**Session 2: Perform to design output primitives using OpenGL**

**Intended Learning Outcome:**

1. Able to demonstrate effective OpenGL programs to solve graphics programming issues including different shapes.
2. Able to appreciate the knowledge along axis (X,Y).

**Expected Skills:**

b. Make some design using OpenGL.

**Tools Required:**

1. CodeBlocks
2. OpenGL and GLUT using CodeBlocks.

**Session Detail:**

**Code for designing colorful star:**

#include <GL/gl.h>



#include <GL/glut.h>

void display(void)

{

/\* clear all pixels \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

/\* draw white polygon (rectangle) with corners at

* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0) \*/

///1st body

glColor3f (1.0, 1.0, 1.0);

glBegin(GL\_POLYGON);

glVertex3f (0.53, 0.42, 0.0);

glVertex3f (0.56, 0.34, 0.0);

glVertex3f (0.59, 0.42, 0.0);

glVertex3f (0.67, 0.45, 0.0);

glVertex3f (0.59, 0.48, 0.0);

glVertex3f (0.56, 0.56, 0.0);

glVertex3f (0.53, 0.48, 0.0);

glVertex3f (0.45, 0.45, 0.0);

glEnd();

///2nd body

glColor3f (1.0, 0.0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f (0.31, 0.42, 0.0);

glVertex3f (0.34, 0.34, 0.0);

glVertex3f (0.37, 0.42, 0.0);

glVertex3f (0.45, 0.45, 0.0);

glVertex3f (0.37, 0.48, 0.0);

glVertex3f (0.34, 0.56, 0.0);

glVertex3f (0.31, 0.48, 0.0);

glVertex3f (0.23, 0.45, 0.0);

glEnd();

///2nd body

glColor3f (0.0, 1.0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f (0.42, 0.53, 0.0);

glVertex3f (0.45, 0.45, 0.0);

glVertex3f (0.48, 0.53, 0.0);

glVertex3f (0.56, 0.56, 0.0);

glVertex3f (0.48, 0.59, 0.0);

glVertex3f (0.45, 0.67, 0.0);

glVertex3f (0.42, 0.59, 0.0);

glVertex3f (0.34, 0.56, 0.0);



glEnd();

///2nd body

glColor3f (0.0, 0.0, 1.0);

glBegin(GL\_POLYGON);

glVertex3f (0.42, 0.31, 0.0);

glVertex3f (0.45, 0.23, 0.0);

glVertex3f (0.48, 0.31, 0.0);

glVertex3f (0.56, 0.34, 0.0);

glVertex3f (0.48, 0.37, 0.0);

glVertex3f (0.45, 0.45, 0.0);

glVertex3f (0.42, 0.37, 0.0);

glVertex3f (0.34, 0.34, 0.0);

glEnd();

/\* don't wait!

* start processing buffered OpenGL routines \*/

glFlush ();

}

void init (void)

{

/\* select clearing (background) color \*/ glClearColor (0.0, 0.0, 0.0, 0.0);

/\* initialize viewing values \*/ glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);

}

/\*

* Declare initial window size, position, and display mode
* (single buffer and RGBA). Open window with "hello"
* in its title bar. Call initialization routines.
* Register callback function to display graphics.
* Enter main loop and process events.

\*/

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (100, 100);

glutCreateWindow ("Star Designing");

init ();

glutDisplayFunc(display);



glutMainLoop();

return 0; /\* ISO C requires main to return int. \*/

}

**Sample input:**

No sample input is needed.

**Sample output:**