**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 given in the class in VUES to the section named Lab Tak-1
* Must include resources for all the section in the table

|  |
| --- |
| **Question-1**  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 rectangle()**  **{**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.6f, -0.6f);**  **glVertex2f(0.8f, -0.6f);**  **glVertex2f(0.8f, 0.4f);**  **glVertex2f(-0.6f, 0.4f);**  **glEnd();**  **}**  **void rectangleline()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glVertex2f(-0.6f, -0.6f);**  **glVertex2f(0.8f, -0.6f);**  **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);**  **glColor3ub(255,255,255);**  **rectangle();**  **glColor3ub(0,0,0);**  **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-2**  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);**  **glVertex2f(-0.6f,-0.4f);**  **glVertex2f(0.8f,-0.4f);**  **glVertex2f(0.4f,0.2f);**  **glVertex2f(-0.2f,0.2f);**  **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);**  **glColor3ub(255,0,0);**  **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-3**  Draw the object-  Octagon Shape | Area & Angles - Video & Lesson Transcript | Study.com |
| **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 octagon()**  **{**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.4f, -0.8f);**  **glVertex2f(0.2f, -0.8f);**  **glVertex2f(0.6f, -0.4f);**  **glVertex2f(0.6f, 0.2f);**  **glVertex2f(0.2, 0.6f);**  **glVertex2f(-0.4f, 0.6f);**  **glVertex2f(-0.8f, 0.2f);**  **glVertex2f(-0.8, -0.4f);**  **glEnd();**  **}**  **void octagonline()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glVertex2f(-0.4f, -0.8f);**  **glVertex2f(0.2f, -0.8f);**  **glVertex2f(0.6f, -0.4f);**  **glVertex2f(0.6f, 0.2f);**  **glVertex2f(0.2, 0.6f);**  **glVertex2f(-0.4f, 0.6f);**  **glVertex2f(-0.8f, 0.2f);**  **glVertex2f(-0.8, -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);**  **glColor3ub(255,0,0);**  **octagon();**  **glColor3ub(0,0,0);**  **octagonline();**  **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-4**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **void Star() {**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.02, -0.06);**  **glVertex2f(0.58, -0.43);**  **glVertex2f(0.4, 0.2);**  **glVertex2f(0.94, 0.57);**  **glVertex2f(0.25, 0.59);**  **glVertex2f(0.03, 1.21);**  **glVertex2f(-0.2, 0.6);**  **glVertex2f(-0.89, 0.58);**  **glVertex2f(-0.34, 0.2);**  **glVertex2f(-0.54, -0.42);**  **glEnd();**  **}**  **void Starboundary() {**  **glBegin(GL\_LINE\_LOOP);**  **glVertex2f(0.02, -0.06);**  **glVertex2f(0.58, -0.43);**  **glVertex2f(0.4, 0.2);**  **glVertex2f(0.94, 0.57);**  **glVertex2f(0.25, 0.59);**  **glVertex2f(0.03, 1.21);**  **glVertex2f(-0.2, 0.6);**  **glVertex2f(-0.89, 0.58);**  **glVertex2f(-0.34, 0.2);**  **glVertex2f(-0.54, -0.42);**  **glEnd();**  **}**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glColor3ub(255, 0, 0);**  **Star();**  **glColor3ub(0, 0, 0);**  **glLineWidth(4);**  **Starboundary();**  **glFlush(); // Render now**  **}**  **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); // Set up 2D orthographic viewing region**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-5**  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);**  **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);**  **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);**  **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);**  **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);**  **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);**  **glColor3ub(0,0,0);**  **axis();**  **glColor3ub(255,0,0);**  **quad();**  **glColor3ub(0,255,0);**  **arrow();**  **glColor3ub(255,255,0);**  **triangle();**  **glColor3ub(128,0,128);**  **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)-** |

|  |
| --- |
| **Question-6**  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 the red polygon**  **void drawRedPolygon() {**  **glBegin(GL\_POLYGON);**  **glVertex2f(-1.27f, 0.03f);**  **glVertex2f(-0.06f, -0.56f);**  **glVertex2f(1.16f, 0.02f);**  **glVertex2f(-0.05f, 0.61f);**  **glEnd();**  **}**  **// Function to draw the yellow triangle**  **void drawYellowTriangle() {**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.72f, -1.1f);**  **glVertex2f(0.57f, -1.1f);**  **glVertex2f(-0.08f, -0.12f);**  **glEnd();**  **}**  **// Function to draw the blue rectangle**  **void drawBlueRectangle() {**  **glBegin(GL\_LINE\_LOOP);**  **glVertex2f(-0.61f, 0.31f);**  **glVertex2f(-0.61f, -0.15f);**  **glVertex2f(0.43f, -0.15f);**  **glVertex2f(0.43f, 0.31f);**  **glEnd();**  **}**  **// Function to draw the orange lines**  **void drawOrangeLines() {**  **glBegin(GL\_LINES);**  **glVertex2f(-0.94f, -0.83f);**  **glVertex2f(0.63f, 0.91f);**  **glVertex2f(-0.76f, 0.78f);**  **glVertex2f(0.83f, -0.75f);**  **glEnd();**  **}**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(2);**  **// Draw filled red polygon**  **glColor3ub(255, 0, 0);**  **drawRedPolygon();**  **// Draw filled yellow triangle**  **glColor3ub(255, 255, 0);**  **drawYellowTriangle();**  **// Draw blue rectangle**  **glColor3ub(25,189,255);**  **drawBlueRectangle();**  **// Draw orange lines**  **glColor3ub(255, 165, 0);**  **drawOrangeLines();**  **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); // Set the coordinate system for the window**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |