Assignment No : 2_4

Problem Statement :

Write C++ program to draw 2-D object and perform following basic transformations, Scaling , Translation, Rotation. Apply the concept of operator overloading.

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <stdlib.h>
#include <math.h>

class trans
{
public:
    float transco[3][3];
    // float orico[3][3];
    float scalco[3][3];
    float rotco[3][3];

    void drawtri(float[3][3]);
    void translation(int, int, float[3][3]);
    void scaling(float, float, float[3][3]);
    void rotation(float, float[3][3]);
};

void trans::drawtri(float co[3][3])
{
    // clrscr();
    line(co[0][0], co[1][0], co[0][1], co[1][1]);
    line(co[0][1], co[1][1], co[0][2], co[1][2]);
    line(co[0][2], co[1][2], co[0][0], co[1][0]);
}
void trans::translation(int tx, int ty, float orico[3][3])

{

    cout << "Enter Translation Factor" << endl;
    cin >> tx >> ty;
    int i, j;

    for (i = 0; i < 3; i++)
    {
```

```
        transco[0][i] = orico[0][i] + tx;
        transco[1][i] = orico[1][i] + ty;
        transco[2][i] = 1;
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            cout << transco[i][j] << " ";
        }
        cout << endl;
    }
}
void trans::scaling(float sx, float sy, float orico[3][3])
{

    cout << "Enter Scaling Factor" << endl;
    cin >> sx >> sy;
    int i, j;

    for (i = 0; i < 3; i++)
    {
        scalco[0][i] = orico[0][i] * sx;
        scalco[1][i] = orico[1][i] * sy;
        scalco[2][i] = 1;
    }

    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            cout << scalco[i][j] << " ";
        }
        cout << endl;
    }
}
void trans::rotation(float theta, float orico[3][3])
{

    cout << "Enter Rotation Angle" << endl;
    cin >> theta;
    cout << theta << endl;

    theta = theta * (3.14 / 180);
    cout << "theta in radious" << theta << endl;
    int i, j, refx, refy;
    for (i = 0; i < 3; i++)
```

```
    {
        for (j = 0; j < 3; j++)
        {

            rotco[i][j] = 0;
        }
    }

    for (i = 0; i < 3; i++)
    {

        rotco[0][i] = orico[0][i] * cos(theta) -

                      orico[1][i] * sin(theta);

        rotco[1][i] = orico[0][i] * sin(theta) + orico[1][i] *
cos(theta);
    }
}
void main()
{
    clrscr();
    int c;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TurboC3\\BGI");
    trans t;
    int tx, ty;
    float sx, sy;
    float theta;
    float orico[3][3] = {{300, 250, 350}, {200, 300, 300}, {1, 1,
1}};
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << "ori"
                 << " " << i << " " << j << "->" << orico[i][j] << "

                                                                  "
;
        }
        cout << endl;
    }

    t.drawtri(orico);
    cout << "Enter your choice" << endl;
    cout << "1. Translation" << endl;
    cout << "2. Scaling" << endl;
```

```
    cout << "3. Rotation" << endl;
    cin >> c;
    switch (c)
    {

    case 1:

        t.translation(tx, ty, orico);
        t.drawtri(t.transco);
        break;
    case 2:
        t.scaling(sx, sy, orico);
        t.drawtri(t.scalco);
        break;
    case 3:

        t.rotation(theta, orico);
        t.drawtri(t.rotco);
        break;

    default:
        cout << ("You have written wrong choice");
    }
    getch();
}
```