**Circle using Line:**

#include <windows.h> // for MS Windows

#include <GL/glut.h> // GLUT, include glu.h and gl.h

#include<math.h>>

# define PI 3.14159265358979323846

/\* Handler for window-repaint event. Call back when the window first appears and

whenever the window needs to be re-painted. \*/

void display() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque

glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)

GLfloat x=.4f; GLfloat y=.4f; GLfloat radius =.2f;

int i;

int lineAmount = 100; //# of triangles used to draw circle

//GLfloat radius = 0.8f; //radius

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_LINE\_LOOP);

for(i = 0; i <= lineAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / lineAmount)),

y + (radius\* sin(i \* twicePi / lineAmount))

);

}

glEnd();

glFlush(); // Render now

}

/\* Main function: GLUT runs as a console application starting at main() \*/

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

glutInit(&argc, argv); // Initialize GLUT

glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title

glutInitWindowSize(320, 320); // Set the window's initial width & height

glutDisplayFunc(display); // Register display callback handler for window re-paint

glutMainLoop(); // Enter the event-processing loop

return 0;

}

**Circle using GL\_TRIANGLE\_FAN**

#include <windows.h> // for MS Windows

#include <GL/glut.h> // GLUT, include glu.h and gl.h

#include<math.h>>

# define PI 3.14159265358979323846

/\* Handler for window-repaint event. Call back when the window first appears and

whenever the window needs to be re-painted. \*/

void display() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque

glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)

int i;

GLfloat x=.4f; GLfloat y=.4f; GLfloat radius =.2f;

int triangleAmount = 20; //# of triangles used to draw circle

//GLfloat radius = 0.8f; //radius

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(x, y); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glFlush(); // Render now

}

/\* Main function: GLUT runs as a console application starting at main() \*/

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

glutInit(&argc, argv); // Initialize GLUT

glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title

glutInitWindowSize(320, 320); // Set the window's initial width & height

glutDisplayFunc(display); // Register display callback handler for window re-paint

glutMainLoop(); // Enter the event-processing loop

return 0;

}