Name – Isha Kishor Katariya

Class - SEIT Div -A

Roll No – 207A056

Experiment -Implement Bresenham circle drawing algorithm to draw any object. The object should be displayed in all the quadrants with respect to center and radius.

#include <GL/glut.h>

#include <cmath> void init() { glClearColor(0,

0, 0, 0);

glMatrixMode(GL\_PROJECTION); gluOrtho2D(0, 500, 0, 500);

}

int centerX = 250; // Center of the circle int centerY = 250; int radius = 100; // Radius of the circle void drawCirclePoints(int x, int y) { glBegin(GL\_POINTS); glColor3f(1, 1, 0); glVertex2i(centerX + x, centerY + y); // Octant 1 glVertex2i(centerX - x, centerY + y); // Octant 2 glColor3f(1, 0, 1); glVertex2i(centerX + x, centerY - y); // Octant 8 glVertex2i(centerX - x, centerY - y); // Octant 7 glColor3f(0, 1, 1); glVertex2i(centerX + y, centerY + x); // Octant 3 glVertex2i(centerX - y, centerY + x); // Octant 4 glColor3f(1, 1, 0); glVertex2i(centerX + y, centerY - x); // Octant 6 glVertex2i(centerX - y, centerY - x); // Octant 5 glEnd(); } void drawCircle() { int x = 0; int y = radius; int decisionParam = 3

- 2 \* radius; while (x <= y) { drawCirclePoints(x, y); if (decisionParam >= 0) { decisionParam += 4 \* (x - y) + 10; y--; } else { decisionParam += 4 \* x + 6;

}x++;

} } void display()

{glClear(GL\_COLOR\_BUFFER\_BIT);

glPointSize(1.0); drawCircle(); glFlush();

}

void reshape(int width, int height) {

glViewport(0, 0, width, height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity(); gluOrtho2D(0, width, 0, height);

glMatrixMode(GL\_MODELVIEW);

} int main(int argc, char\*\* argv) { glutInit(&argc, argv); glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500); glutInitWindowPosition(450, 100); glutCreateWindow("Bresenham’s Circle Drawing Algorithm"); glutDisplayFunc(display); glutReshapeFunc(reshape); glutMainLoop(); return 0;

}