

Lab Report: 03
Title: 2D Geometric Transformation – Translation, Rotation, Scaling

Course title: Computer Graphics Laboratory
Course code: CSE-304
3rd Year 1st Semester Examination 2022

Date of Submission: 11.06.2023



Submitted to-

Dr. Mohammad Shorif Uddin
Dr. Morium Akter

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka-1342

Sl	Class Roll	Exam Roll	Name
01	402		Jubaer Ahmad Khan

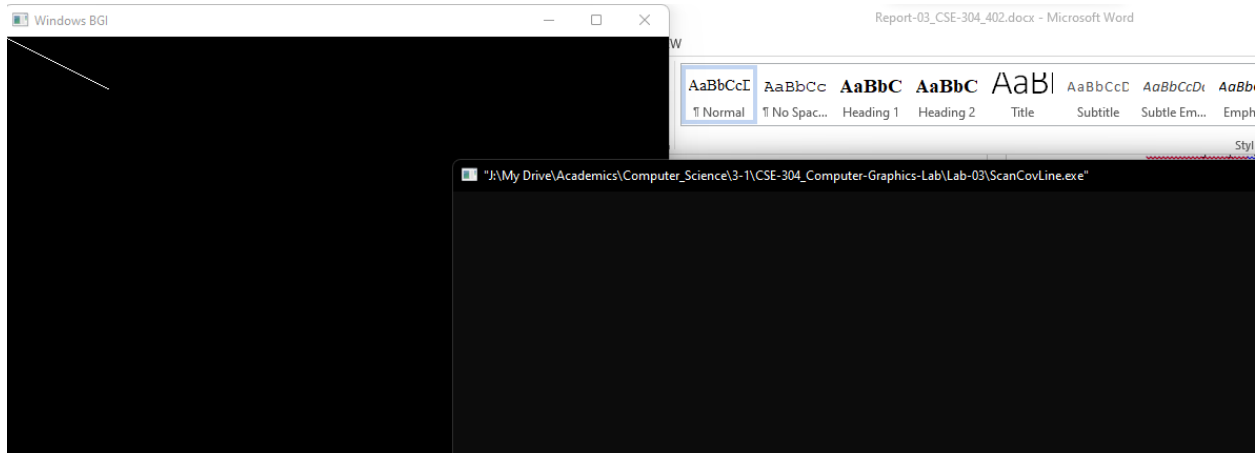
Serial No.	Title	Page No.
1	Scan Converting a Line	2 - 3
2	Rotating the line by 30 Degree	3 – 4
3	Scaling the line by 50%	5 – 6
4	Translating the line by 75px	6 - 7
5	Draw a kite using Bresenham Algorithm	7 - 8

Algorithm Title: Scan Converting a Line

Source Code:

```
#include <graphics.h>
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    // Scan Convert a line object from (0, 0) to (100, 50)
    line(0, 0, 100, 50);
    getch();
    closegraph();
    return 0;
}
```

GUI Output:

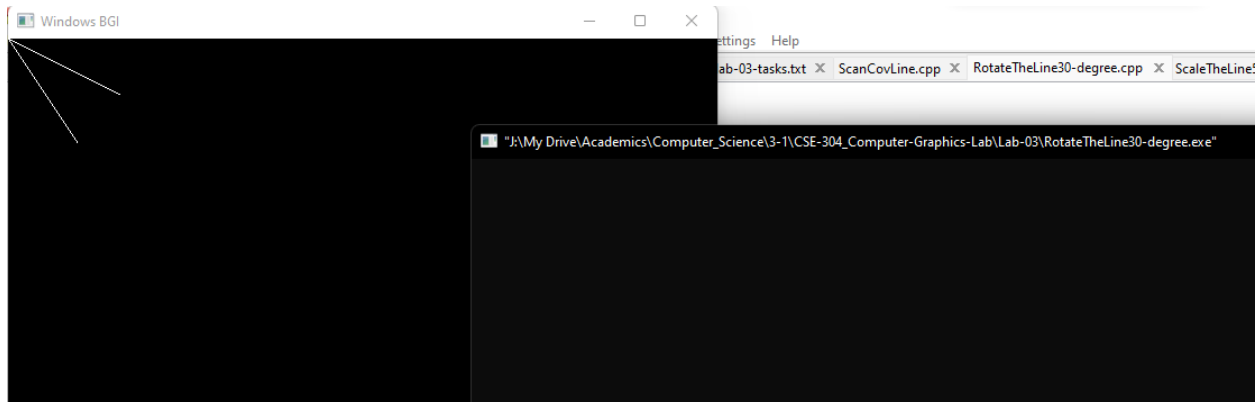


Algorithm Title: Rotating the line by 30 degree

Source Code:

```
#include <graphics.h>
#include <math.h>
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    // Scan Convert a line object from (0, 0) to (100, 50)
    line(0, 0, 100, 50);
    // Rotate the line by 30 degrees
    int angle = 30;
    float radian = angle * (M_PI / 180.0);
    int x0 = 0, y0 = 0, x1 = 100, y1 = 50;
    int new_x0 = round(x0 * cos(radian) - y0 * sin(radian));
    int new_y0 = round(x0 * sin(radian) + y0 * cos(radian));
    int new_x1 = round(x1 * cos(radian) - y1 * sin(radian));
    int new_y1 = round(x1 * sin(radian) + y1 * cos(radian));
    line(new_x0, new_y0, new_x1, new_y1);
    getch();
    closegraph();
    return 0;
}
```

GUI Output:

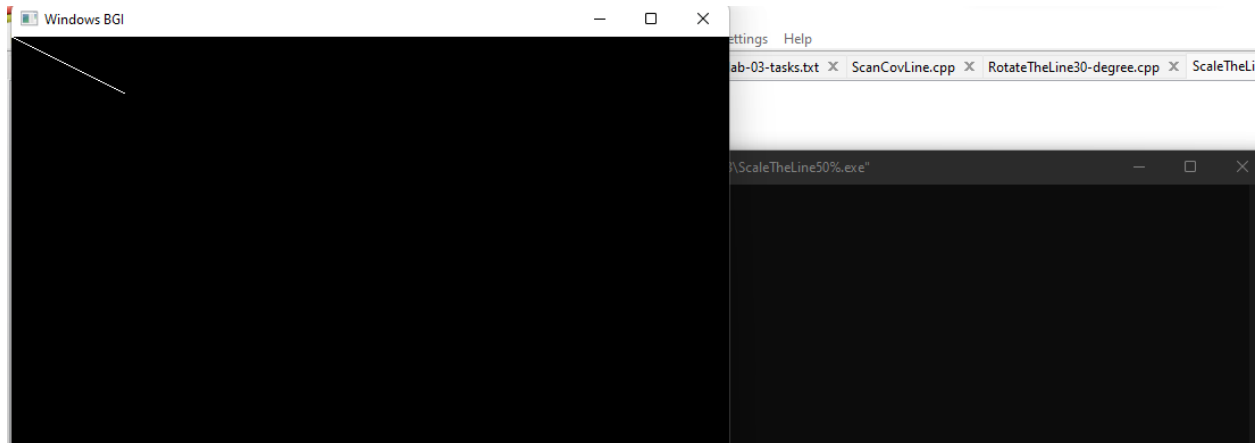


Algorithm Title: Scaling the line by 50%

Source Code:

```
#include <graphics.h>
#include <math.h>
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    // Scan Convert a line object from (0, 0) to (100, 50)
    line(0, 0, 100, 50);
    // Scale the line to 50%
    float scale = 0.5;
    int new_x0 = round(new_x0 * scale);
    int new_y0 = round(new_y0 * scale);
    int new_x1 = round(new_x1 * scale);
    int new_y1 = round(new_y1 * scale);
    line(new_x0, new_y0, new_x1, new_y1);
    getch();
    closegraph();
    return 0;
}
```

GUI Output:

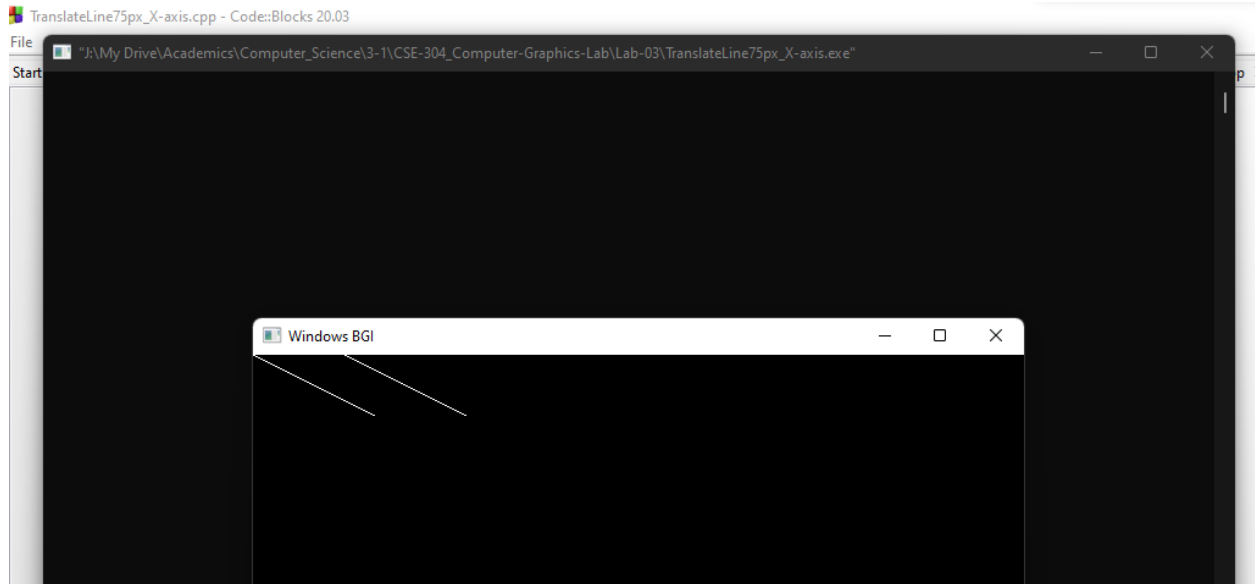


Algorithm Title: Translate the line x-axis by 75px

Source Code:

```
#include <graphics.h>
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    // Scan Convert a line object from (0, 0) to (100, 50)
    line(0, 0, 100, 50);
    // Translate the line on the x-axis by 75 pixels
    int x0, y0, x1, y1;
    x0 = 0; y0 = 0; x1 = 100; y1 = 50;
    int translation_x = 75;
    x0 += translation_x;
    x1 += translation_x;
    line(x0, y0, x1, y1);
    getch();
    closegraph();
    return 0;
}
```

GUI Output:



Algorithm Title: Draw a kite using lines

Source Code:

```
#include<time.h>
#include<graphics.h>
int main() {
    int gd = DETECT, gm;
    int x = 10, y = 480;
    initgraph(&gd, &gm, (char*) "");
    while(!kbhit()) {
        cleardevice();
        if(y==0) {
            y=rand()%480;
            x=rand()%640;
        }
        else {
            y=y-1;
            x=x+1;
            line(x-50, y, x, y-70);
            line(x, y-70, x+50, y);
            line(x+50, y, x, y+70);
            line(x, y+70, x-50, y);
            line(x, y-70, x, y+70);
        }
    }
}
```

```

        line(x,y+70,x+10,y+140);
        line(x,y+70,x-10,y+140);
        line(x-50,y,x+50,y);
        line(x,y,x+130,y+640);
    }
    delay(20);
}
closegraph();
restorecrtmode();
}

```

GUI Output:

