

```
1 using System;
2
3 namespace OO_Tasks_Inheritance {
4     public abstract class Shape {
5         private string _name;
6         private string _colour;
7
8         public Shape (string name, string colour)
9         {
10             _name = name;
11             _colour = colour;
12         }
13     }
14
15     public abstract class Quadrilateral : Shape{
16         public const int _numsides = 4;
17
18         public Quadrilateral(string name, string colour) : base(name, colour)
19         {
20         }
21     }
22
23     public class Square : Quadrilateral{
24         private double _sidelength;
25
26         public double GetArea()
27         {
28             double area = Math.Round(_sidelength * _sidelength, 2);
29             return area;
30         }
31
32         public double GetPerimeter()
33         {
34             double perimeter = Math.Round(_sidelength * _numsides, 2);
35             return perimeter;
36         }
37
38         public Square(string name, string colour, double sidelength) : base  ➤
39             (name, colour)
40         {
41             _sidelength = sidelength;
42         }
43     }
44
45     public class Rectangle : Quadrilateral {
46         private double _length;
47         private double _width;
48
49         public double GetArea()
50         {
51             double area = Math.Round(_length * _width, 2);
52             return area;
53         }
54
55         public double GetPerimeter()
56         {
```

```
56         double perimeter = Math.Round((_length * 2) + (_width * 2), 2);
57         return perimeter;
58     }
59
60     public Rectangle(string name, string colour, double length, double width) : base(name, colour)
61     {
62         _length = length;
63         _width = width;
64     }
65 }
66
67 public class Circle : Shape{
68     public const double PI = 3.142;
69     private double _radius;
70
71     public double GetArea()
72     {
73         double area = Math.Round(PI * (_radius * _radius), 2);
74         return area;
75     }
76
77     public double GetCircumference()
78     {
79         double circumference = Math.Round(2 * PI * _radius, 2);
80         return circumference;
81     }
82
83     public double GetPerimeter()
84     {
85         double perimeter = Math.Round(2 * PI * _radius, 2);
86         return perimeter;
87     }
88
89     public Circle(string name, string colour, double radius) : base(name, colour)
90     {
91         _radius = radius;
92     }
93 }
94
95
96 }
```