Computer Graphics

Lab-2: Basic Primitives in OpenGL

```
glColor3f(0.0, 1.0, 0.0);
#include <GL/glut.h>
                                                    gIVertex2i(300, 200);
void myInit(void) {
glClearColor(1.0, 1.0,1.0, 1.0); // white backgrd
                                                     glColor3f(1.0, 1.0, 0.0);
elMatrixMode(GL PROJECTION);
                                                    glVertex2i(300, 50);
glLoadIdentity();
                                                     glColor3f(1.0, 0.0, 0.0);
gluOrtho2D(0.0, 640.0, 0.0, 480.0);
                                                     glVertex2i(500, 50);
                                                     glEnd();
                                                    //Circle Drawing
void display(void)
                                                     GLint x=0,y,p,r = 100,xc=500,yc=350;
glClear(GL COLOR BUFFER BIT);
                                                    y = r_i
                                                     p = 3-2*r;
GLint p1[] = \{200, 100\};
                                                     glBegin(GL_POINTS);
GLint p2[] = {50, 0};
                                                     glColor3f(1.0,0,1.0);
GLint p3[] = \{100, 200\};
                                                     while(x<=y)
GLint p4[] = {150, 0};
                                                      ( if(p<0)
GLint p5[] = \{0, 100\};
                                                         p = p+(4*x+6);
glBegin(GL_LINES);
                                                         else
glColor3f(1.0, 0, 0); //red
                                                         \{y = y-1;
 glVertex2i(200, 350);
                                                           p = p+4*(x-y)+10;
 glVertex2i(50, 250);
 glVertex2i(100, 450);
                                                     glVertex2i(xc+x, yc+y);
 glVertex2i(150, 250);
                                                     g/Vertex2i(xc-x, yc+y);
 glVertex2i(0, 350);
                                                     gIVertex2i(xc+x, yc-y);
glEnd():
                                                     gIVertex2i(xc-x, yc-y);
glBegin(GL LINE STRIP);
                                                     gIVertex2i(xc+y, yc+x);
glColor3f(0, 1.0, 0); //green
                                                     gIVertex2i(xc-y, yc+x);
glVertex2i(450, 250);
                                                     glVertex2i(xc+y, yc-x);
 glVertex2i(300, 150);
                                                     glVertex2i(xc-y, yc-x);
 glVertex2i(350, 350):
                                                     x = x+1;
 glVertex2i(400, 150);
 glVertex2i(250, 250);
                                                     glEnd();
glEnd():
                                                     glFlush(); // send all output to the display
glLineWidth(4.0);
glBegin(GL_LINE_LOOP);
                                                     int main(int argo, char *argv[])
glColor3f(0, 0, 1.0); //blue
                                                        glutlnit(&argc, argv);
 glVertex2iv(p1);
                                                         glutInitWindowSize(640, 480);
 glVertex2iv(p2);
                                                         glutInitWindowPosition(10, 10);
 glVertex2iv(p3);
                                                         glutCreateWindow("Basic primitives");
 glVertex2iv(p4);
                                                         glutDisplayFuno(display):
 glVertex2iv(p5);
                                                         mylnit();
glEnd();
                                                         glutMainLoop();
glLineWidth(3.0);
glBegin(GL LINE LOOP);
```

Computer Graphics

Lab-3: Primitives in OpenGL

```
#include <GL/glut.h>
                                                     glVertex2i(xc-y, yc-x);
#include<math.h>
#define PI 3.1415926535897932384626433832795
                                                     x = x+1;
  void myInit(void) {
  glClearColor(1.0, 1.0, 1.0, 1.0); // white backgrd
                                                      glEnd();
  glMatrixMode(GL_PROJECTION);
                                                      //Ellipse drawing
  glLoadIdentity();
  gluOrtho2D(0.0, 640.0, 0.0, 480.0);
                                                      xc = 500, yc = 250;
                                                      GLfloat t, d;
  void display(void)
                                                      glColor3f(1.0,0,1.0);
                                                      for(i = 0; i<360; i++)
  glClear(GL_COLOR_BUFFER_BIT);
   //Circle Drawing
   GLint x = 0, y, p, r = 100, xc = 150, yc = 150;
                                                      t = PI/180;
                                                      d = i*t;
   y = r.
   p=3-2*r,
   glBegin(GL_POINTS);
                                                       glVertex2i(x,y);
   glColor3f(0,1.0,0);
   while(x<=y)
                                                       glEnd();
   ( if(p<0)
                                                       glFlush();
   p = p+(4*x+6);
   else
    {y = y-1;}
     p = p+4*(x-y)+10;
    gIVertex2i(xc+x, yc+y);
    glVertex2i(xe-x, ye+y);
    gIVertex2i(xc+x, yc-y);
                                                          myInit();
    glVertex2i(xc-x, yc-y);
                                                          glutMainLoop();
    glVertex2i(xc+y, yc+x);
    gIVertex2i(xc-y, yc+x);
```

```
glVertex2i(xc+y, yc-x);
GLint i, r1 = 75, r2 = 150;
glBegin(GL_POINTS);
 x = xc + ceil(r1 * cos(d));
 y = yc+ceil(r2*sin(d));
 int main(int arge, char *argv[])
 { glutlnit(&argc, argv);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(10, 10);
    glutCreateWindow("Primitives in OpenGL");
    glutDisplayFunc(display);
```

OpenGL fill area routines gIVertex2i(300,25); #include<Gl/glut.h> gIVertex2i(200,75); void mylnit(void) { glEnd(); glClearColor(1.0,1.0,1.0,1.0); glClcar(GL_COLOR_BUFFER_BIT); // Clear Screen glMatrixMode(GL_PROJECTION); glBegin(GL_QUADS); glLoadIdentity(); glColor3f(0.0,1.0,1.0); gluOrtho2D(0.0, 640.0, 0.0, 480.0); glVertex2i(350,400); gIVertex2i(250,450); void display(void) gIVertex2i(150,400); glVertex2i(300,300); glClear(GL COLOR BUFFER BIT); glVertex2i(200,250); glColor3f(0,0,1.0); glVertex2i(150,100); glRecti(50, 450, 150, 275); glVertex2i(50,150); glBegin(GL_POLYGON); gIVertex2i(100,300); glColor3f(1.0,0,1.0); glEnd(); glVertex2i(200,150); glBegin(GL_QUAD_STRIP); glVertex2i(150,50); glColor3f(1.0,1.0,0.0); gIVertex2i(50,50); glVertex2i(450,250); glVertex2i(0,150); gIVertex2i(350,350); glVertex2i(100,250); glVertex2i(250,250); glEnd(); glVertex2i(400,150); glBegin(GL TRIANGLES); glVertex2i(300,150); glColor3f(0,1.0,1.0); glVertex2i(200,200); glVertex2i(200,350); glEnd(); gIVertex2i(300,450); //Character primitives in openGL glVertex2i(400,350); glColor3f(1.0,0,1.0); glVertex2i(250,300); glRasterPos2i(400, 50); gIVertex2i(350,300); glutBitmapCharacter(GLUT_BITMAP_9_BY_15, 'a'); glVertex2i(300,200); glRasterPos2i(400, 30); glEnd(); glutBitmapCharacter(GLUT_BITMAP_HELVETICA_10,'a'); glBegin(GL_TRIANGLE_STRIP); glRasterPos2i(300, 100); glColor3f(1.0,1.0,0); glutStrokeCharacter(GLUT_STROKE_MONO_ROMAN, 'a'); glVertex2i(600,350); glFlush(); glVertex2i(500,450); gIVertex2i(400,350); int main(int arge, char** argv) gIVertex2i(550,250); glutInit(&arge, argv); gIVertex2i(500,150); glutInitWindowSize(640, 480); gIVertex2i(350,250); glutInitWindowPosition(50, 50); glEnd(); glutCreateWindow("OpenGL fill area Primitives"); glBegin(GL_TRIANGLE_FAN); glutDisplayFunc(display); glColor3f(0,1.0,0); mylnit(); gIVertex2i(450,125); glutMainLoop(); glVertex2i(350,225); return 0; gIVertex2i(250,125); glVertex2i(400,25);