**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**------------------------------------------------------------------------------------------------------------**

**Lab Exercise 4 : Midpoint Circle Drawing Algorithm in C++ using OpenGL**

1. **To plot points that make up the circle with center (xc,yc) and radius r using Midpoint circle drawing algorithm. Give at least 2 test cases.**

**Case 1: With center (0,0)**

**Case 2: With center (xc,yc)**

1. **To draw any object using line and circle drawing algorithms.**

**Code:**

#include <stdio.h>

#include <iostream>

#include <GL/glut.h>

using namespace std;

int pntX1, pntY1, r;

void plot(int x, int y)

{

glBegin(GL\_POINTS);

glVertex2i(x + pntX1, y + pntY1);

glEnd();

}

void myInit(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glColor3f(0.0f, 0.0f, 0.0f);

glPointSize(4.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 1000.0, 0.0, 1000.0);

}

void midPointCircleAlgo()

{

int x = 0;

int y = r;

float decision = 5 / 4 - r;

plot(x, y);

while (y > x)

{

if (decision < 0)

{

x++;

decision += 2 \* x + 1;

}

else

{

y--;

x++;

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

}

plot(x, y);

plot(x, -y);

plot(-x, y);

plot(-x, -y);

plot(y, x);

plot(-y, x);

plot(y, -x);

plot(-y, -x);

}

}

void myDisplay(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 0.5, 0.7);

glPointSize(2.0);

midPointCircleAlgo();

glFlush();

}

int main(int argc, char\*\* argv)

{

cout << "Enter the coordinates of the center: " << endl;

cout << "X-coordinate : "; cin >> pntX1;

cout << "\nY-coordinate : "; cin >> pntY1;

cout << "\nEnter radius : "; cin >> r;

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(100, 150);

glutCreateWindow("Circle");

glutDisplayFunc(myDisplay);

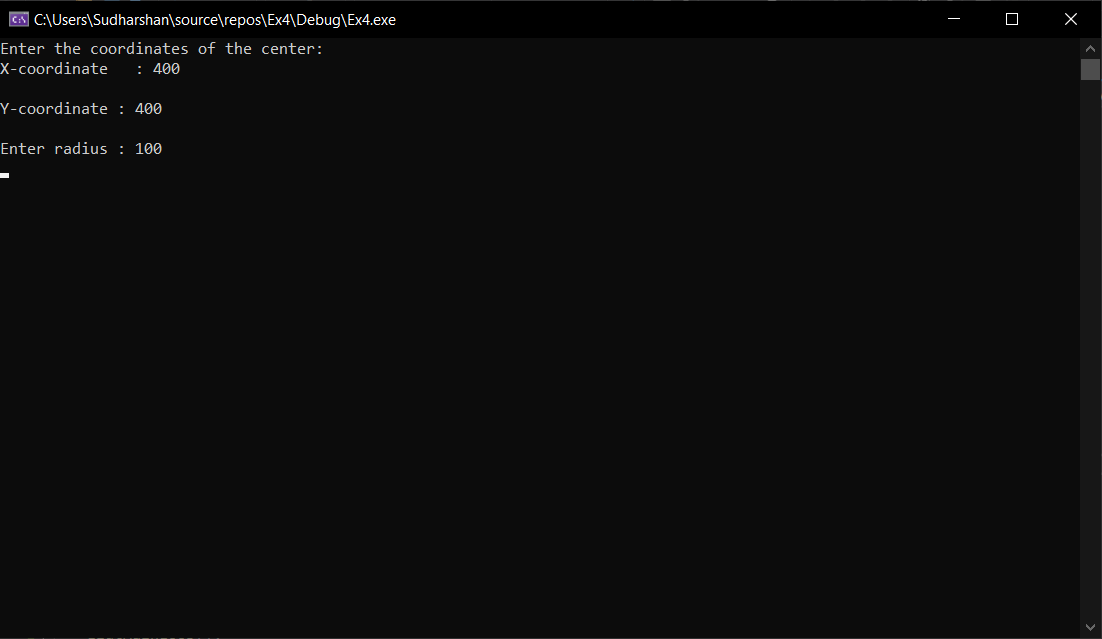
myInit();

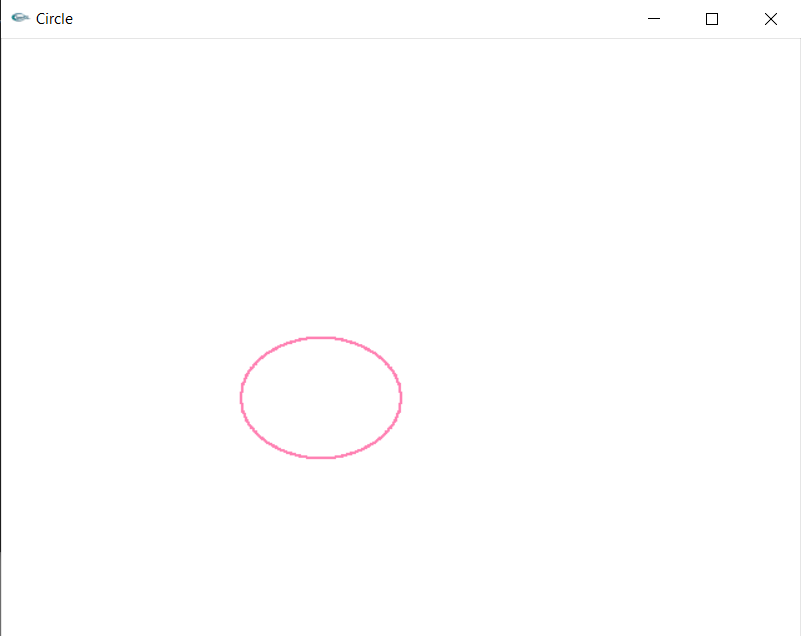
glutMainLoop();

return 0;

}

**Output:**

****

****