**Practical 3**

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

**Program Code:-**

#include<GL/glut.h>

#include<iostream>

using namespace std;

int r;

void E\_way(int x, int y){

glBegin(GL\_POINTS);

glVertex2i(x+320,y+240);

glVertex2i(y+320,x+240);

glVertex2i(y+320, -x+240);

glVertex2i(x+320, -y+240);

glVertex2i(-x+320,-y+240);

glVertex2i(-y+320,-x+240);

glVertex2i(-y+320,x+240);

glVertex2i(-x+320,y+240);

glEnd();

glFlush();

}

void B\_circle(){

float d;

d = 3 - 2\*r;

int x,y;

x = 0 ;

y = r ;

do{

E\_way(x,y);

if(d<0){

d=d+4\*x+6;

}

else{

d= d+4\*(x-y)+10;

y=y-1;

}

x=x+1;

}while(x<y);

}

void init(){

glClearColor(1,1,1,0);

glColor3f(1,0,0);

gluOrtho2D(0,640,0,480);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

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

cout<<"\n Enter Radius \t ";

cin>>r;

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(100,100);

glutInitWindowSize(640,480);

glutCreateWindow("Circle");

init();

glutDisplayFunc(B\_circle);

glutMainLoop();

return 0;

}