**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(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(5);**  **glColor3ub(255,255,255);**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.2,0.2);**  **glVertex2f(0.8,0.2);**  **glVertex2f(0.8,0.6);**  **glVertex2f(0.2,0.6);**  **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(320, 320); // 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(-1,1,-1,1);**  **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)**  **glPointSize(5);**  **glColor3ub(255,0,0);**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.2,0.2);**  **glVertex2f(1.2,0.2);**  **glVertex2f(1,0.6);**  **glVertex2f(0.4,0.6);**  **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(320, 320); // 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(-2,2,-2,2);**  **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**  **// Function to draw axes**  **void drawAxes() {**  **glColor3f(0, 0, 0); // Black color**  **glLineWidth(2.0);**  **glBegin(GL\_LINES);**  **glVertex2f(-6.0, 0.0);**  **glVertex2f(6.0, 0.0);**  **glVertex2f(0.0, -6.0);**  **glVertex2f(0.0, 6.0);**  **glEnd();**  **}**  **// Function to draw a red square**  **void drawRedSquare() {**  **glColor3f(1.0, 0.0, 0.0); // Red color**  **glBegin(GL\_QUADS);**  **glVertex2f(-5,1);**  **glVertex2f(-5,5);**  **glVertex2f(-1,5);**  **glVertex2f(-1,1);**  **glEnd();**  **}**  **// Function to draw a green arrow**  **void drawGreenArrow() {**  **glColor3f(0.0, 1.0, 0.0); // Green color**  **glBegin(GL\_QUADS);**  **glVertex2f(1,2);**  **glVertex2f(1,4);**  **glVertex2f(4,4);**  **glVertex2f(4,2);**  **glEnd();**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(4,1);**  **glVertex2f(4,5);**  **glVertex2f(5.5,3);**  **glEnd();**  **}**  **// Function to draw a purple left triangle**  **void drawPurpleTriangle() {**  **glColor3f(0.5, 0.0, 0.5); // Purple color**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(-1,-1);**  **glVertex2f(-1,-5);**  **glVertex2f(-5,-3);**  **glEnd();**  **}**  **// Function to draw a yellow triangle**  **void drawYellowTriangle() {**  **glColor3f(1.0, 1.0, 0.0); // Yellow color**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(1,-5);**  **glVertex2f(5,-5);**  **glVertex2f(3,-1);**  **glEnd();**  **}**  **// Display function**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **drawAxes();**  **drawRedSquare();**  **drawGreenArrow();**  **drawPurpleTriangle();**  **drawYellowTriangle();**  **glFlush();**  **}**  **// Main function**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**  **glutInitWindowSize(500, 500);**  **glutCreateWindow("OpenGL Setup Test");**  **glClearColor(1.0, 1.0, 1.0, 1.0);**  **glMatrixMode(GL\_PROJECTION);**  **glLoadIdentity();**  **gluOrtho2D(-6, 6, -6, 6);**  **glutDisplayFunc(display);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |