

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SplashKitSDK;
using ShapeDrawing;

namespace ShapeDrawing
{
    public class MyCircle : Shape
    {
        //Set attributes
        private int _radius;
        //Set constructors and methods
        public int Radius
        {
            get
            {
                return _radius;
            }
            set
            {
                _radius = value;
            }
        }
        public MyCircle() : this(SplashKitSDK.Color.White, 100, 100, 50)
        {
        }
        public MyCircle(SplashKitSDK.Color color, float x, float y, int radius) : base(color)
        {
            X = x;
            Y = y;
            _radius = radius;
        }
        public override void Draw()
        {
            if (Selected)
            {
                DrawOutline();
            }
            SplashKit.FillCircle(Color, X, Y, Radius);
        }
        public override void DrawOutline()
        {
            SplashKit.FillCircle(SplashKitSDK.Color.Black, X, Y, Radius + 2);
        }
    }
}
```

```
public override bool IsAt(Point2D pt)
{
    return SplashKit.PointInCircle(pt, SplashKit.CircleAt(X, Y,
        _radius));
}
//Step 7 of task 5.3C
public override void SaveTo(StreamWriter writer)
{
    writer.WriteLine("Circle");
    base.SaveTo(writer);
    writer.WriteLine(Radius);
}
//Step 15 of task 5.3C
public override void LoadFrom(StreamReader reader)
{
    base.LoadFrom(reader);
    Radius = reader.ReadInteger();
}
}
}
```