```
 1: //CPSC 1045
 2: //Example solution lab 6:
 3: //Do not copy
 4: //Author: Kim Lam
 5:
 6: //Convert degrees to radians
 7: /*global
 8:    console, prompt
 9: */
10: function toRadians(angleDeg) {
11:     "use strict";
12:     if (isNaN(angleDeg)) {
13:         console.log("solnError, toRadians: angleDeg is not a number. The cats are mad.");
14:     }
15:     return Number(angleDeg) * Math.PI / 180;
16: }
17:
18: //Check is a input is in range, and is a number
19: function isValidNum(inputNum, validMin, validMax) {
20:     "use strict";
21:     if (isNaN(inputNum)) {
22:         console.log("inputNum is not a number! The cats are sad.");
23:     }
24:     //returning the boolean result directly.  You could also use an if else block.
25:     return (!isNaN(inputNum) && (validMin <= inputNum) && (inputNum <= validMax));
26: }
27:
28: //Checking if a number is valid.
29: function drawCat(ctx, x, y) {
30:     "use strict";
31:     var i;
32:     ctx.save();
33:     ctx.translate(x, y);
34:
35:
36:
37:     //ears
38:     for (i = -50; i >= -130; i -= 80) {
39:         ctx.save();
40:         ctx.rotate(toRadians(i));
41:         ctx.translate(50, 0);
42:         ctx.fillRect(-10, -10, 20, 20);
43:         ctx.restore();
44:     }
45:
46:     //face
47:     ctx.beginPath();
48:     ctx.arc(0, 0, 50, 0, 2 * Math.PI);
```

```
49:     ctx.fillStyle = "gray";
50:     ctx.fill();
51:     //eyes Left
52:     ctx.beginPath();
53:     ctx.arc(-20, -30, 5, 0, 2 * Math.PI);
54:     ctx.fillStyle = "black";
55:     ctx.fill();
56:
57:     //eyes Right
58:     ctx.beginPath();
59:     ctx.arc(20, -30, 5, 0, 2 * Math.PI);
60:     ctx.fillStyle = "black";
61:     ctx.fill();
62:
63:     //Nose
64:     ctx.beginPath();
65:     ctx.arc(0, 0, 5, 0, 2 * Math.PI);
66:     ctx.fillStyle = "black";
67:     ctx.fill();
68:
69:     //Whiskers
70:     ctx.rotate(toRadians(-15));
71:     for (i = 0; i < 3; i = i + 1) {
72:         ctx.beginPath();
73:         ctx.lineTo(-70, 0);
74:         ctx.lineTo(70, 0);
75:         ctx.stroke();
76:         ctx.rotate(toRadians(15));
77:     }
78:     ctx.restore();
79: }
80:
81: //Gather the number of cats
82: var numCats = NaN;
83: var i;
84: while (!isValidNum(numCats, 1, 10)) {
85:     numCats = prompt("How many cats would you like? ( 1 to 10)");
86:     numCats = Number(numCats);
87: }
88:
89: //Obtain the radius
90: var radius = NaN;
91: while (!isValidNum(radius, 0, 200)) {
92:     radius = prompt("What is the radius(0 to 200)?");
93:     radius = Number(radius);
94: }
95: var canvas = document.getElementById("solnSurface");
96: var ctx = document.getElementById("solnSurface").getContext("2d");
```

```
 97: //First move our local coordinate system to the
 98: //center of the canvas.
 99: ctx.translate(canvas.width / 2, canvas.height / 2);
100: //Use a loop to draw all the cats requested.
101: for (i = 0; i < numCats; i = i + 1) {
102:     //Rotate on every itration.
103:     ctx.rotate(toRadians(360 / numCats));
104:     drawCat(ctx, radius, 0);
105: }
```