# Computer Graphics-Assignment 1

Shape1(display function):

void displaysquare(void){ //Display function

glClearColor(0.0, 1.0, 0.0, 0.0); // establishes what color the window will be cleared to (green)

glClear(GL\_COLOR\_BUFFER\_BIT); // It specifies which color buffer to be cleared

glColor4f(1.0, 1.0, 1.0, 1.0); // It makes square color white

glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0); //specifies the *coordinate system* OpenGL assumes as it draws the final

image and how the image is mapped to the screen.

glBegin(GL\_POLYGON);

glVertex3f(0.25, 0.25, 0.0); //it sets a vertex for use to make a geometric object.

glVertex3f(0.75, 0.25, 0.0);

glVertex3f(0.75, 0.75, 0.0);

glVertex3f(0.25, 0.75, 0.0);

glEnd(); //It marks the end of a vertex-data list.

glFlush(); } //ensures that the drawing commands are actually executed , rather than

stored in a *buffer* awaiting additional OpenGL commands.

Shape1(main):

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

glutInit(&argc, argv); //Initializes the interaction between openGL and windowing system.

glutInitDisplayMode (GLUT\_SINGLE); //Initializes the interaction between openGL and windowing system.

glutInitWindowSize (300, 300); //Specifies the size, in pixels, of your window

glutInitWindowPosition (400, 100); // Specifies the screen location

glutCreateWindow ("solid-white-square"); //create a window and put a title to the window

glutDisplayFunc(displaysquare); //put the display callback for the current window.

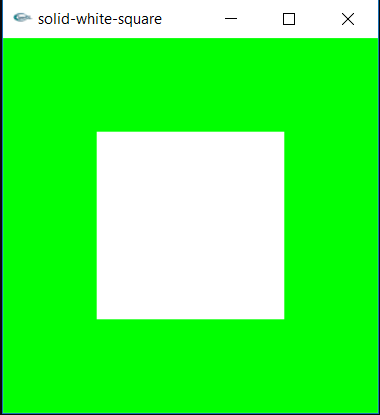
glutMainLoop(); //Event processing begins, and the registered display callback is

triggered and All windows that have been created are now shown

return 0;

}

**The output of shape1**

****

Shape2(display function):

void displaylines(void) { //Display function

glColor4f(1, 0, 0, 0); // It makes line color red

glClear(GL\_COLOR\_BUFFER\_BIT); //It specifies which color buffer to be cleared

glBegin(GL\_LINE\_LOOP); //it marks the beginning of a vertex-data list that describes a geometric primitive.

glVertex2f(5.0, 10.0); //it sets a vertex for use to make a geometric object.

glVertex2f(10.0, 10.0);

glLineWidth(3.0); //It sets the line width

glEnd(); //It marks the end of a vertex-data list.

glFlush();} //ensures that the drawing commands are actually executed , rather than stored in a *buffer*

awaiting additional OpenGL commands.

Shape2(main):

int main(int argc, char\*\* argv) { // Main function

glutInit(&argc, argv); //Initializes the interaction between openGL and windowing system.

glutInitDisplayMode(GLUT\_SINGLE); // put the frame buffer and coloring modes and here we use single buffer

glutInitWindowPosition(200, 200); // Specifies the screen location

glutInitWindowSize(500, 500); //Specifies the size, in pixels, of your window

glutCreateWindow("Square-using-lines"); //create a window and put a title to the window

glClearColor(0.0, 0.0, 0.0, 0.0); //establishes what color the window will be cleared to

glutDisplayFunc(displaylines); //put the display callback for the current window.

glOrtho(0.0, 15.0, 0.0, 15.0, -5, 1.0); //specifies the *coordinate system* OpenGL assumes as it draws the final

image and how the image is mapped to the screen.

glutMainLoop(); } //Event processing begins, and the registered display callback is

triggered and All windows that have been created are now shown

**The output of shape 2**

