EXPERIMENT 3:

```
#include<GL/glut.h>
#include<iostream>
#include<math.h>

using namespace std;

float r, g, b, radius, S;
float x, y;
float x_1, y_1,w,z;
bool flag = true;

void mouse(int button, int state,
	int mousex, int mousey)
{
	if (button == GLUT_LEFT_BUTTON
		&& state == GLUT_DOWN) {
		flag = true;
		x_1 = (-300)+(mousex*0.53);

		y_1 = 278-mousey;
		cout << "x = " << x_1;
		cout << "y = " << y_1;
	}
}

void Circle()
{
	//x_1 = w;
```

```
//y_1 = z;
radius = 20;
S = 3 - (2 * radius);
x = 0;
y = radius;
glBegin(GL_POINTS);
glVertex2f(x + x_1, y + y_1);
glVertex2f(-x + x_1, y + y_1);
glVertex2f(x + x_1, -y + y_1);
glVertex2f(-x + x_1, -y + y_1);
glVertex2f(y + x_1, x + y_1);
glVertex2f(-y + x_1, x + y_1);
glVertex2f(y + x_1, -x + y_1);
glVertex2f(-y + x_1, -x + y_1);
glEnd();
glFlush();
while (x < y)
{
        if (S <= 0)
        {
                S = S + (4 * x) + 6;
                x = ++x;
                y = y;
                glBegin(GL_POINTS);
                glVertex2f(x + x_1, y + y_1);
                glVertex2f(-x + x_1, y + y_1);
                glVertex2f(x + x_1, -y + y_1);
                glVertex2f(-x + x_1, -y + y_1);
                glVertex2f(y + x_1, x + y_1);
                glVertex2f(-y + x_1, x + y_1);
                glVertex2f(y + x_1, -x + y_1);
```

```
                glVertex2f(-y + x_1, -x + y_1);
                glEnd();
                glFlush();
            }
            else
            {
                S = S + (4 * (x - y)) + 10;
                x = ++x;
                y = --y;
                glBegin(GL_POINTS);
                glVertex2f(x + x_1, y + y_1);
                glVertex2f(-x + x_1, y + y_1);
                glVertex2f(x + x_1, -y + y_1);
                glVertex2f(-x + x_1, -y + y_1);
                glVertex2f(y + x_1, x + y_1);
                glVertex2f(-y + x_1, x + y_1);
                glVertex2f(y + x_1, -x + y_1);
                glVertex2f(-y + x_1, -x + y_1);
                glEnd();
                glFlush();
            }
        }
}

void init(void)
{
        glClearColor(1.0, 1.0, 0.0, 0.0);
        glColor3f(1.0, 0.0, 0.0);
        gluOrtho2D(-300, 300, -300, 300);
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_LINES);
```

```
        glVertex3f(-300,0,0);
        glVertex3f(300, 0, 0);
        glVertex3f(0,-300, 0);
        glVertex3f(0,300, 0);
        glEnd();
        glFlush();
}


int main(int argc, char** argv)
{
        //cout << "Enter the center point of circle";
        //cin >> x_1 >> y_1;
        //cout << "Enter the radius of a circle";
        //cin >> radius;
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowPosition(-300,-300);
        glutInitWindowSize(-300, 300);
        glutCreateWindow("Bresenhams Circle Generation Window");
        init();
        glutMouseFunc(mouse);
        glutDisplayFunc(Circle);
        glutMainLoop();
        return 0;
}
```