**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-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **//glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glPointSize(13.0);**  **glBegin(GL\_LINES);**  **glColor3ub(20,20,20);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(-2.0f, 2.0f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(20,20,20);**  **glVertex2f(-2.0f, -2.0f);**  **glVertex2f(2.0f, -2.0f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(20,20,20);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(2.0f, -2.0f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(20,20,20);**  **glVertex2f(-2.0f,- 2.0f);**  **glVertex2f(-2.0f, 2.0f);**  **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**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **glutInitWindowSize(320, 320); // Set the window's initial width & height**  **gluOrtho2D(-5, 5, -7, 7);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glPointSize(10.0);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_QUADS); // Each set of 4 vertices form a quad**  **glColor3ub(255, 0, 0); // Red**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(6.0f, 0.0f);**  **glVertex2f(4.0f, 3.0f);**  **glVertex2f(2.0f, 3.0f);**  **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**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **glutInitWindowSize(300, 380); // Set the window's initial width & height**  **gluOrtho2D(-5, 15,-10, 20);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(2);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(21, 21, 21); // Black**  **glVertex2f(-1.0f, 0.0f); // x, y**  **glVertex2f(1.0f, 0.0f);**  **glEnd();**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(20, 20, 20); // Black**  **glVertex2f(0.0f, 1.0f);**  **glVertex2f(0.0f, -1.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-0.9f, 0.2f);**  **glVertex2f(-0.3f, 0.2f);**  **glVertex2f(-0.3f, 0.8f);**  **glVertex2f(-0.9f, 0.8f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(70, 150, 57); // Green**  **glVertex2f(0.6f, 0.2f);**  **glVertex2f(0.9f, 0.4f);**  **glVertex2f(0.6f, 0.6f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(70, 150, 57); // Green**  **glVertex2f(0.2f, 0.3f);**  **glVertex2f(0.6f, 0.3f);**  **glVertex2f(0.6f, 0.5f);**  **glVertex2f(0.2f, 0.5f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(114, 39, 135); // Violet**  **glVertex2f(-0.3f, -0.2f);**  **glVertex2f(-0.8f, -0.5f);**  **glVertex2f(-0.3f, -0.8f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(1.0f, 1.0f, 0.0f); // Yellow**  **glVertex2f(0.2f, -0.7f);**  **glVertex2f(0.8f, -0.7f);**  **glVertex2f(0.5f, -0.3f);**  **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**  **glutCreateWindow("OpenGL Setup"); // Create a window with the given title**  **glutInitWindowSize(320, 320); // Set the window's initial width & height**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |