

Lab Report: 01
Title: Implementation of Algorithms for drawing 2D primitives

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

Date of Submission: 28.05.2023



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

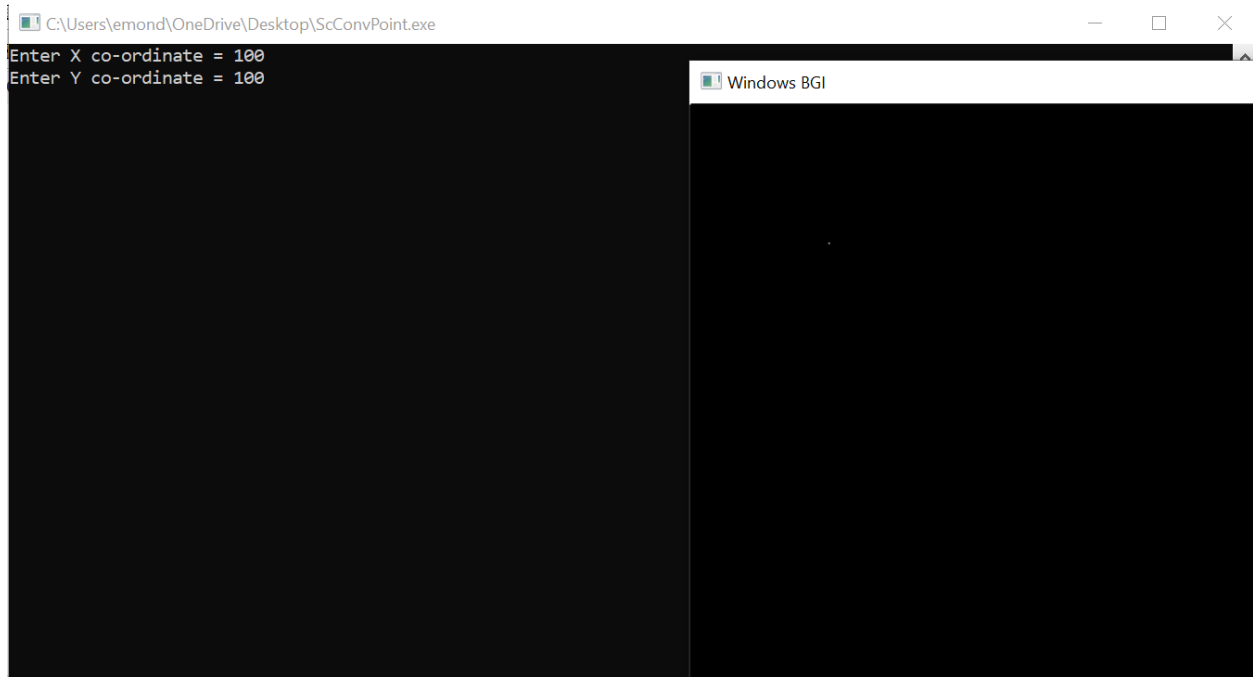
Serial No.	Algorithm Name	Page No.
1	Scan Conversion of a Point	2 - 3
2	Line Scan Conversion using DDA Algorithm	3 - 4
3	Line Scan Conversion using Bresenham's Line Algorithm	5 - 6
4	Circle Scan Conversion using Bresenham's Circle Algorithm	6 - 7

Algorithm Title: Scan Conversion of a Point

Source Code:

```
#include<bits/stdc++.h>
#include<graphics.h>
int main()
{
    float x,y;
    printf("Enter X co-ordinate = ");
    scanf("%f",&x);
    printf("Enter Y co-ordinate = ");
    scanf("%f",&y);
    int gd= DETECT, gm;
    initgraph(&gd, &gm, (char*)"");
    putpixel(floor(x),floor(y),WHITE);
    getch();
    closegraph();
}
```

GUI Output:



Algorithm Title: Line Scan Conversion using DDA Algorithm

Source Code:

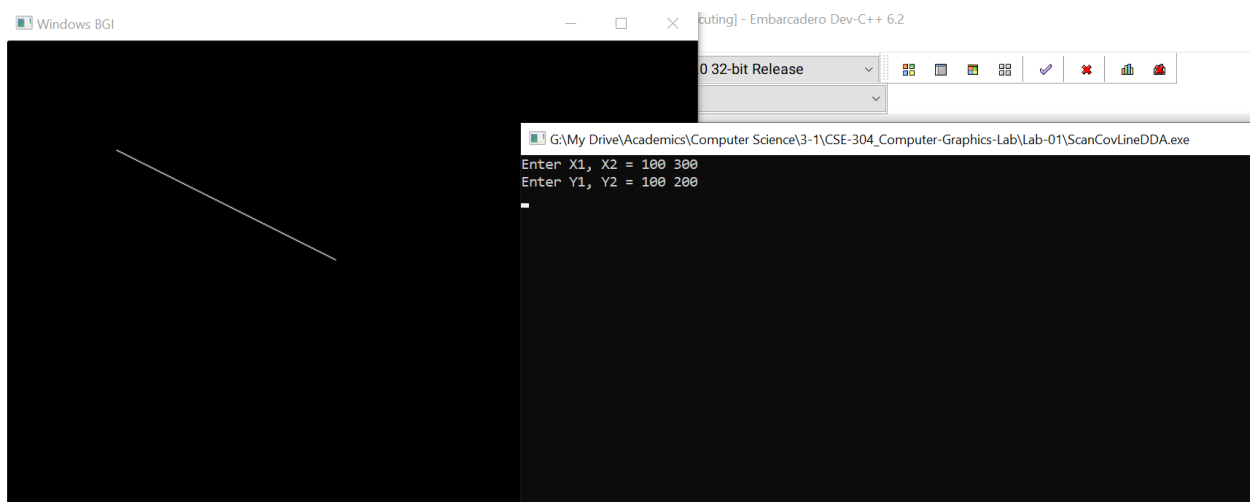
```
#include <graphics.h>
void drawLine(int x1, int y1, int x2, int y2) {
    int dx = x2 - x1;
    int dy = y2 - y1;
    int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);
    float xIncrement = static_cast<float>(dx) / steps;
    float yIncrement = static_cast<float>(dy) / steps;
    float x = x1;
    float y = y1;
    for (int i = 0; i <= steps; i++) {
        putpixel(static_cast<int>(x), static_cast<int>(y),
WHITE);
        x += xIncrement;
        y += yIncrement;
    }
}
```

```

    }
}
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, (char*)"");
    float x1, x2, y1, y2;
    printf("Enter X1, X2 = ");
    scanf("%f%f",&x1, &x2);
    printf("Enter Y1, Y2 = ");
    scanf("%f%f",&y1, &y2);
    drawLine(x1, y1, x2, y2); // Draw the line from (x1, y1) to
(x2, y2)
    delay(5000); // Delay to show the output
    closegraph();
    return 0;
}

```

GUI Output:



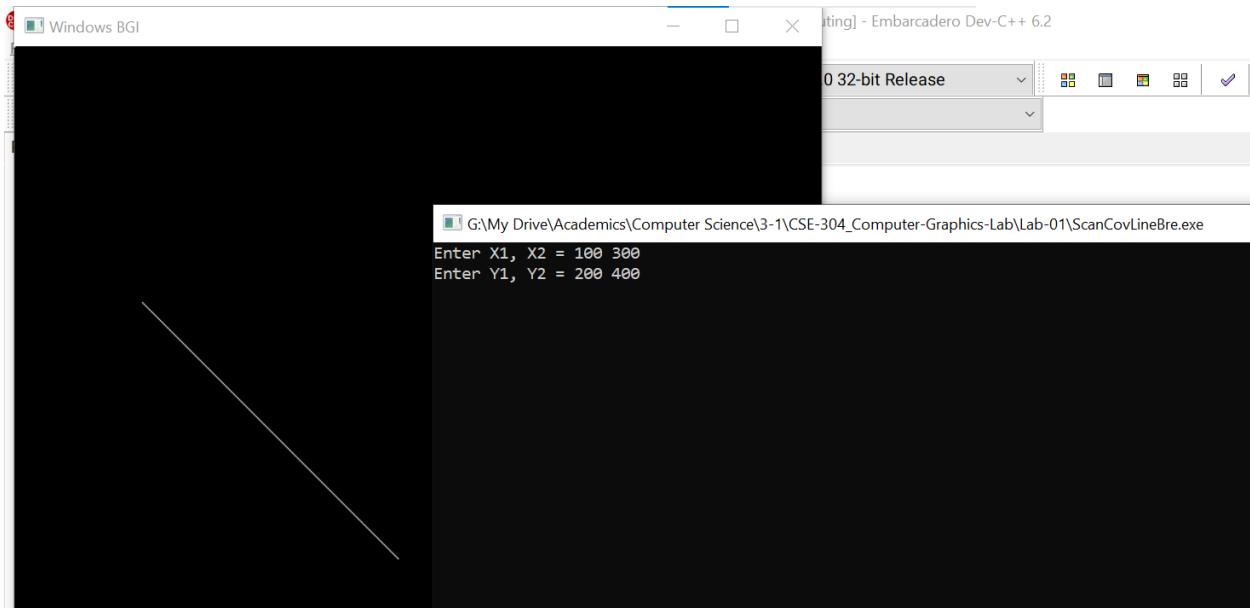
Algorithm Title: Line Scan Conversion using Bresenham's Line Algorithm

Source Code:

```
#include <graphics.h>
void drawLine(int x1, int y1, int x2, int y2) {
    int dx = abs(x2 - x1);
    int dy = abs(y2 - y1);
    int sx = (x1 < x2) ? 1 : -1;
    int sy = (y1 < y2) ? 1 : -1;
    int err = dx - dy;
    while (true) {
        putpixel(x1, y1, WHITE);
        if (x1 == x2 && y1 == y2)
            break;
        int e2 = 2 * err;
        if (e2 > -dy) {
            err -= dy;
            x1 += sx;
        }
        if (e2 < dx) {
            err += dx;
            y1 += sy;
        }
    }
}

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, (char*)"");
    int x1, x2, y1, y2;
    printf("Enter X1, X2 = ");
    scanf("%d%d", &x1, &x2);
    printf("Enter Y1, Y2 = ");
    scanf("%d%d", &y1, &y2);
    drawLine(x1, y1, x2, y2); // Draw the line from (x1, y1) to
(x2, y2)
    delay(5000); // Delay to show the output
    closegraph();
    return 0;
}
```

GUI Output:



Algorithm Title: Circle Scan Conversion using Bresenham's Circle Algorithm

Source Code:

```
#include <graphics.h>
void drawCircle(int xc, int yc, int radius) {
    int x = 0;
    int y = radius;
    int d = 3 - 2 * radius;
    while (x <= y) {
        putpixel(xc + x, yc + y, WHITE);
        putpixel(xc - x, yc + y, WHITE);
        putpixel(xc + x, yc - y, WHITE);
        putpixel(xc - x, yc - y, WHITE);
        putpixel(xc + y, yc + x, WHITE);
        putpixel(xc - y, yc + x, WHITE);
        putpixel(xc + y, yc - x, WHITE);
        putpixel(xc - y, yc - x, WHITE);

        if (d < 0)
            d += 4 * x + 6;
    }
}
```

```

        else {
            d += 4 * (x - y) + 10;
            y--;
        }

        x++;
    }
}

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int xc, yc, radius;
    printf("Enter X coordiate of the center: ");
    scanf("%d", &xc);
    printf("Enter Y co-ordinate of the center: ");
    scanf("%d", &yc);
    printf("Enter the circle radius: ");
    scanf("%d", &radius);
    drawCircle(xc, yc, radius); // Draw the circle
    delay(5000); // Delay to show the output
    closegraph();
    return 0;
}

```

GUI Output:

