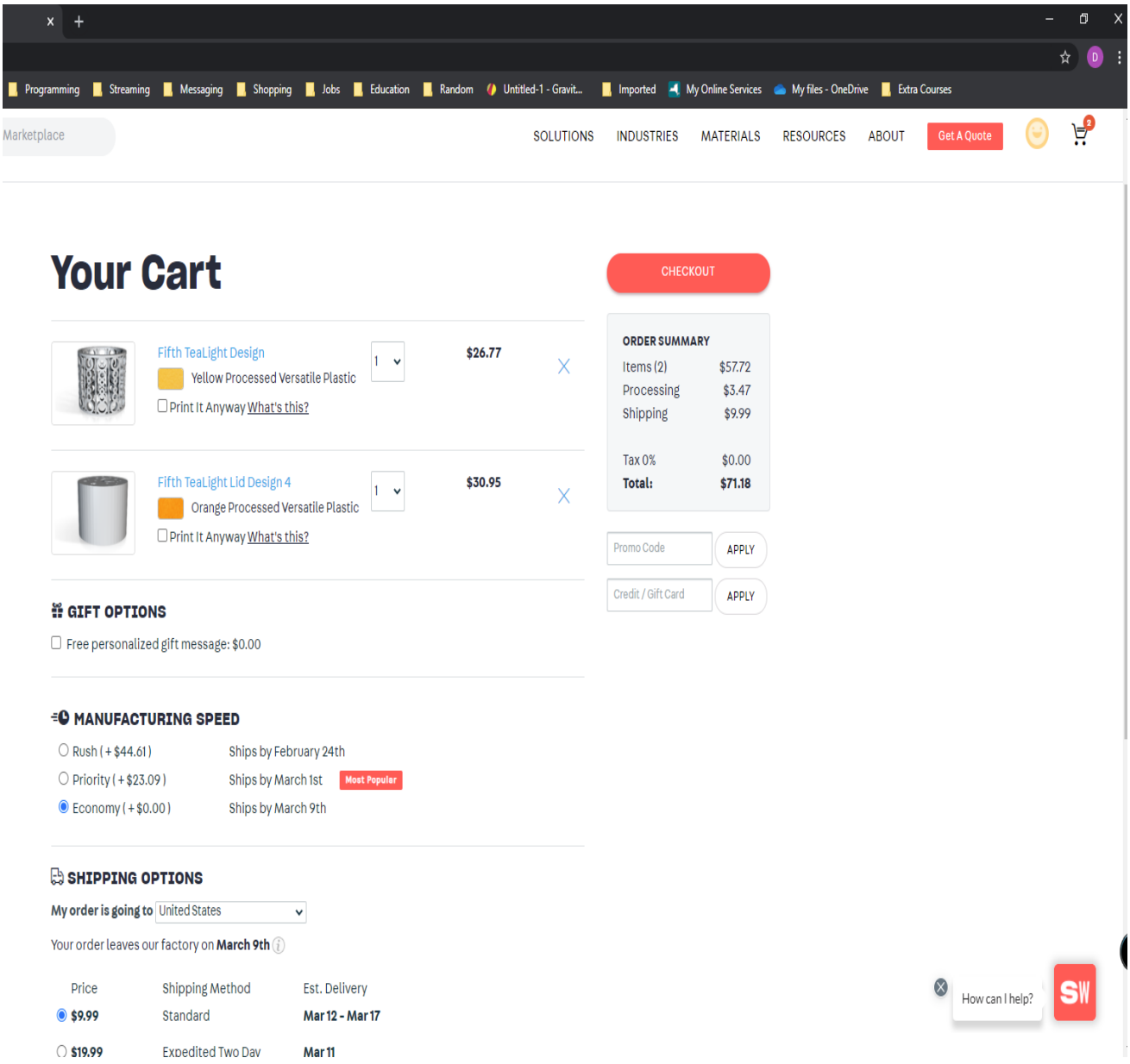
First I created a large cylinder and then hollowed it out by using the difference function and subtracting out a smaller cylinder. That created the body. Then I used a for loop to create a number of cylinders that generated around a single axis. I, then, rotated those cylinders 90°s so that it was intersecting the hollow cylinder. After that, I translated another set of cylinders rotated 90°s from the same axis down and repeated that to get rows of rotated cylinders that went around the circumference of the hollow cylinder. The top and bottom rotated cylinders are the same size and the same distance from each other. That pattern is true for each corresponding row and finally, a center row that houses the smallest rings. The rings themselves were created using the intersection function which only left the rings that was created from intersecting the rotated cylinders and the hollow cylinder. Then I translated two more cylinders to the top and the bottom of the rings to create a base. That led to the interior lamp shade.

The Tea Light Candle Lamp Shade Exterior

The main shapes used are cubes and cylinders. The main function used were the difference and union function. The main body was used using the difference function, the stars were made using cubes, the union function, and the difference function, and the moon was created using cylinders and the difference function.

First the cylinders were created by generating a large cylinder and then using the difference function to subtract a smaller cylinder to create a hollow cylinder. Then another cylinder was created as the lid/top of the hollow cylinder. Then I used the union function to combine one cube with another cube rotated by 45° using the rotate function at the same axis. Then extended that past the height of the cylinder and used the difference function to punch holes into the lid, creating the stars. This was done multiple times and was moved around using the translate function. The moon was created using the same method as the stars. Two cylinders were extended past the height of the cylinder through the lid and then the difference function was used to punch a hole through the lid. The crescent shape was obtained by moving the smaller cylinder around using the translate function until the ideal shape was created.

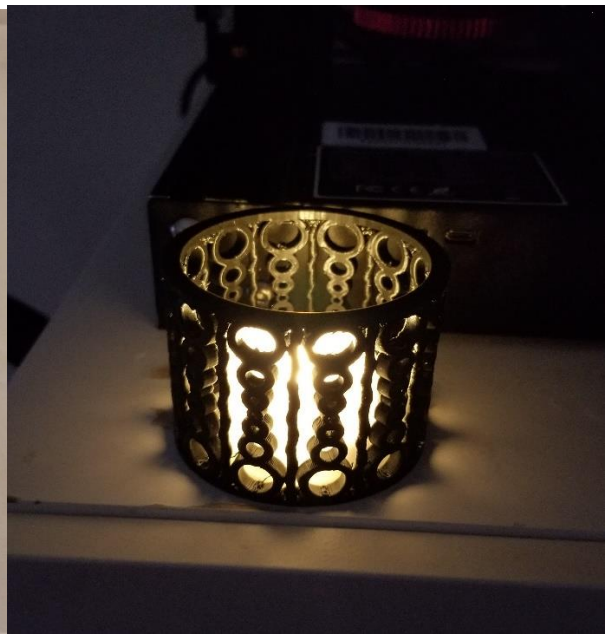
Section 3



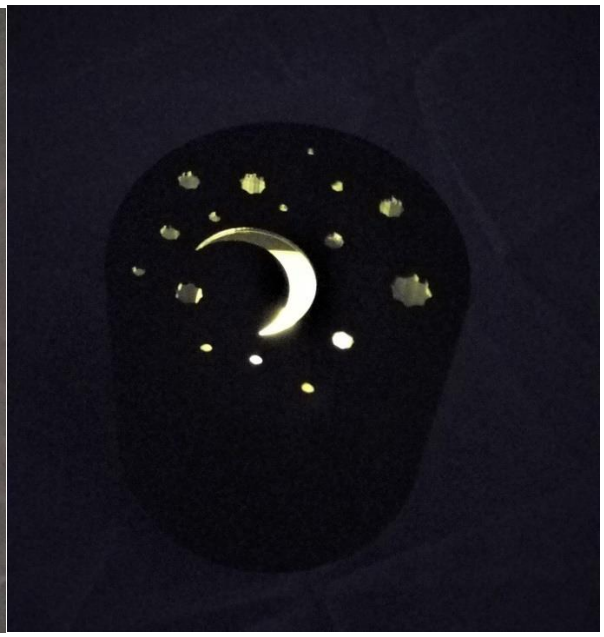
In Shapeway, I chose to use Versatile Plastic for both the interior and exterior lamp shade. The cost to manufacture both of these **was a total of $71.18** minus shipping and handling. I chose Versatile Plastic because it gave a quick finish and it was relatively cheap. My prints were able to be printed in this material and made costs low. Color selection has little impact on the cost, it would increase or decrease the total cost by about $2 either way.

Section 4

Tea Light Lamp Shade Interior



Tea Light Lamp Shade Exterior



Appendix

Code for Tea Light Candle Lamp Shade Interior

```
$fn = 50;

difference(){
intersection(){
difference(){
cylinder(h = 50, r = 30, center = true);
cylinder(h = 51, r = 26, center = true);
}

for (n = [1:12])
   {
      rotate([0,0,n*30])
      {
         translate([0, 0, 19])rotate([0, 90, 0])cylinder(h = 65, r = 6, center = true);
         translate([0, 0, -19])rotate([0, 90, 0])cylinder(h = 65, r = 6, center = true);
      }
   }
}
for (n = [1:12])
   {
      rotate([0,0,n*30])
      {
         translate([0, 0, 19])rotate([0, 90, 0])cylinder(h = 66, r = 4, center = true);
         translate([0, 0, -19])rotate([0, 90, 0])cylinder(h = 66, r = 4, center = true);
      }
   }
}

difference(){
intersection(){
difference(){
cylinder(h = 50, r = 30, center = true);
cylinder(h = 51, r = 26, center = true);
}
for (n = [1:12])
   {
      rotate([0,0,n*30])
      {
         translate([0, 0, 10])rotate([0, 90, 0])cylinder(h = 65, r = 4, center = true);
         translate([0, 0, -10])rotate([0, 90, 0])cylinder(h = 65, r = 4, center = true);
      }
```

```
   }
}
for (n = [1:12])
   {
      rotate([0,0,n*30])
      {
         translate([0, 0, 10])rotate([0, 90, 0])cylinder(h = 66, r = 2, center = true);
         translate([0, 0, -10])rotate([0, 90, 0])cylinder(h = 66, r = 2, center = true);
      }
   }
}

difference(){
intersection(){
difference(){
cylinder(h = 50, r = 30, center = true);
cylinder(h = 51, r = 26, center = true);
}
   for (n = [1:12])
   {
      rotate([0,0,n*30])
      {
         translate([0, 0, 4])rotate([0, 90, 0])cylinder(h = 65, r = 3, center = true);
         translate([0, 0, -4])rotate([0, 90, 0])cylinder(h = 65, r = 3, center = true);
      }
   }
}
for (n = [1:12])
   {
      rotate([0,0,n*30])
      {
         translate([0, 0, 4])rotate([0, 90, 0])cylinder(h = 66, r = 1, center = true);
         translate([0, 0, -4])rotate([0, 90, 0])cylinder(h = 66, r = 1, center = true);
      }
   }
}

difference(){
intersection(){
difference(){
cylinder(h = 50, r = 30, center = true);
cylinder(h = 51, r = 26, center = true);
}
   for (n = [1:12])
   {
      rotate([0,0,n*30])
```

```
        {
            //translate([0, 0, 4])rotate([0, 90, 0])cylinder(h = 65, r = 1.5, center = true);
            rotate([0, 90, 0])cylinder(h = 66, r = 1.75, center = true);
            //translate([0, 0, -4])rotate([0, 90, 0])cylinder(h = 65, r = 1.5, center = true);
        }
    }
}
for (n = [1:12])
    {
        rotate([0,0,n*30])
        {
            //translate([0, 0, 4])rotate([0, 90, 0])cylinder(h = 66, r = .5, center = true);
            rotate([0, 90, 0])cylinder(h = 66, r = 0.75, center = true);
            //translate([0, 0, -4])rotate([0, 90, 0])cylinder(h = 66, r = .5, center = true);
        }
    }
}

difference(){
    translate([0, 0, 25])cylinder(h = 2, r = 30, center = true);
    translate([0, 0, 25])cylinder(h = 3, r = 26, center = true);
}

difference(){
    translate([0, 0, -25])cylinder(h = 2, r = 30, center = true);
    translate([0, 0, -25])cylinder(h = 3, r = 26, center = true);
}

translate([28.5 * sin(15), 28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
    circle(r = 1);
}

translate([28.5 * sin(15), 28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
    circle(r = 1);
}
```

```
translate([-28.5 * sin(15), 28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(15), 28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(15), -28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(15), -28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(15), -28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}
```

```
translate([-28.5 * sin(15), -28.5 * sin(90 - 15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(90 - 15), 28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(90 - 15), 28.5* sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(90 - 15), 28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(90 - 15), 28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}
```

```
translate([28.5 * sin(90 - 15), -28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(90 - 15), -28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(90 - 15), -28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(90 - 15), -28.5 * sin(15), 0])linear_extrude(height = 50, center = true,
convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(45), 28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity =
10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}
```

```
translate([28.5 * sin(45), 28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(45), 28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(45), 28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(45), -28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([28.5 * sin(45), -28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}
```

```
translate([-28.5 * sin(45), -28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0, 0, 0])

difference()
{
   circle(r = 1);
}

translate([-28.5 * sin(45), -28.5 * sin(45), 0])linear_extrude(height = 50, center = true, convexity = 10, twist = -2000)
translate([0.5, 0, 0])

difference()
{
   circle(r = 1);
}
```

Code for Lamp Shade Exterior

```
$fn = 50;

difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
difference(){
translate([0, 0, 27.5])cylinder(h = 1.5, r = 34, center = true);

difference(){
translate([6, 6, 0])cylinder(h=65, r = 12, center = true);
translate([8.5, 8.5, 0])cylinder(h=65, r = 10.5, center = true);
}
}
translate([-7, -7, 0])union(){
color([0, 1, 0])cube([3, 3, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([3,3,65], center = true);
}
}

translate([-16, 11, 0])union(){
color([0, 1, 0])cube([4, 4, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([4,4,65], center = true);
}
}

translate([14, -13, 0])union(){
color([0, 1, 0])cube([3, 3, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([3,3,65], center = true);
}
}
```

```
translate([0, 23, 0])union(){
color([0, 1, 0])cube([3, 3, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([3, 3, 65], center = true);
}
}

translate([0, -23, 0])union(){
color([0, 1, 0])cube([6, 6, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([6, 6, 65], center = true);
}
}

translate([22, 1, 0])union(){
color([0, 1, 0])cube([2, 2, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([2, 2, 65], center = true);
}
}

translate([-20, -5, 0])union(){
color([0, 1, 0])cube([2, 2, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([2, 2, 65], center = true);
}
}

translate([12, 16, 0])union(){
color([0, 1, 0])cube([4, 4, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([4, 4, 65], center = true);
}
}

translate([-18, -16, 0])union(){
color([0, 1, 0])cube([4, 4, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([4, 4, 65], center = true);
}
}

translate([-12, 4, 0])union(){
color([0, 1, 0])cube([1.5, 1.5, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([1.5, 1.5, 65], center = true);
}
}

translate([-27, 2, 0])union(){
color([0, 1, 0])cube([1.5, 1.5, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([1.5, 1.5, 65], center = true);
}
```

```
}

translate([-13, 23, 0])union(){
color([0, 1, 0])cube([3.5, 3.5, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([3.5, 3.5, 65], center = true);
}
}

translate([22, 10, 0])union(){
color([0, 1, 0])cube([2, 2, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([2, 2, 65], center = true);
}
}

translate([-6, 16, 0])union(){
color([0, 1, 0])cube([2, 2, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([2, 2, 65], center = true);
}
}

translate([10, 26, 0])union(){
color([0, 1, 0])cube([2, 2, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([2, 2, 65], center = true);
}
}

translate([25, -9, 0])union(){
color([0, 1, 0])cube([2, 2, 65], center = true);
color([0, 1, 0])rotate([0, 0, 45])cube([2, 2, 65], center = true);
}
}

difference(){
        cylinder(h = 55, r = 34, center = true);
        cylinder(h = 55.5, r = 32.5, center = true);
}
```