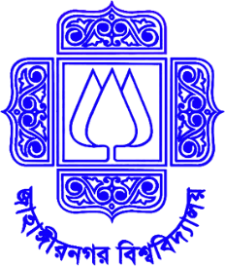
***Course Title:*** *Computer Graphics Laboratory*

***Course code:*** *CSE-304*

*3rd year 1st semester*

**Date of Submission**: 04/06/2023



###### **Submitted to-**

###### Dr. Mohammad Shorif Uddin

###### **Professor**

###### and

###### Dr. Morium Akter

###### **Associate Professor**

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

| **Sl** | Class Roll | Exam Roll | Name |
| --- | --- | --- | --- |
| 01 | 395 | 202207 | Md. Mahfuz Molla |

**Experiment No : 05**

**Name OF Experiment : Scan Conversion OF an Ellipse**

**Source Code of Ellipse:**

#include <iostream>

#include <graphics.h>

void plotEllipsePoints(int xc, int yc, int x, int 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);

}

void drawEllipse(int xc, int yc, int rx, int ry)

{

int x = 0;

int y = ry;

int rxSq = rx \* rx;

int rySq = ry \* ry;

int twoRxSq = 2 \* rxSq;

int twoRySq = 2 \* rySq;

int p;

int px = 0;

int py = twoRxSq \* y;

plotEllipsePoints(xc, yc, x, y);

p = rySq - (rxSq \* ry) + (0.25 \* rxSq);

while (px < py)

{

x++;

px += twoRySq;

if (p < 0)

{

p += rySq + px;

}

else

{

y--;

py -= twoRxSq;

p += rySq + px - py;

}

plotEllipsePoints(xc, yc, x, y);

}

p = rySq \* (x + 0.5) \* (x + 0.5) + rxSq \* (y - 1) \* (y - 1) - rxSq \* rySq;

while (y > 0)

{

y--;

py -= twoRxSq;

if (p > 0)

{

p += rxSq - py;

}

else

{

x++;

px += twoRySq;

p += rxSq - py + px;

}

plotEllipsePoints(xc, yc, x, y);

}

}

int main()

{

int gd = DETECT, gm;

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

int xc = 250; // X-coordinate of the center of the ellipse

int yc = 250; // Y-coordinate of the center of the ellipse

int rx = 150; // X-radius of the ellipse

int ry = 100; // Y-radius of the ellipse

drawEllipse(xc, yc, rx, ry);

delay(5000);

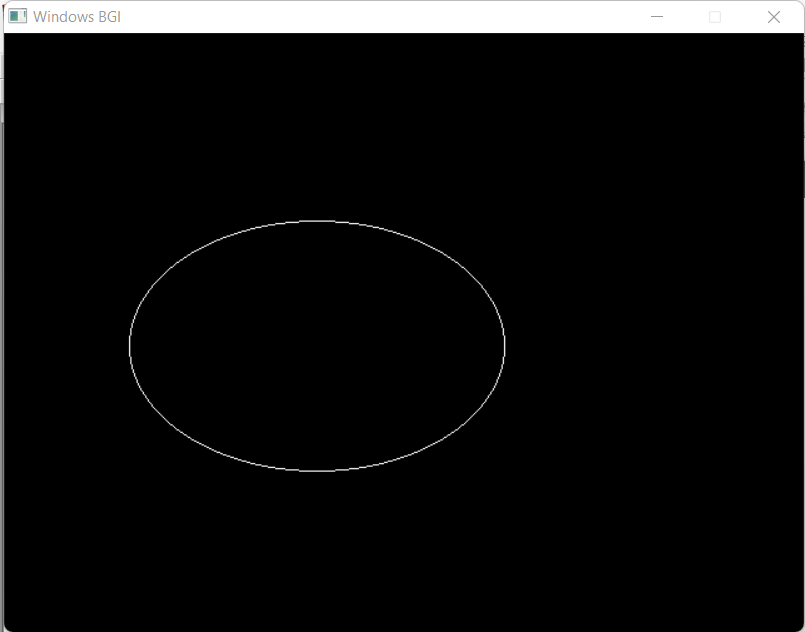
getch();

closegraph();

return 0;

}

**Output :**



**Experiment No : 06**

**Name OF Experiment : Scan Conversion a Circle usinPoint g Mid-**

**Source Code of Circle(Mid-Point):**

#include <iostream>

#include <graphics.h>

using namespace std;

void drawCircle(int xc, int yc, int radius) {

int x = 0;

int y = radius;

int p = 1 - radius;

while (x <= y) {

putpixel(xc + x, yc + y, WHITE);

putpixel(xc + y, yc + x, WHITE);

putpixel(xc - x, yc + y, WHITE);

putpixel(xc - y, yc + x, WHITE);

putpixel(xc + x, yc - y, WHITE);

putpixel(xc + y, yc - x, WHITE);

putpixel(xc - x, yc - y, WHITE);

putpixel(xc - y, yc - x, WHITE);

x++;

if (p < 0) {

p += 2 \* x + 1;

} else {

y--;

p += 2 \* (x - y) + 1;

}

}

}

int main() {

int xc, yc, radius;

cout << "Enter center coordinates (xc and yc): ";

cin >> xc >> yc;

cout << "Enter radius: ";

cin >> radius;

int gd = DETECT, gm;

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

drawCircle(xc, yc, radius);

getch();

closegraph();

return 0;

}

**Output:**

