#include <iostream>

#include <graphics.h>

#include <math.h>

using namespace std;

class Tdimen

{

int x1, y1, x2, y2, x3, y3;

public:

void translate();

void scale();

void rotate();

// Operator overloading for assignment

void operator=(const Tdimen &T)

{

x1 = T.x1;

y1 = T.y1;

x2 = T.x2;

y2 = T.y2;

x3 = T.x3;

y3 = T.y3;

}

// Draw the initial triangle

void draw()

{

// Define the triangle's initial coordinates

x1 = 150;

y1 = 50;

x2 = 100;

y2 = 100;

x3 = 200;

y3 = 100;

// Draw the triangle

cleardevice();

line(x1, y1, x2, y2);

line(x2, y2, x3, y3);

line(x3, y3, x1, y1);

}

// Result to redraw the updated triangle

void result()

{

cleardevice();

line(x1, y1, x2, y2);

line(x2, y2, x3, y3);

line(x3, y3, x1, y1);

}

};

// Translate function to move the triangle

void Tdimen::translate()

{

float tx, ty;

cout << "Enter tx & ty: ";

cin >> tx >> ty;

x1 += tx;

x2 += tx;

x3 += tx;

y1 += ty;

y2 += ty;

y3 += ty;

}

// Scale function to resize the triangle

void Tdimen::scale()

{

float sx, sy;

cout << "Enter sx & sy: ";

cin >> sx >> sy;

x1 \*= sx;

x2 \*= sx;

x3 \*= sx;

y1 \*= sy;

y2 \*= sy;

y3 \*= sy;

}

// Rotate function to rotate the triangle

void Tdimen::rotate()

{

float deg;

cout << "Enter Angle (in degrees): ";

cin >> deg;

// Convert degrees to radians

float rad = deg \* 3.14159 / 180;

// Store original coordinates to use in calculations

int x, y;

// Rotate point 1

x = x1;

y = y1;

x1 = x \* cos(rad) - y \* sin(rad);

y1 = x \* sin(rad) + y \* cos(rad);

// Rotate point 2

x = x2;

y = y2;

x2 = x \* cos(rad) - y \* sin(rad);

y2 = x \* sin(rad) + y \* cos(rad);

// Rotate point 3

x = x3;

y = y3;

x3 = x \* cos(rad) - y \* sin(rad);

y3 = x \* sin(rad) + y \* cos(rad);

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, NULL); // Initialize graphics mode

Tdimen T;

int ch;

// Loop to display menu and perform transformations

do

{

cout << "\nEnter your choice!";

cout << "\n1. Translate \n2. Scaling \n3. Rotation \n4. Exit\n";

cin >> ch;

switch (ch)

{

case 1:

T.draw(); // Draw original triangle

T.translate(); // Perform translation

T.result(); // Display the result

break;

case 2:

T.draw(); // Draw original triangle

T.scale(); // Perform scaling

T.result(); // Display the result

break;

case 3:

T.draw(); // Draw original triangle

T.rotate(); // Perform rotation

T.result(); // Display the result

break;

case 4:

break;

default:

cout << "Sorry, Invalid Choice!";

}

} while (ch != 4);

closegraph(); // Close the graphics window

return 0;

}