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

namespace ShapeDrawer
{

    //Step 1
    public class Shape
    {
        //Set the variable color (Step 1.1)
        private SplashKitSDK.Color _color;
        private float _x;
        private float _y;
        private int _width;
        private int _height;

        //Step 1.2 - Constructor adapted to requirements
        public Shape(int param)
        {
            // IMPORTANT: Replace 'H' with the actual first letter of your
              first name.
            // Ensure it's an uppercase letter for consistent comparison (e.
              g., 'A', 'K', 'S', etc.)
            char FirstName = 'H'; // <--- VERIFY THIS IS YOUR ACTUAL FIRST
              NAME'S INITIAL

            // FIX: Changed single '&' to '&&' for logical AND
            if (FirstName >= 'A' && FirstName <= 'K')
            {
                _color = SplashKitSDK.Color.Azure;
            }
            else
            {
                _color = SplashKitSDK.Color.Chocolate;
            }

            _x = 0.0f;
            _y = 0.0f;
            _width = param;
            _height = param;
        }

        //Step 1.3 - Get Color <<property>>
        public SplashKitSDK.Color Color
        {
            get { return _color; }
            set { _color = value; }
        }

        //Step 1.4 - Set X: Float <<property>>
        public float X
        {
            get { return _x; }
            set { _x = value; }
        }

        //Step 1.5 - Set Y: Float <<property>>
        public float Y
        {
            get { return _y; }
            set { _y = value; }
        }

        //Step 1.6 - Set Width: Int <<property>>
        public int Width
        {
            get { return _width; }
            set { _width = value; }
        }

        //Step 1.7 - Set Height: Int <<property>>
        public int Height
        {
            get { return _height; }
            set { _height = value; }
        }

        //Step 1.8 - Set draw() method
        public void Draw()
        {
            // FIX: Removed redundant Console.WriteLine as the requirement is to draw visually.
            SplashKit.FillRectangle(_color, _x, _y, _width, _height);
        }

        //Step 1.9 - The IsAt() method - Adapted to use Point2D struct
        public virtual bool IsAt(Point2D pt)
        {
            return pt.X >= X && pt.X <= (X + Width) &&
                   pt.Y >= Y && pt.Y <= (Y + Height);
        }
    }
}
```