```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text.RegularExpressions;
5
6  public class CSharpTutorial {
7      static void Main() {
8          // instantiate an array to hold our shapes
9          Shape[] a = new Shape[3];
10         // instantiate some shapes into the array
11         a[0] = new Rectangle(0, 0, 3, 4);
12         a[1] = new Triangle(7, 7, 3, 4, 90);
13         a[2] = new Circle(9, 9, 2);
14         // draw them
15         for (int i=0; i < a.Length; i++) {
16             a[i].Draw();
17         }
18     }
19 }
20 abstract class Shape {
21     public int x; public int y;
22     public Shape() {
23         this.x = 0; this.y = 0;
24     }
25     public Shape(int x, int y) {
26         this.x = x; this.y = y;
27     }
28     abstract public void Draw();
29 }
30 class Rectangle : Shape {
31     int width; int height;
32     public Rectangle(int x, int y, int w, int h) : base(x, y) {
33         this.width = w; this.height = h;
34     }
35     public override void Draw() {
36         Console.WriteLine("Rectangle: " + width + " by " + height + " at " + x + "," +
   y);
37     }
38 }
39 class Triangle : Shape {
40     int side1; int side2; int angle;
41     public Triangle(int x, int y, int s1, int s2, int a) : base(x, y) {
42         this.side1 = s1; this.side2 = s2; this.angle = a;
43     }
44     public override void Draw() {
45         Console.WriteLine("Triangle: side1=" + side1 + " side2=" + side2 + " angle=" + a
   ngle + " at " + x + "," + y);
46     }
47 }
48 class Circle : Shape {
49     int radius;
50     public Circle(int x, int y, int r) : base(x, y) {
51         this.radius = r;
52     }
53     public override void Draw() {
54         Console.WriteLine("Circle:  radius=" + radius + " at " + x + "," + y);
55     }
56 }
```