LAB MANUAL

for

Computer Graphics (PCS 702)

B.Tech CSE

GRAPHIC ERA HILL UNIVERSITY

(Society Area, Clement Town, Dehradun)

www.gehu.ac.in

Department of Computer Science & Engineering

OBJECTIVES OF COMPUTER GRAPHICS LAB

The main objective of this lab is to introduce to the students, the notions and aspects of developing graphics for computers using both primitive and derived. A student will be acquainted with the basic algorithms and their implementations in a programming language to create basic primitives like line triangles, line segments, polygons etc. They will be exposed to different kinds of transformations in graphics. This lab course offers a comprehensive coverage of Computer Graphics which is in sync with the industry practices and consumer demand regarding graphical software products.

In this lab students are going to learn about

- Creating various **computer graphics primitives** like lines, polygons, etc.
- Applying various **transformations** in computer graphics like translation, rotation, etc.
- Creating various **2d and 3d graphics**in opengl library.
- Creating **bezier curves**of various degrees with animating line segment.
- Creating and using various **opengl functions**to rotate 3d models.
- Creating **color buffer** and enabling **Z depth test** in opengl.
- Creating models in Computer graphics which can be transformed using**keys** and mouse

PREREQUISITES FOR THIS LAB:

Students must have a good knowledge about JavaScript and C++ programming languages and how to use a build system, IDE and UNIX operating system. She must be familiar with the concepts of application development and should be able to write code for simple logic based problems and handle operational aspects of the underlying operating system **.**

Tools Required :

1. Unix Os
2. Opengl library 3.3+
3. GCC compiler tools for c++
4. IDE for c++.

List of Programs

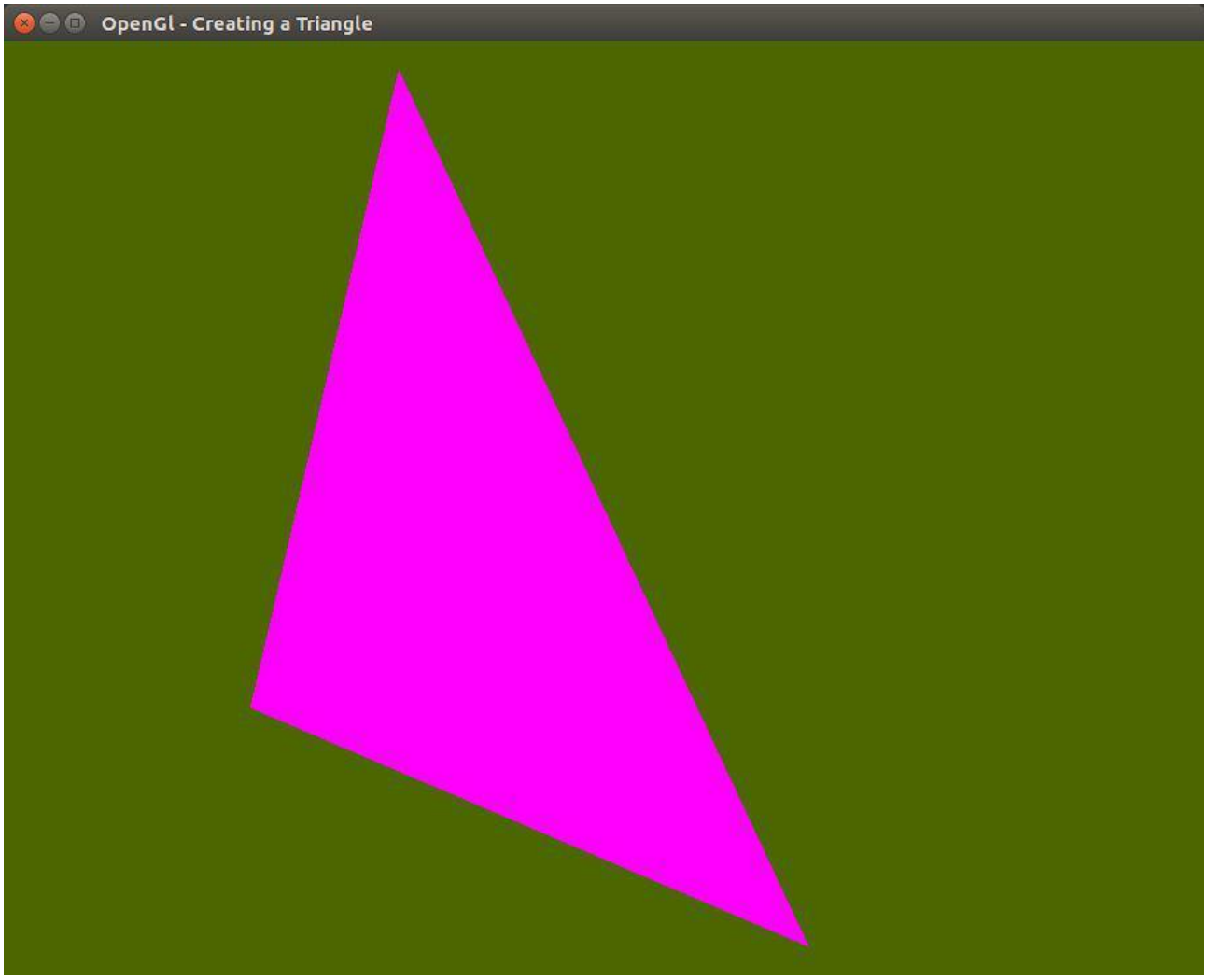
S.No.	Practical Name
1.	Write a Code in opengl to draw a triangle with a given color.
2.	Write a Code in opengl to create a polygon with a given color.
3.	Write a Code in opengl to create a line loop with a given color.
4.	Write a Code in opengl to create a line strip with a given color.
5.	Write a Code in opengl to create a triangle fan with a given color.
6.	Write a Code in opengl to create a triangle strip with a given color.
7.	Write a Code in opengl to create a self rotating triangle with a given color.
8.	Write a Code in opengl to create a triangle with a given color and rotate it with keys.
9.	Write a Code in opengl to create a self rotating polygons with a given color.
10.	Write a Code in opengl to create a self rotating cube with its each face having different color.
11.	Write a Code in opengl to draw a Bezier curve with a given color.
12.	Write a Code in opengl to create a sine wave with a given color.

Objective- Write a Code in opengl to draw a triangle with a given color.

1. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
void drawTriangle(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
 glBegin(GL_TRIANGLES);
   glVertex3f(0,1.0,0);
   glVertex3f(0,-1,0);
   glVertex3f(0.7,0.2,0);
   glEnd();
  glFlush();
}


int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a Triangle");
 glutDisplayFunc(drawTriangle);
 glutMainLoop();
 return 0;
}
```

OUTPUT-

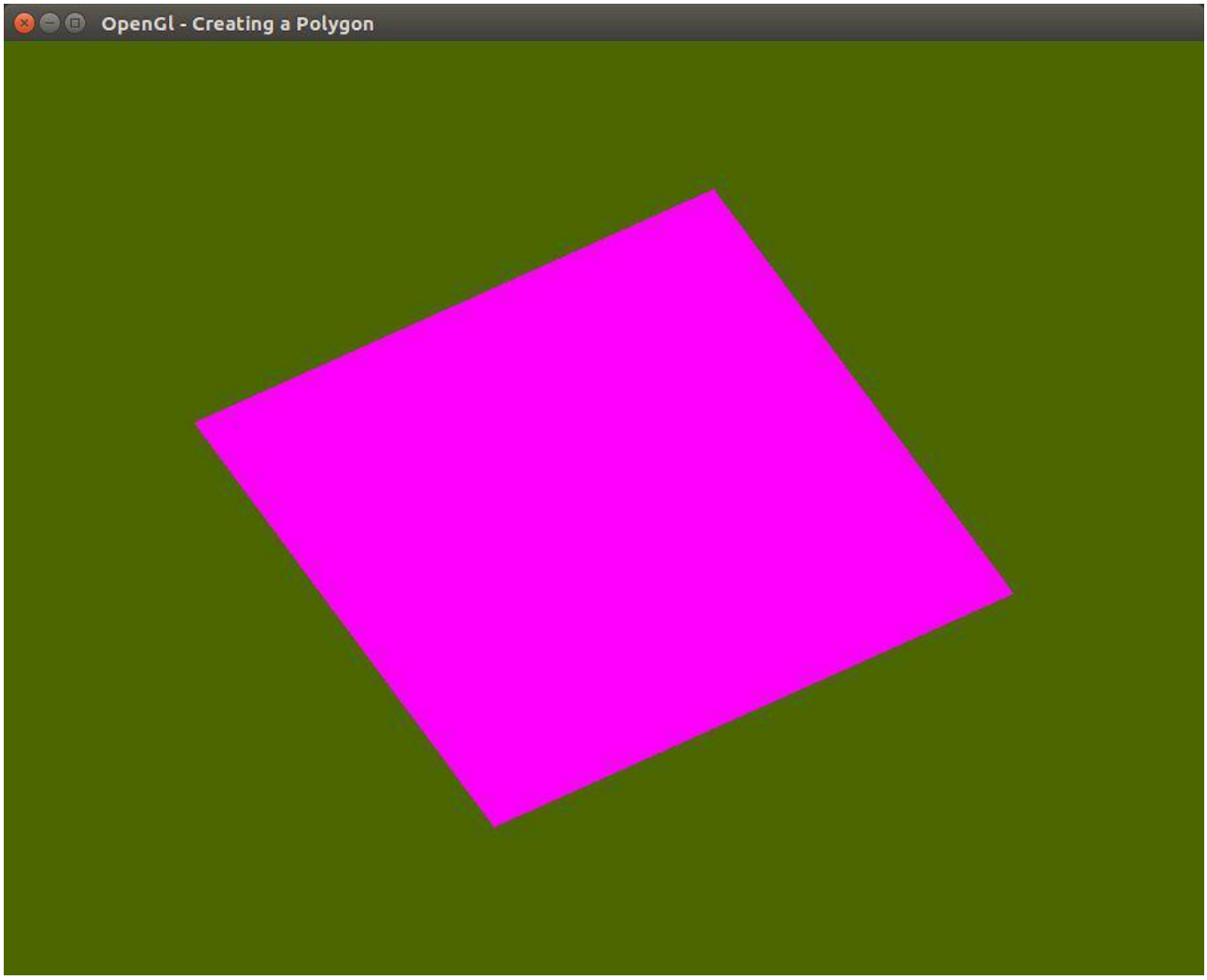


Objective- Write a Code in opengl to create a polygon with a given color.

2. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
void drawShape(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
 glBegin(GL_POLYGON);
   glVertex3f(-0.5,-0.5,0);
   glVertex3f(0.5,-0.5,0);
   glVertex3f(0.5,0.5,0);
   glVertex3f(-0.5,0.5,0);
   glEnd();
 glFlush();
}


int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a Polygon");
 glutDisplayFunc(drawShape);
 glutMainLoop();
 return 0;
}
```

OUTPUT-



Objective- Write a Code in opengl to create a line loop with a given color 3. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawShape(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glRotatef(15,0.0,0.0,1.0);
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
  glBegin(GL_LINE_LOOP);
  glVertex3f(0,0,0);
  glVertex3f(0,1,0);
  glVertex3f(1,1,0);
  glVertex3f(0.5,0,0);
  glVertex3f(0,0,0);
  glEnd();
 glFlush();
}


int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a LineLoop");
 glutDisplayFunc(drawShape);
```

```
 glutMainLoop();
 return 0;
}
```

OUTPUT-

Objective-Write a Code in opengl to create a line strip with a given color.

4. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawShape(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glRotatef(15,0.0,0.0,1.0);
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
  glBegin(GL_LINE_STRIP);
   glVertex3f(0,1.0,0);
   glVertex3f(0.7,0.7,0);
   glVertex3f(0,-1,0);
   glVertex3f(-0.7,-0.7,0);
   glEnd();
  glFlush();
}
int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a LineStrip");
 glutDisplayFunc(drawShape);
 glutMainLoop();
 return 0;
}
```

OUTPUT-

Objective-Write a Code in opengl to create a triangle fan with a given color..

5. Code

```
 #include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawTriangle(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
glBegin(GL_TRIANGLE_FAN);
    glVertex2f(0.0,0.0);
    glVertex2f(0.0,0.3);
    glVertex2f(0.2,0.3);
    glVertex2f(0.3,0.2);
    glVertex2f(0.35,0.05);
    glVertex2f(0.3,-0.1);
    glVertex2f(0.2,-0.2);
    glVertex2f(0.0,-0.25);
    glVertex2f(-0.2,-0.2);
    glVertex2f(-0.3,-0.1);
   glEnd();
  glFlush();
}

int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
```

```
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a Triangle
 Fan"); glutDisplayFunc(drawTriangle); glutMainLoop();

 return 0;
}
```

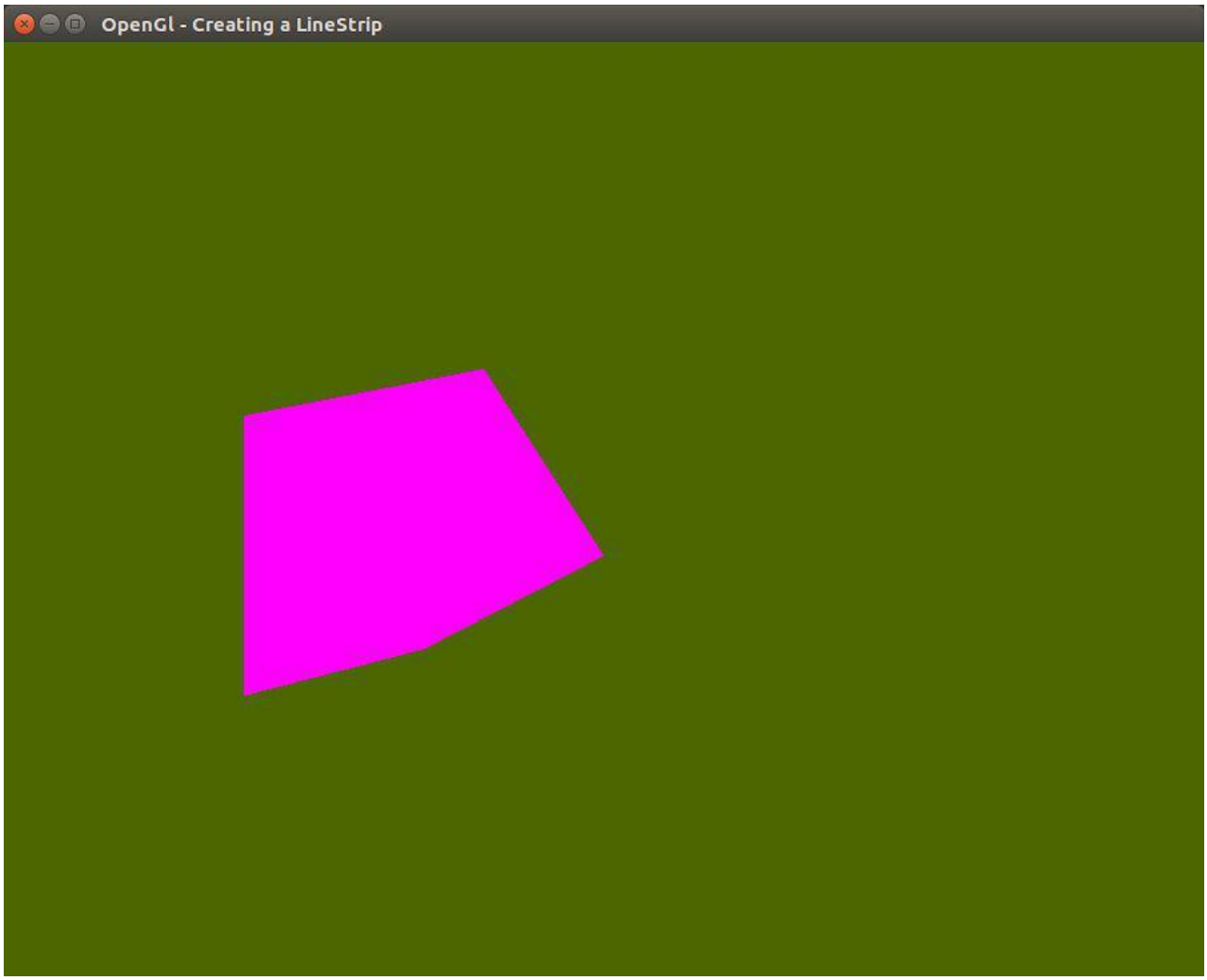
OUTPUT-

Objective- Write a Code in opengl to create a triangle strip with a given color.

6. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawTriangle(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
   glBegin(GL_TRIANGLE_STRIP);
   glVertex2f(-0.6,-0.4);
   glVertex2f(-0.6,0.2);
   glVertex2f(-0.3,-0.3);
   glVertex2f(-0.2,0.3);
   glVertex2f(0.0,-0.1);
   glEnd();
  glFlush();
}
int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a LineStrip");
 glutDisplayFunc(drawTriangle);
 glutMainLoop();
 return 0;
}
```

OUTPUT-



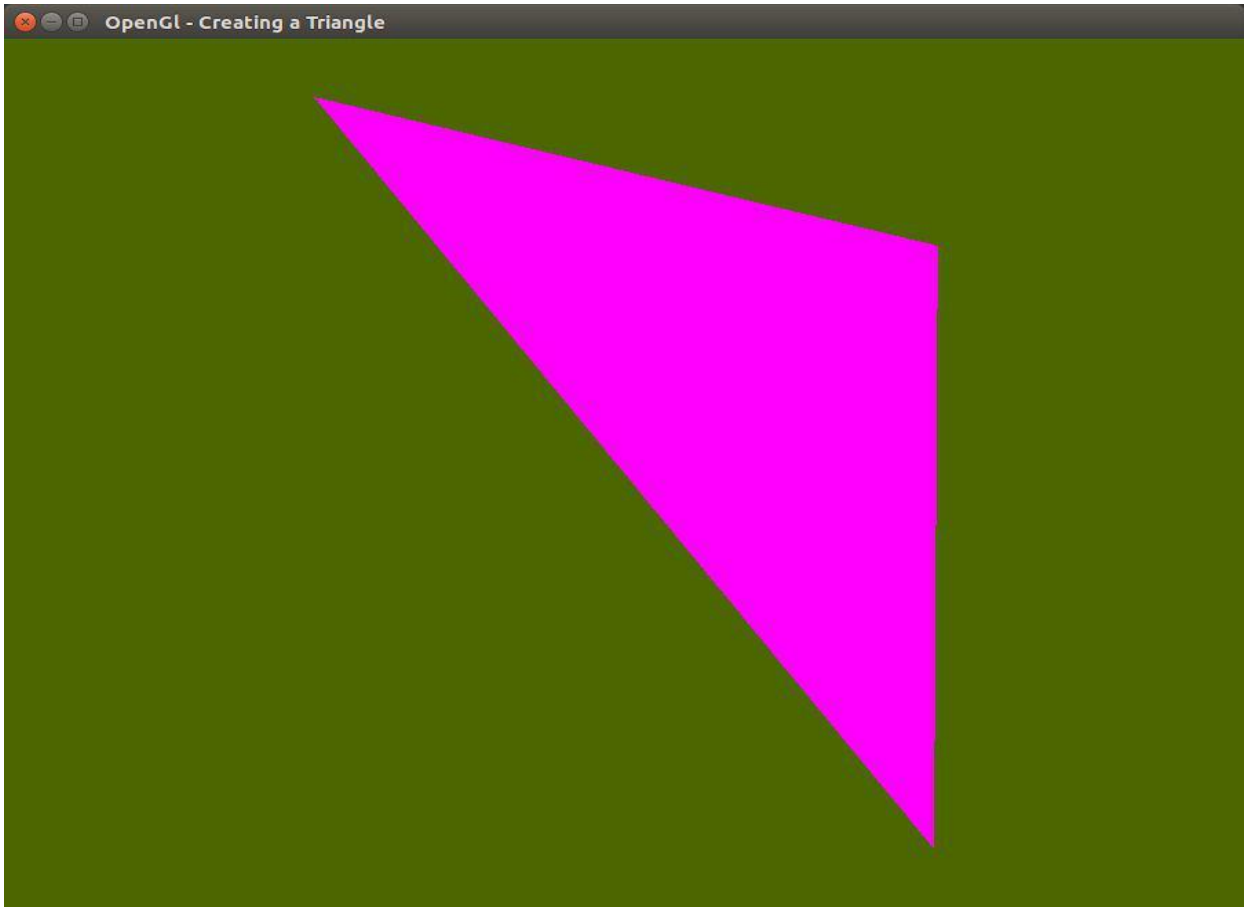
Objective- Create a rotating triangle in opengl.

7. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawShape(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glRotatef(_angle,0.0,0.0,1.0);//constant value for self rotation
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
 glBegin(GL_TRIANGLES);
   glVertex3f(0,1.0,0);
   glVertex3f(0,-1,0);
   glVertex3f(0.7,0.2,0);
   glEnd();
 glFlush();
}

void update(int value) {
    _angle += 2.0f;
    if (_angle > 360) {
        _angle -= 360;
    }

    glutPostRedisplay(); //Tell GLUT that the display has
    changed //Tell GLUT to call update again in 150 milliseconds
    glutTimerFunc(150, update, 0);
}
```

```
int main(int argc, char **argv){
glutInit(&argc,argv);
glutInitDisplayMode(GLUT_SINGLE);
glutInitWindowSize(900,700);
glutInitWindowPosition(100,100);
glutCreateWindow("OpenGl - Creating a Triangle");
glutDisplayFunc(drawShape);
glutTimerFunc(50, update, 0);//self rotation
glutMainLoop();
return 0;
}
```

OUTPUT-



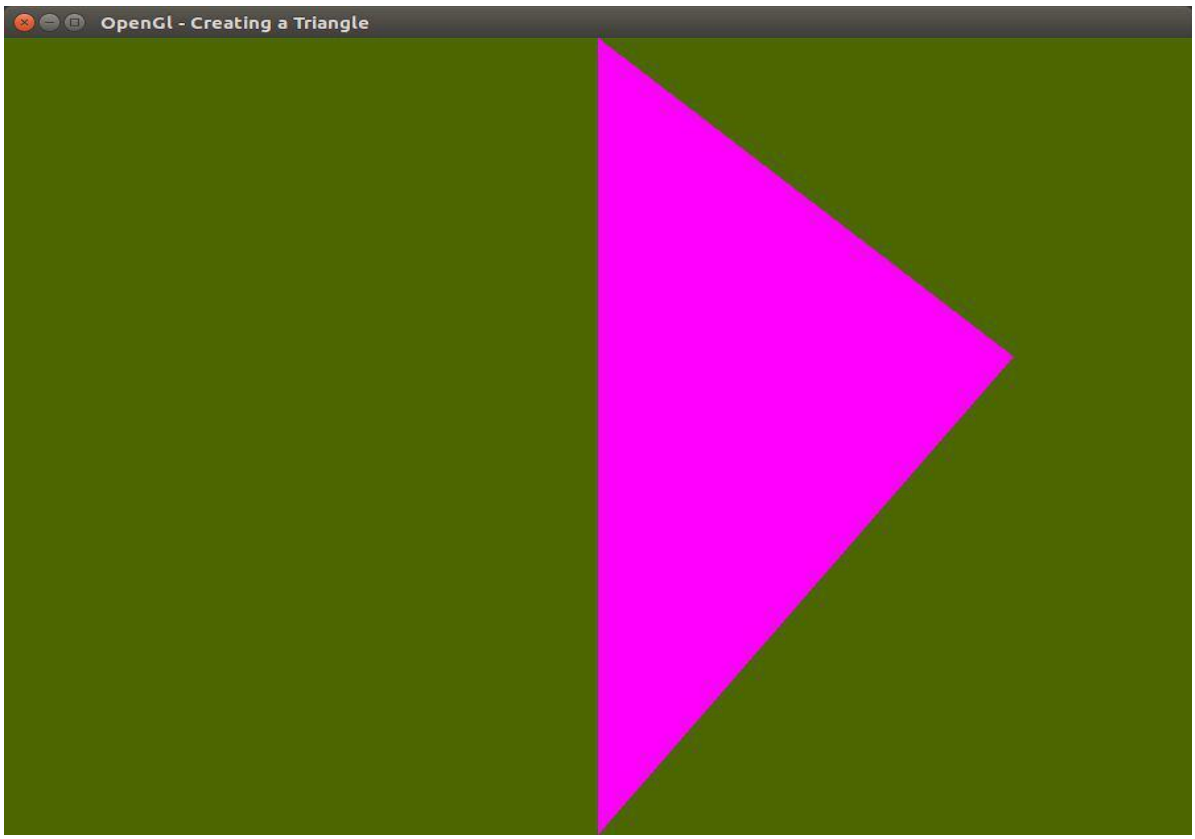
Objective- Write a Code in opengl to create a triangle with a given color and rotate it with keys.

8. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawTriangle(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glRotatef(_angle,0.0,0.0,1.0);//constant value for self rotation
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
 glBegin(GL_TRIANGLES);
   glVertex3f(0,1.0,0);
   glVertex3f(0,-1,0);
   glVertex3f(0.7,0.2,0);
   glEnd();
 glFlush();
}
void keyPress(int key,int x,int y)
{
    if(key==27)
          exit(0);
    if(key==GLUT_KEY_RIGHT)
        _angle+=5;
    if(key==GLUT_KEY_LEFT)
        _angle-=5;
    glutPostRedisplay();
   }
```

```
int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a Triangle");
 glutDisplayFunc(drawTriangle);
 glutSpecialFunc(keyPress);
 glutMainLoop();
 return 0;
}
```

OUTPUT-



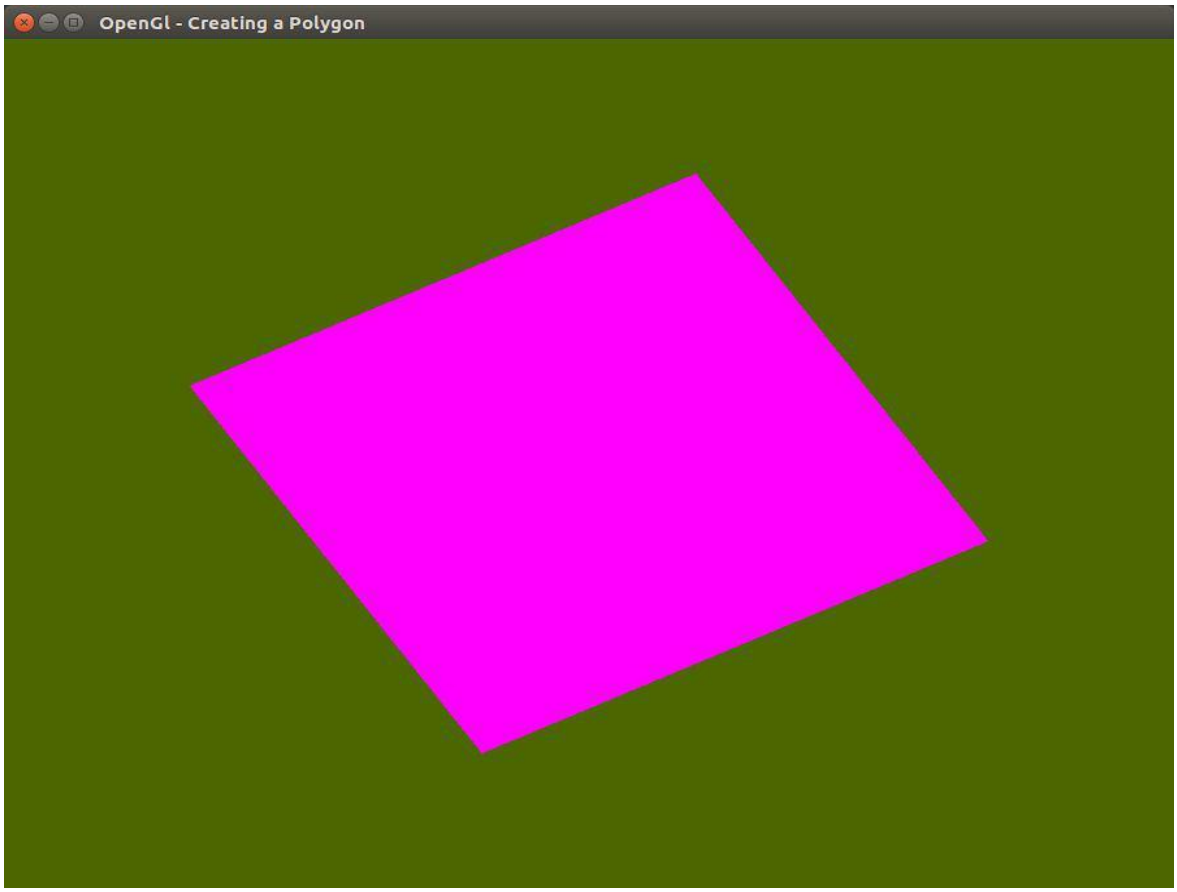
Objective- Write a Code in opengl to create a self rotating polygons with a given color.

9. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawShape(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glRotatef(_angle,0.0,0.0,1.0);//constant value for self rotation
glColor3f(1.0,0.0,1.0);
glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
 glBegin(GL_POLYGON);
   glVertex3f(-0.5,-0.5,0);
   glVertex3f(0.5,-0.5,0);
   glVertex3f(0.5,0.5,0);
   glVertex3f(-0.5,0.5,0);
   glEnd();
 glFlush();
}
void update(int value) {
    _angle += 2.0f;
    if (_angle > 360) {
        _angle -= 360;
    }

    glutPostRedisplay(); //Tell GLUT that the display has
    changed //Tell GLUT to call update again in 150 milliseconds
    glutTimerFunc(150, update, 0);
}
```

```
int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a Triangle");
 glutDisplayFunc(drawShape);
 glutTimerFunc(50, update, 0);//self rotation
 glutMainLoop();
 return 0;
}
```

OUTPUT-



Objective- Write a Code in opengl to create a self rotating cube with its each face having different color.

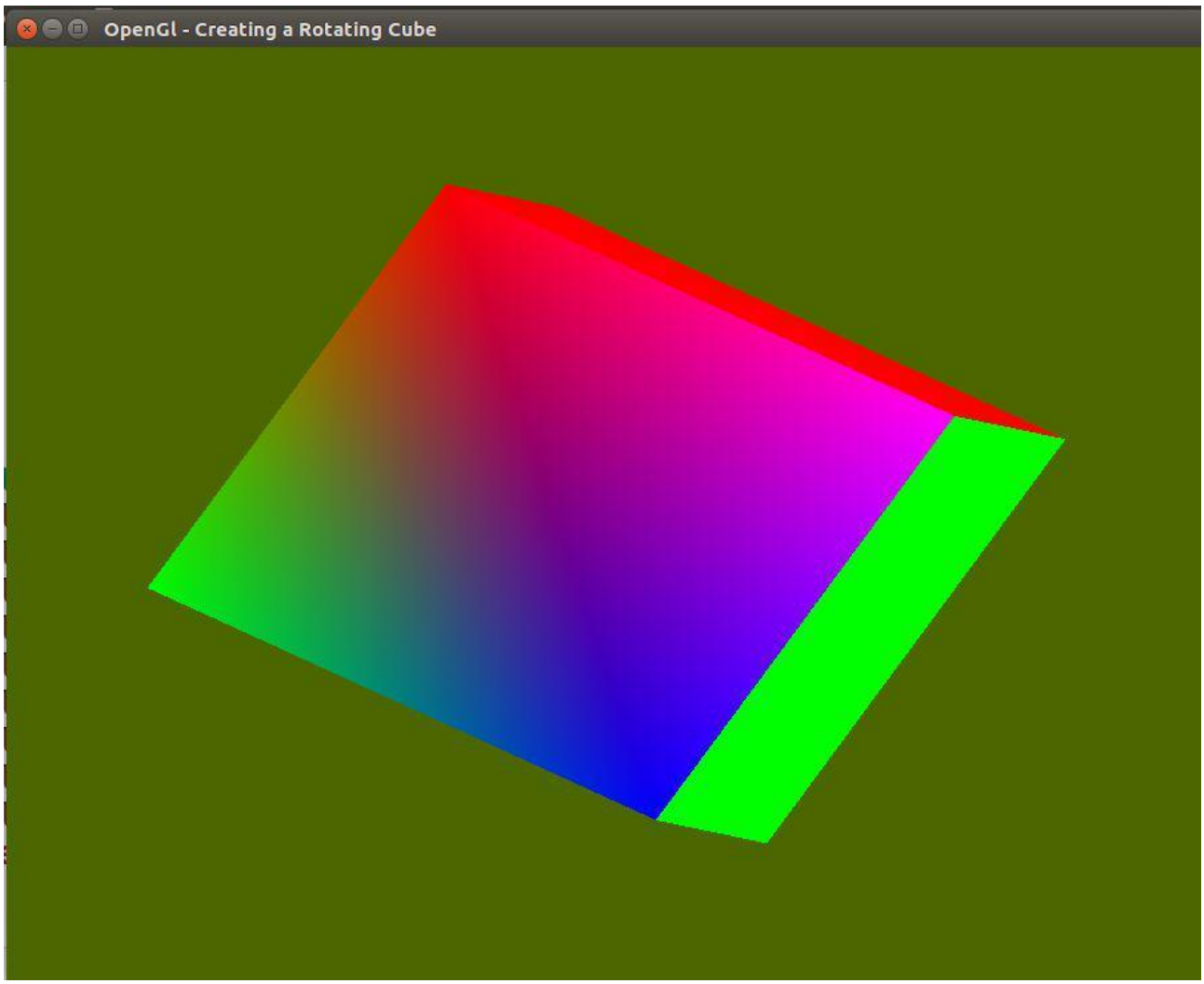
10. Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
float _angle=0.0f;
void drawTriangle(){
glClearColor(0.3,0.4,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glRotatef(15,0.1,0.0,1.0);
//glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);
 glBegin(GL_POLYGON);

  glColor3f( 1.0, 0.0, 0.0 );     glVertex3f( 0.5, -0.5, -0.5 );      // P1 is red
  glColor3f( 0.0, 1.0, 0.0 );     glVertex3f( 0.5, 0.5, -0.5 );       // P2 is green
  glColor3f( 0.0, 0.0, 1.0 );     glVertex3f( -0.5, 0.5, -0.5 );      // P3 is blue
  glColor3f( 1.0, 0.0, 1.0 );     glVertex3f( -0.5, -0.5, -0.5 );     // P4 is purple
   glEnd();

  // White side - BACK
  glBegin(GL_POLYGON);
  glColor3f( 1.0, 1.0, 1.0 );
  glVertex3f( 0.5, -0.5, 0.5 );
  glVertex3f( 0.5, 0.5, 0.5 );
  glVertex3f( -0.5, 0.5, 0.5 );
  glVertex3f( -0.5, -0.5, 0.5
  ); glEnd();

  // Purple side - RIGHT
  glBegin(GL_POLYGON);
```

```
glColor3f( 1.0, 0.0, 1.0 );
 glVertex3f( 0.5, -0.5, -0.5 );
 glVertex3f( 0.5, 0.5, -0.5 );
 glVertex3f( 0.5, 0.5, 0.5 );
 glVertex3f( 0.5, -0.5, 0.5 );
 glEnd();

 // Green side - LEFT
 glBegin(GL_POLYGON);
 glColor3f( 0.0, 1.0, 0.0 );
 glVertex3f( -0.5, -0.5, 0.5 );
 glVertex3f( -0.5, 0.5, 0.5 );
 glVertex3f( -0.5, 0.5, -0.5 );
 glVertex3f( -0.5, -0.5, -0.5
 ); glEnd();

 // Blue side - TOP
 glBegin(GL_POLYGON);
 glColor3f( 0.0, 0.0, 1.0 );
 glVertex3f( 0.5, 0.5, 0.5 );
 glVertex3f( 0.5, 0.5, -0.5 );
 glVertex3f( -0.5, 0.5, -0.5 );
 glVertex3f( -0.5, 0.5, 0.5 );
 glEnd();

 // Red side - BOTTOM
 glBegin(GL_POLYGON);
 glColor3f( 1.0, 0.0, 0.0 );
 glVertex3f( 0.5, -0.5, -0.5 );
```

```
  glVertex3f( 0.5, -0.5, 0.5 );
  glVertex3f( -0.5, -0.5, 0.5 );
  glVertex3f( -0.5, -0.5, -0.5 );
  glEnd();
glFlush();
glutSwapBuffers();
}

void update(int value) {
   _angle += 2.0f;
  /* if (_angle > 360) {
      _angle -= 360;
   }*/
     glutPostRedisplay(); //Tell GLUT that the display has
     changed //Tell GLUT to call update again in 150 milliseconds
   glutTimerFunc(300, update, 0);
}
int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB|GLUT_DEPTH);
 glutInitWindowSize(900,700);
 glutInitWindowPosition(100,100);
 glutCreateWindow("OpenGl - Creating a Triangle");
 glEnable(GL_DEPTH_TEST);
 glutDisplayFunc(drawTriangle);
 glutTimerFunc(50, update, 0);
 glutMainLoop();
 return 0;
}
```

OUTPUT-



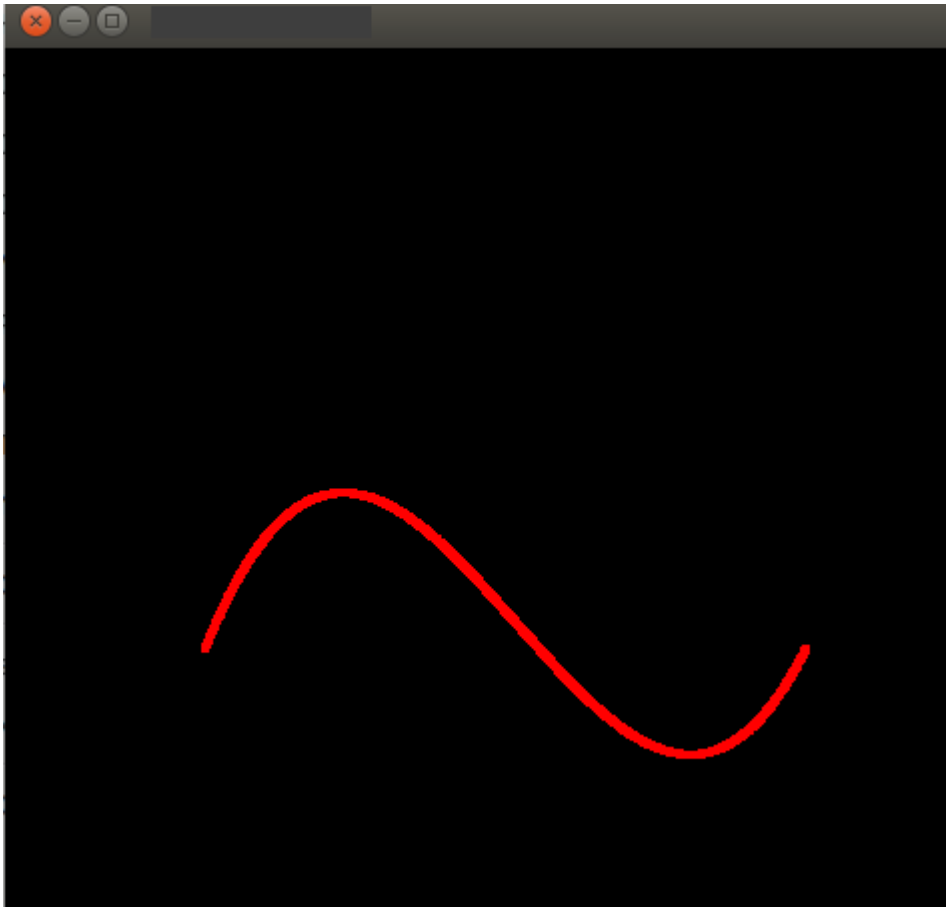
Objective- Write a Code in opengl to draw a Bezier curve with a given color.

11.Code

```
#include<GL/glut.h>
#include<GL/gl.h>
#include<stdio.h>
const int WINH= 500;
const int WINW=500;
float x_vert[4]={100.0,200.0,300.0,400.0};
float y_vert[4]={200,450,0,200};
class Point{
public:
static GLfloat getX(GLfloat x){
return (x-WINW/2)/(WINW/2);
}
static GLfloat getY(GLfloat y){
return (y-WINH/2)/(WINH/2);
}
};

void display(){
glClear(GL_COLOR_BUFFER_BIT);
glClearColor(0.0,0.0,0.0,1.0);
glPointSize(4.0);
glBegin(GL_POINTS);
glColor3f(1.0,0.0,0.0);
Point p;
GLfloat xi0=p.getX(x_vert[0]);
GLfloat xi1=p.getX(x_vert[1]);
GLfloat xi2=p.getX(x_vert[2]);
GLfloat xi3=p.getX(x_vert[3]);
GLfloat yi0=p.getY(y_vert[0]);
GLfloat yi1=p.getY(y_vert[1]);
GLfloat yi2=p.getY(y_vert[2]);
GLfloat yi3=p.getY(y_vert[3]);
GLfloat xp,yp;
for(float i=0.0; i<=1.0;  i=i+0.001){
//printf("hello %d\n",i);
xp= i*i*i*xi0+3*i*i*xi1*(1-i)+3*(1-i)*(1-i)*i*xi2+(1-i)*(1-i)*(1-i)*xi3;
yp= i*i*i*yi0+3*i*i*yi1*(1-i)+3*(1-i)*(1-i)*i*yi2+(1-i)*(1-i)*(1-i)*yi3;
glVertex2f(xp,yp);
}
glEnd();
glFlush();
}
```

```
int main(int argc, char ** argv)
{
glutInit(&argc,argv);//initialize driver
glutInitDisplayMode(GLUT_SINGLE);
glutInitWindowSize(WINH,WINW);
glutInitWindowPosition(100,100);
glutCreateWindow("Shivam Singhal");
glutDisplayFunc(display);

glutMainLoop();
}
```

OUTPUT-



Objective- Write a Code in opengl to create a sine wave with a given color.

12.Code

```
#include "GL/freeglut.h"
#include "GL/gl.h"
#include<math.h>

void drawTriangle(){
glClearColor(1,1,1,0.0);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(0,1.0,0);
glBegin(GL_LINES);
glVertex2f(0,-1);
glVertex2f(0,1);
glVertex2f(-1,0);
glVertex2f(1,0);
glEnd();
glColor3f(0.0,0.0,0.0);
GLubyte y[7]={"Y-AXIS"};
GLubyte x[7]={"X-AXIS"};
for(int k=0;k<6; k++){
glRasterPos2f(0.05+k*0.05,0.95);
glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12,y[k]);
}
glPointSize(4.0);
glColor3f(1.0,0.0,0);
glBegin(GL_POINTS);
   for(float i=0;i<=3.14*2;i+=0.005)
   glVertex2f(i/(3.14*2)-0.5,sin(i));
   glEnd();
  glFlush();
}


int main(int argc, char **argv){
 glutInit(&argc,argv);
 glutInitDisplayMode(GLUT_SINGLE);
 glutInitWindowSize(600,600);
 glutInitWindowPosition(100,100);
 glutCreateWindow("Shivam Singhal");
 glutDisplayFunc(drawTriangle);
 glutMainLoop();
 return 0;
}
```

OUTPUT-