**Lab Taks-1**

Submission Guidelines-

* Rename the file with your serial number only
* 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 rectangleline()**  **{**  **glColor3ub(0,0,0);**  **glBegin(GL\_LINE\_LOOP);**    **glVertex2f(-0.4f, 0.3f);**  **glVertex2f(0.4f, 0.3f);**  **glVertex2f(0.4f, -0.3f);**  **glVertex2f(-0.4f, -0.3f);**  **glEnd();**  **}**  **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);**      **rectangleline();**  **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(1080, 1080); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **gluOrtho2D(-1.5,1.5,-1.5,1.5);**  **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 trapezium()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(255,0,0);**  **glVertex2f(-0.2f,0.2f);**  **glVertex2f(0.4f,0.2f);**  **glVertex2f(0.8f,-0.4f);**  **glVertex2f(-0.6f,-0.4f);**  **glEnd();**  **}**  **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);**  **trapezium();**  **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(1080, 1080); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **gluOrtho2D(-1.5,1.5,-1.5,1.5);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

| **Question-**  Draw the object- |
| --- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **void initGL()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // White and opaque**  **}**  **/\* Draw the black axes \*/**  **void axis()**  **{**  **glBegin(GL\_LINES);**  **glColor3ub(0,0,0);**  **glVertex2f(-1.3f, -0.36f);**  **glVertex2f(1.3f, -0.36f);**  **glVertex2f(-0.12f, -1.3f);**  **glVertex2f(-0.12f, 0.56);**  **glEnd();**  **}**  **/\* Draw the red square \*/**  **void quad()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(255,0,0);**  **glVertex2f(-1.03f, -0.14f);**  **glVertex2f(-0.34f, -0.14f);**  **glVertex2f(-0.34f, 0.44f);**  **glVertex2f(-1.03f, 0.44f);**  **glEnd();**  **}**  **/\* Draw the green arrow \*/**  **void arrow()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,255,0);**  **glVertex2f(0.2f, 0);**  **glVertex2f(0.89f, 0);**  **glVertex2f(0.89f, 0.33);**  **glVertex2f(0.2f, 0.33f);**  **glEnd();**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(0.89f, 0.48f);**  **glVertex2f(0.89f, -0.16f);**  **glVertex2f(1.21f, 0.16f);**  **glEnd();**  **}**  **/\* Draw the purple triangle \*/**  **void triangle()**  **{**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(255,255,0);**  **glVertex2f(0.2f, -1.03f);**  **glVertex2f(0.93f, -1.02f);**  **glVertex2f(0.56f, -0.53f);**  **glEnd();**  **}**  **/\* Draw the yellow triangle \*/**  **void triangle2()**  **{**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(128,0,128);**  **glVertex2f(-1.03f, -0.79f);**  **glVertex2f(-0.4f, -1.16f);**  **glVertex2f(-0.4f, -0.4f);**  **glEnd();**  **}**  **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);**  **axis();**  **quad();**  **arrow();**  **triangle();**  **triangle2();**  **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(1080, 1080); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **gluOrtho2D(-1.5,1.5,-1.5,1.5);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |