**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)-**  4[1] |
| **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(255.0f, 255.0f, 255.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(5);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_LINES);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(-5.0f,-3.0f);**  **glVertex2f(5.0f,-3.0f);**  **glVertex2f(5.0f,-3.0f);**  **glVertex2f(5.0f,3.0f);**  **glVertex2f(5.0f,3.0f);**  **glVertex2f(-5.0f,3.0f);**  **glVertex2f(-5.0f,3.0f);**  **glVertex2f(-5.0f,-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(320, 320);**  **gluOrtho2D(-8,8,-8,8); // 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)-**  4 |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-**  5 |
| **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(255.0f, 255.0f, 255.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(10);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 0, 0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(-4.0f,0.0f);**  **glVertex2f(4.0f,0.0f);**  **glVertex2f(3.0f,4.0f);**  **glVertex2f(-3.0f,4.0f);**  **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);**  **gluOrtho2D(-10,10,-10,10); // 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)-**  **2** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-**  **7** |
| **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(255.0f, 255.0f, 255.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(5);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_LINES);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(-8.0f,0.0f);**  **glVertex2f(8.0f,0.0f);**  **glVertex2f(0.0f,8.0f);**  **glVertex2f(0.0f,-8.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 0, 0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(-6.0f,2.0f);**  **glVertex2f(-2.0f,2.0f);**  **glVertex2f(-2.0f,6.0f);**  **glVertex2f(-6.0f,6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(138,43,226); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(-2.0f,-2.0f);**  **glVertex2f(-6.0f,-4.0f);**  **glVertex2f(-2.0f,-6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255,255,0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(4.0f,-2.0f);**  **glVertex2f(2.0f,-5.0f);**  **glVertex2f(6.0f,-5.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(124,252,0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(5.0f,3.0f);**  **glVertex2f(5.0f,5.0f);**  **glVertex2f(1.0f,5.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(124,252,0); // Each set of 4 vertices form a quad// Red // x, y**  **glVertex2f(5.0f,6.0f);**  **glVertex2f(5.0f,2.0f);**  **glVertex2f(7.0f,4.0f);**  **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);**  **gluOrtho2D(-8,8,-8,8); // 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)-**  **3** |