

**Department of Computer Science & Engineering(CSE)**

**Lab -08**

Name : Jabed Iqbal Joy

Student ID : C193049

Semester : 7th

Section : 7BM

Email : c193049@ugrad.iiuc.ac.bd

Contact : 01837844828

Course Code : CSE-4742

Course Title : Computer Graphics Lab

Name of the course Teacher :

**Mahadi Hassan**

Assistant Professor

Department of CSE, IIUC

Date of Submission **:** 11/05/23

1. Scaling a point about origin.

Code: #include <graphics.h>

#include <stdlib.h>

void scale\_point(int x1, int y1, float sx, float sy, int \*new\_x, int \*new\_y)

{

// Scale point

\*new\_x = x1 \* sx;

\*new\_y = y1 \* sy;

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

// Original point

int x1 = 100, y1 = 100;

circle(x1, y1, 3);

// Scaling factors

float sx = 2.0, sy = 3.0;

// Scale point

int new\_x, new\_y;

scale\_point(x1, y1, sx, sy, &new\_x, &new\_y);

// Display scaled point

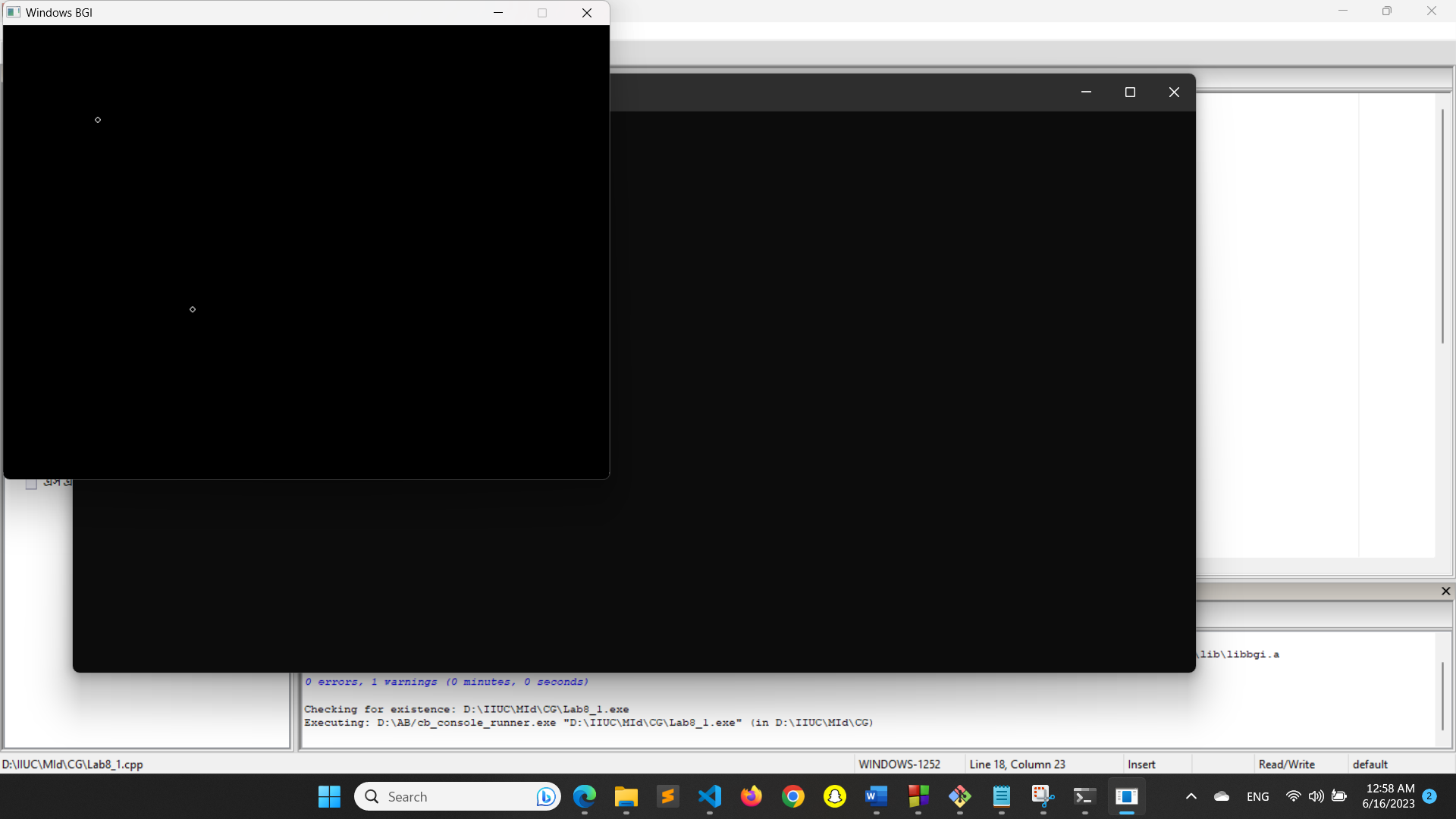
circle(new\_x, new\_y, 3);

getch();

closegraph();

return 0;

}



1. Scaling a triangle about origin.

Code:

#include <graphics.h>

#include <stdlib.h>

void scale\_triangle(int x1, int y1, int x2, int y2, int x3, int y3, float sx, float sy,

int \*new\_x1, int \*new\_y1, int \*new\_x2, int \*new\_y2, int \*new\_x3, int \*new\_y3)

{

// Scale points

\*new\_x1 = x1 \* sx;

\*new\_y1 = y1 \* sy;

\*new\_x2 = x2 \* sx;

\*new\_y2 = y2 \* sy;

\*new\_x3 = x3 \* sx;

\*new\_y3 = y3 \* sy;

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

// Original triangle

int x1 = 100, y1 = 100, x2 = 200, y2 = 200, x3 = 150, y3 = 50;

line(x1, y1, x2, y2);

line(x2, y2, x3, y3);

line(x3, y3, x1, y1);

// Scaling factors

float sx = 2.0, sy = 2.0;

// Scale triangle

int new\_x1, new\_y1, new\_x2, new\_y2, new\_x3, new\_y3;

scale\_triangle(x1, y1, x2, y2, x3, y3, sx, sy, &new\_x1, &new\_y1, &new\_x2, &new\_y2, &new\_x3, &new\_y3);

// Display scaled triangle

line(new\_x1, new\_y1, new\_x2, new\_y2);

line(new\_x2, new\_y2, new\_x3, new\_y3);

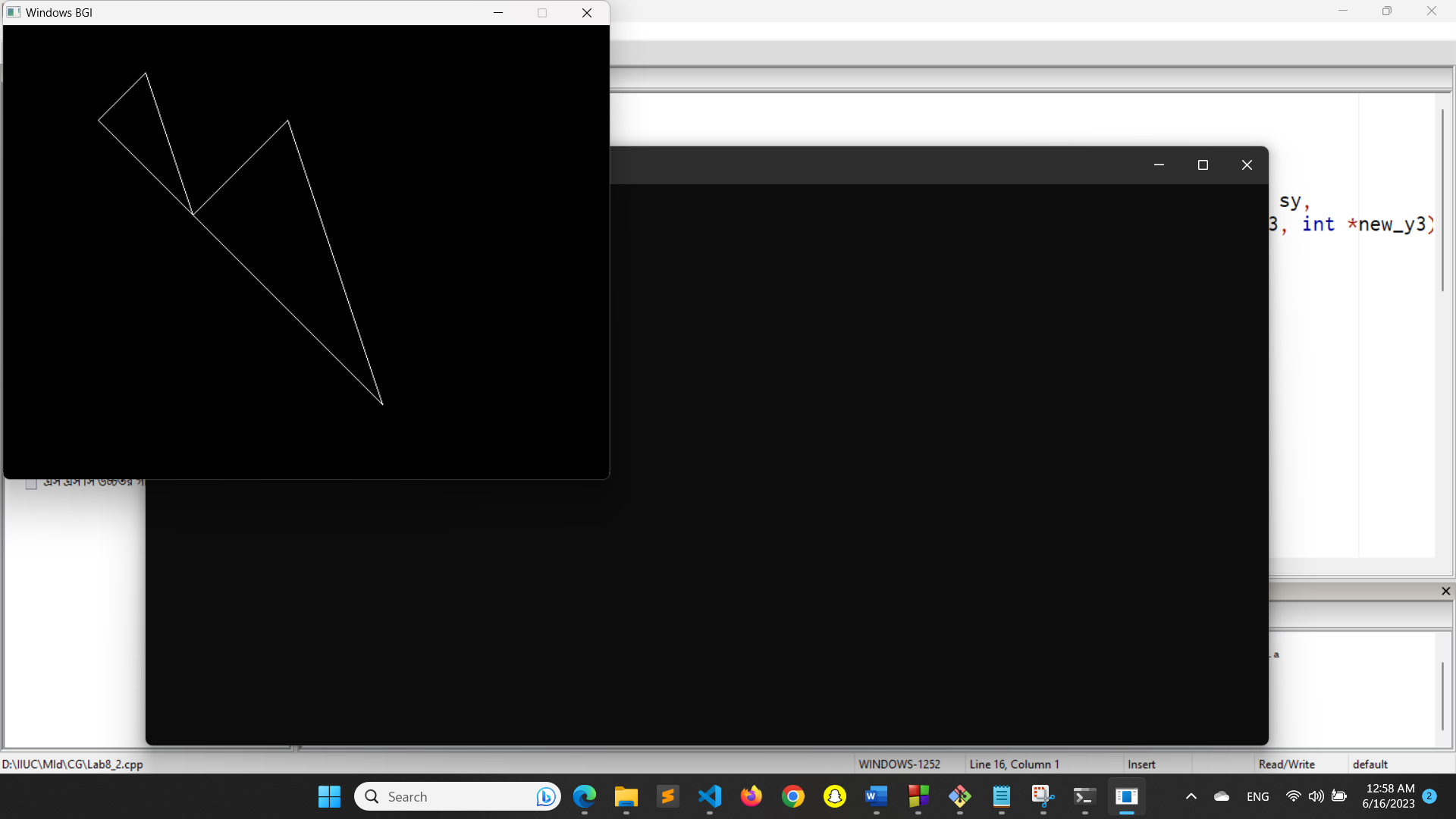
line(new\_x3, new\_y3, new\_x1, new\_y1);

getch();

closegraph();

return 0;

}



1. Scaling a triangle about another point.

Code:

#include <graphics.h>

#include <stdlib.h>

void scale\_triangle(int x1, int y1, int x2, int y2, int x3, int y3, int cx, int cy, float sx, float sy,

int \*new\_x1, int \*new\_y1, int \*new\_x2, int \*new\_y2, int \*new\_x3, int \*new\_y3)

{

// Translate points

x1 -= cx;

y1 -= cy;

x2 -= cx;

y2 -= cy;

x3 -= cx;

y3 -= cy;

// Scale points

x1 \*= sx;

y1 \*= sy;

x2 \*= sx;

y2 \*= sy;

x3 \*= sx;

y3 \*= sy;

// Translate points back to original position

x1 += cx;

y1 += cy;

x2 += cx;

y2 += cy;

x3 += cx;

y3 += cy;

\*new\_x1 = x1;

\*new\_y1 = y1;

\*new\_x2 = x2;

\*new\_y2 = y2;

\*new\_x3 = x3;

\*new\_y3 = y3;

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

// Original triangle

int x1 = 100, y1 = 100, x2 = 200, y2 = 200, x3 = 150, y3 = 50;

line(x1, y1, x2, y2);

line(x2, y2, x3, y3);

line(x3, y3, x1, y1);

// Point to scale around

int cx = 150, cy = 150;

circle(cx, cy, 3);

// Scaling factors

float sx = 2.0, sy = 3.0;

// Scale triangle

int new\_x1, new\_y1, new\_x2, new\_y2, new\_x3, new\_y3;

scale\_triangle(x1, y1, x2, y2, x3, y3, cx, cy, sx, sy, &new\_x1, &new\_y1, &new\_x2, &new\_y2, &new\_x3, &new\_y3);

// Display scaled triangle

line(new\_x1, new\_y1, new\_x2, new\_y2);

line(new\_x2, new\_y2, new\_x3, new\_y3);

line(new\_x3, new\_y3, new\_x1, new\_y1);

getch();

closegraph();

return 0;

}

