

```

#include <graphics.h>
#include <iostream>

using namespace std;

class Shape {
public:
    // Virtual function to be overridden in derived classes
    virtual void draw() = 0;
    virtual void fillColor(int color) = 0;
};

class ConcavePolygon : public Shape {
private:
    int n;           // Number of vertices
    int points[20]; // Array to hold the points of the polygon

public:
    ConcavePolygon(int vertices, int pts[]) {
        n = vertices;
        for (int i = 0; i < 2 * n; i++) {
            points[i] = pts[i];
        }
    }

    // Function to draw the concave polygon
    void draw() override {
        setcolor(WHITE);
        fillpoly(n, points);
    }

    // Function to apply the scan fill algorithm
    void fillColor(int fillColor) override {
        int maxY = getmaxy();
        int x, y;

        for (y = 0; y <= maxY; y++) {
            bool inside = false;
            for (x = 0; x < getmaxx(); x++) {
                if (getpixel(x, y) == WHITE) {
                    inside = !inside;
                }
                if (inside) {
                    putpixel(x, y, fillColor);
                }
            }
        }
    }
};

int main() {
    // Initialize graphics mode

```

```

int gd = DETECT, gm;
initgraph(&gd, &gm, const_cast<char*>("/"));

// Define points of the concave polygon
int points[] = {150, 200, 250, 100, 350, 200, 300, 250, 200,
250};

// Create ConcavePolygon object
ConcavePolygon poly(5, points);

// Draw the concave polygon
poly.draw();

// Fill the polygon with desired color using scan fill
algorithm
poly.fillColor(RED);

// Wait for the user to close the window
getch();
closegraph();
return 0;
}
OUTPUT:

```