**Lab Taks-1**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* Must submit within the given deadline in VUES to the section named Lab Tak-1
* Must include resources for all the section in the table

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **/\***  **\* GL02Primitive.cpp: Vertex, Primitive and Color**  **\* Draw Simple 2D colored Shapes: quad, triangle and polygon.**  **\*/**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Initialize OpenGL Graphics \*/**  **void initGL() {**  **// Set "clearing" or background color**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **}**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color**  **glLineWidth(2.0);**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(1.0f, 0.0f, 0.0f); //**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(0.5f, 0.0f);**  **glVertex2f(0.5f, 0.5f);**  **glVertex2f(0.0f, 0.5f);**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(500, 500); // Set the window's initial width & height**  **glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title**  **glutInitWindowPosition(50, 50); // Position the window's initial top-left corner**  **glutDisplayFunc(display); // Register callback handler for window re-paint event**  **initGL(); // Our own OpenGL initialization**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **/\***  **\* GL02Primitive.cpp: Vertex, Primitive and Color**  **\* Draw Simple 2D colored Shapes: quad, triangle and polygon.**  **\*/**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Initialize OpenGL Graphics \*/**  **void initGL() {**  **// Set "clearing" or background color**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **}**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color**  **glLineWidth(2.0);**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(1.0f, 0.0f, 0.0f); //**  **glVertex2f(-0.1f, 0.0f);**  **glVertex2f(0.8f, 0.0f);**  **glVertex2f(0.6f, 0.6f);**  **glVertex2f(0.1f, 0.6f);**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(500, 500); // Set the window's initial width & height**  **glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title**  **glutInitWindowPosition(50, 50); // Position the window's initial top-left corner**  **glutDisplayFunc(display); // Register callback handler for window re-paint event**  **initGL(); // Our own OpenGL initialization**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **/\***  **\* GL02Primitive.cpp: Vertex, Primitive and Color**  **\* Draw Simple 2D colored Shapes: quad, triangle and polygon.**  **\*/**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Initialize OpenGL Graphics \*/**  **void initGL() {**  **// Set "clearing" or background color**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **}**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color**  **glLineWidth(6.0);**  **glColor3f(1.0f, 0.0f, 0.0f); //**  **//divider line**  **glBegin(GL\_LINES);**  **glColor3f(1.0,0.0,0.0);**  **glVertex2f(-1.0,0);**  **glVertex2f(1.0,0);**  **glColor3f(0.0,0.0,0.0);**  **glVertex2f(0,1);**  **glVertex2f(0,-1);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(1.0,0.0,0.0);**  **glVertex2f(-0.8,0.2);**  **glVertex2f(-0.3,0.2);**  **glVertex2f(-0.3,0.6);**  **glVertex2f(-0.8,0.6);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0,0.7,0.6);**  **glVertex2f(0.2,-0.2);**  **glVertex2f(0.4,-0.2);**  **glVertex2f(0.4,-0.4);**  **glVertex2f(0.2,-0.4);**  **glEnd();**  **// left side triangle**  **glBegin(GL\_TRIANGLES);**  **glColor3f(0.7,0.7,0.6);**  **glVertex2f(-0.2,-0.2);**  **glVertex2f(-0.2,-0.8);**  **glVertex2f(-0.8,-0.5);**  **glEnd();**  **// triingle**  **glBegin(GL\_TRIANGLES);**  **glColor3f(1.0,1.0,0.0);**  **glVertex2f(0.2,0.2);**  **glVertex2f(0.8,0.2);**  **glVertex2f(0.5,0.8);**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(500, 500); // Set the window's initial width & height**  **glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title**  **glutInitWindowPosition(50, 50); // Position the window's initial top-left corner**  **glutDisplayFunc(display); // Register callback handler for window re-paint event**  **initGL(); // Our own OpenGL initialization**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |