

```
#include<windows.h>
#include<GL/glut.h>
#define QSTRIP 1
//function to define display list
void display()
{
    glClearColor(0.0,0.0,0.0,0.0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glNewList(QSTRIP, GL_COMPILE);//display list
        glBegin(GL_QUAD_STRIP);
            glColor3f(0.0,0.0,1.0);
            glVertex3f(0.0,0.0,0.0);//v0
            glVertex3f(2.0,0.0,0.0);//v1
            glColor3f(0.0,1.0,0.0);
            glVertex3f(0.0,2.0,0.0);//v2
            glVertex3f(2.0,2.0,0.0);//v3
            glColor3f(1.0,0.0,0.0);
            glVertex3f(0.0,4.0,0.0);//v4
            glVertex3f(2.0,4.0,0.0);//v5
        glEnd();
    glEndList();
}
//functions which calls display list
void quad()
{
    glCallList(QSTRIP);
    glFlush();
}
//reshape event handling function
void reshape(GLint w, GLint h)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-5,5,-5,5);
    glMatrixMode(GL_MODELVIEW);
    glViewport(0,0,w,h);
}
//mouse event handling function
void myMouse(int button, int state, int x, int y)
{
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
        quad();
    if(button==GLUT_RIGHT_BUTTON && state==GLUT_DOWN)
        exit(0);
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH|GLUT_RGB|GLUT_SINGLE);
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    glutCreateWindow("display list");
    glutDisplayFunc(display);
    glutMouseFunc(myMouse);
    glutReshapeFunc(reshape);
    glutMainLoop();
}
```

/* program to draw a small square on the screen when the left button is Clicked and window is closed when 'q' key is pressed */

```
#include<windows.h>
#include<GL/glut.h>
#define QSTRIP 1
GLfloat size=50.0;
GLsizei wh=300, ww=300;
void display()
{
    glClearColor(0.0,0.0,0.0,0.0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
}
void drawsquare(int x, int y)
{
    glBegin(GL_POLYGON);
        glColor3f(0.0,0.0,1.0);
        glVertex2f(x+size, y+size);
        glVertex2f(x-size, y+size);
        glVertex2f(x-size, y-size);
        glVertex2f(x+size, y-size);
    glEnd();
    glFlush();
}
void reshape(GLint w, GLint h)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,(GLdouble)w,0.0,(GLdouble)h);
    glMatrixMode(GL_MODELVIEW);
    glViewport(0,0,w,h);
    ww=w;
    wh=h;
}
void myMouse(int button, int state, int x, int y)
{
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
        drawsquare(x,y);
}
void myKey(unsigned char key, int x, int y)
{
    if(key=='q' || key=='Q')
        exit(0);
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH|GLUT_RGB|GLUT_SINGLE);
    glutInitWindowSize(ww,wh);
    glutInitWindowPosition(100,100);
    glutCreateWindow("display list");
    glutDisplayFunc(display);
    glutMouseFunc(myMouse);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(myKey);
    glutMainLoop();
}
```

```
#include<windows.h>
#include<GL/glut.h>
GLint size=2;
void display()
{
    glClearColor(0.0,0.0,0.0,1.0);
    glClear(GL_COLOR_BUFFER_BIT);
}
void reshape(GLint w, GLint h)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-10,10,-10,10);
    glMatrixMode(GL_MODELVIEW);
    glViewport(0,0,w,h);
}
void draw(int size)
{
    glBegin(GL_POLYGON);
        glVertex2f(-size, -size);
        glVertex2f(size, -size);
        glVertex2f(size, size);
        glVertex2f(-size, size);
    glEnd();
    glFlush();
}
void demo_menu(int id)
{
    switch(id)
    {
        case 1: exit(0);
                break;
        case 3: size=2*size;
                glColor3f(0.0,0.0,1.0);
                draw(size);
                break;
        case 4: if(size>3)
                size=size/2;
                glColor3f(0.0,1.0,0.0);
                draw(size);
                break;
        case 2: glColor3f(1.0,0.0,1.0);
                draw(size);
                break;
    }
    glutPostRedisplay();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH|GLUT_RGB|GLUT_SINGLE);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(500,500);
    glutCreateWindow("sample menu");
    glutCreateMenu(demo_menu);
    glutAddMenuEntry("Quit", 1);
    glutAddMenuEntry("original",2);
    glutAddMenuEntry("Increase",3);
    glutAddMenuEntry("Decrease",4);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
    glutDisplayFunc(display);
```

```
}  
  
/* Program on Menu and sub menu */  
  
#include<windows.h>  
#include<GL/glut.h>  
GLint size=2;  
int sub_menu;  
void display()  
{  
    glClearColor(0.0,0.0,0.0,1.0);  
    glClear(GL_COLOR_BUFFER_BIT);  
}  
void reshape(GLint w, GLint h)  
{  
    glMatrixMode(GL_PROJECTION);  
    glLoadIdentity();  
    gluOrtho2D(-10,10,-10,10);  
    glMatrixMode(GL_MODELVIEW);  
    glViewport(0,0,w,h);  
}  
void draw(int size)  
{  
    glBegin(GL_POLYGON);  
        glVertex2f(-size, -size);  
        glVertex2f(size, -size);  
        glVertex2f(size, size);  
        glVertex2f(-size, size);  
    glEnd();  
    glFlush();  
}  
void demo_menu(int id)  
{  
    switch(id)  
    {  
        case 1: exit(0);  
                break;  
        case 2: glColor3f(1.0,0.0,1.0);  
                draw(size);  
                break;  
        case 3: size=2*size;  
                glColor3f(0.0,0.0,1.0);  
                draw(size);  
                break;  
        case 4: if(size>3)  
                {  
                    size=size/2;  
                    glColor3f(0.0,1.0,0.0);  
                    draw(size);  
                    break;  
                }  
    }  
    glutPostRedisplay();  
}  
  
int main(int argc, char **argv)  
{  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_DEPTH|GLUT_RGB|GLUT_SINGLE);  
    glutInitWindowPosition(50,50);  
    glutInitWindowSize(500,500);  
    glutCreateWindow("sample menu");  
    sub_menu=glutCreateMenu(demo_menu);  
    glutAddMenuEntry("Increase",3);
```

```
glutAddMenuEntry("Decrease",4);
glutCreateMenu(demo_menu);
glutAddMenuEntry("Quit", 1);
glutAddMenuEntry("original",2);
glutAddSubMenu("Resize", sub_menu);
glutAttachMenu(GLUT_RIGHT_BUTTON);
glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutMainLoop();}

/*Program for displaying stroke and raster text*/

#include<windows.h>
#include<GL/glut.h>
void displayText( int r, int g, int b, const char *string ) {
    int j = strlen( string );
    glColor3f( r, g, b );
    glRasterPos2f(100.0, 10.0 );
    for( int i = 0; i < j; i++ ) {
        //glutStrokeCharacter( GLUT_STROKE_MONO_ROMAN, string[i] );
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, string[i]);
    }
    glFlush();
}
void display()
{
    glClearColor(0.0,0.0,0.0,1.0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    displayText(1.0,0.0,0.0,"M. Praveen Kumar, Assistant Professor, MECS");
}
void reshape(GLsizei w, GLsizei h)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,(GLdouble)w, 0.0,(GLdouble)h);
    glMatrixMode(GL_MATRIX_MODE);
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH|GLUT_RGB|GLUT_SINGLE);
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Text Display");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
}
```