

Course title: Computer Graphics Lab
Course code: CSE-304

Date of Submission: 28-05-23



Submitted to-

Dr. Mohammad Shorif Uddin

Professor

Dr. Morium Akter

Associate Professor

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

Sl	Class Roll	Exam Roll	Name
01	376	202188	Md. Shakhwat Hosen

Point Conversion:

Code:

```
#include <iostream>
using namespace std;
#include <graphics.h>

int main() {
    int x; // x-coordinate of the point
    int y ; // y-coordinate of the point
    cout<<"Enter position: ";
    cin>>x>>y;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    // Set the pixel at (x, y) to a specific color
    putpixel(x, y, WHITE);

    //delay(50000); // Pause for 5 seconds
    getch();
    closegraph();
    return 0;
}
```

Output:

 E:\393\Point.exe

```
Enter position: 200 300

Process returned 0 (0x0)   execution time : 12.310 s
Press any key to continue.
```

Windows BGI



Line Conversion:

Code:

```
#include <bits/stdc++.h>
using namespace std;
#include <graphics.h>
```

```
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
```

```
    int x1 = 100; // x-coordinate of the starting point
    int y1 = 100; // y-coordinate of the starting point
    int x2 = 300; // x-coordinate of the ending point
    int y2 = 200; // y-coordinate of the ending point
```

```
    int dx = x2 - x1;
    int dy = y2 - y1;
```

```
int steps =abs(dy);

float xIncrement = dx / (float)steps;
float yIncrement = dy / (float)steps;

float x = x1;
float y = y1;

for (int i = 0; i <= steps; i++) {
    putpixel(x, y, WHITE);
    x += xIncrement;
    y += yIncrement;
}

// delay(5000); // Pause for 5 seconds
getch();
closegraph();
return 0;
}
```

Output:

E:\393\line.exe



Line Conversion using bresenham algorithm:

Code:

```
#include<bits/stdc++.h>
using namespace std;
#include<graphics.h>

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    int x1 = 100; // x-coordinate of the starting point
    int y1 = 100; // y-coordinate of the starting point
    int x2 = 300; // x-coordinate of the ending point
    int y2 = 200; // y-coordinate of the ending point

    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);

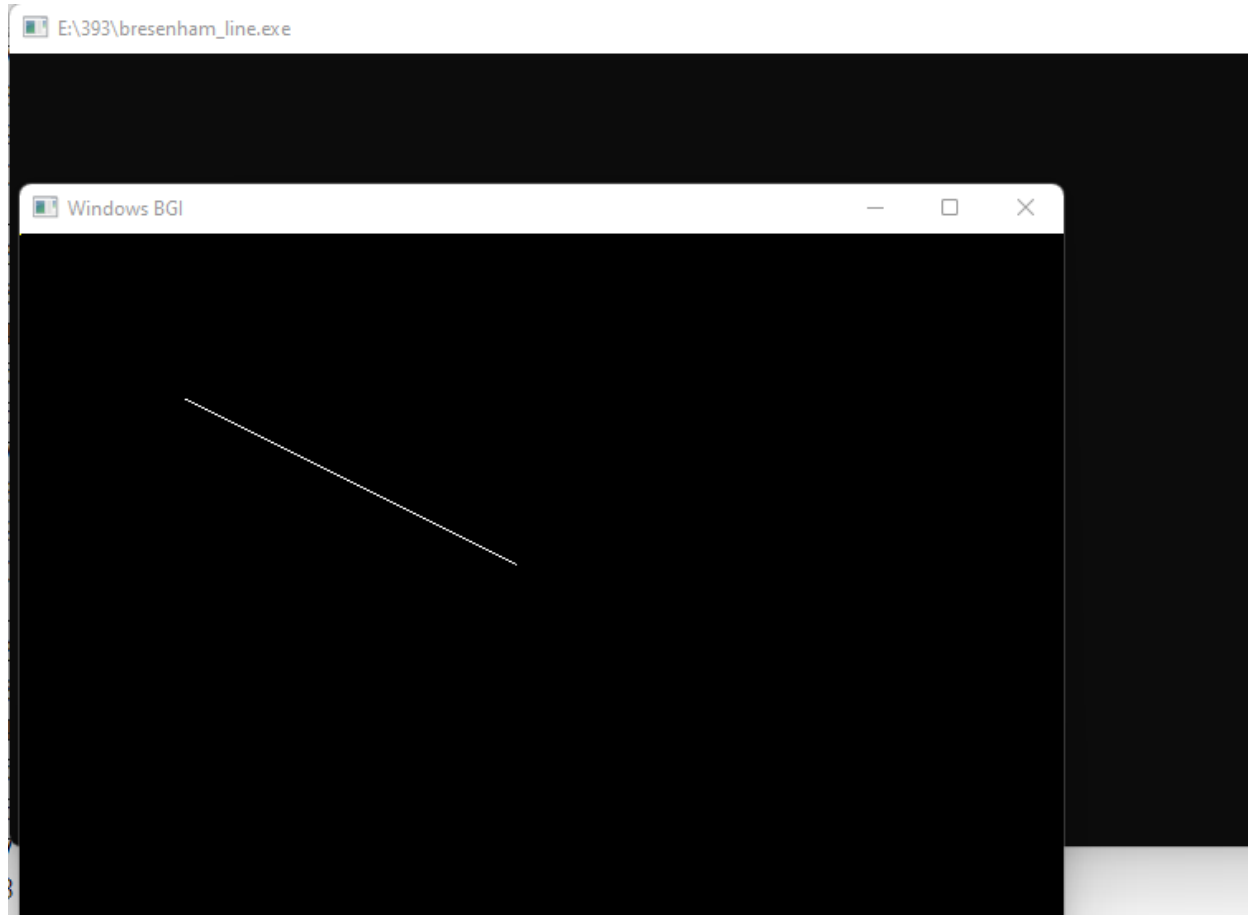
        if (x1 == x2 && y1 == y2)
            break;

        int e2 = 2 * err;
        if (e2 > -dy) {
            err -= dy;
            x1 += sx;
        }

        if (e2 < dx) {
            err += dx;
            y1 += sy;
        }
    }
    delay(5000); // Pause for 5 seconds

    closegraph();
    return 0;
}
```

Output:



Circle Conversion using bresenham algorithm:

Code:

```
#include <iostream>
using namespace std;
#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 = 200; // x-coordinate of the center
    int yc = 200; // y-coordinate of the center
    int radius = 100; // radius of the circle
    drawCircle(xc, yc, radius);
    // delay(5000); // Pause for 5 seconds
    getch();
    closegraph();
    return 0;
}

```

Output:

E:\393\bresenham_circle.exe

