**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)-**  **A grid with lines and dots** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Program entry point \*/**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void task1() {**  **glClearColor(256.0f, 256.0f, 256.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glPointSize(5.0);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_POINTS); // Each set of 4 vertices form a quad**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, 4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(5.0f, 4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, -4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(5.0f,-4.0f); // x, y**  **glEnd();**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, 4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, -4.0f); // x, y**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, -4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(5.0f, -4.0f); // x, y**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(5.0f, -4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(5.0f, 4.0f); // x, y**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(5.0f, 4.0f); // x, y**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, 4.0f); // x, y**  **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**  **glutReshapeWindow (1024,720);**  **glutDisplayFunc(task1); // Register display callback handler for window re-paint**  **gluOrtho2D(-10,10,-10,10);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-2**  Draw the object- |
| **Graph Plot (Picture)-**  **A graph with lines and dots  Description automatically generated** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Program entry point \*/**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void task2() {**  **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(5.0);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-3.0f, 2.0f); // x, y**  **glColor3f(1.0f, 0.0f, 1.0f); // Violet**  **glVertex2f(-6.0f, -4.0f); // x, y**  **glColor3f(1.0f, 1.0f, 0.0f); // Red**  **glVertex2f(5.0f, -4.0f); // x, y**  **glColor3f(1.0f, 1.0f, 1.0f); // Violet**  **glVertex2f(2.0f, 2.0f); // x, y**  **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**  **glutReshapeWindow (1024,720);**  **glutDisplayFunc(task2); // Register display callback handler for window re-paint**  **gluOrtho2D(-10,10,-10,10);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-3**  Draw the object-  Octagon Shape | Area & Angles - Video & Lesson Transcript | Study.com |
| **Graph Plot (Picture)-**  **A graph with lines and dots  Description automatically generated** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Program entry point \*/**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void drawText(const char\* text, float x, float y) {**  **glColor3f(0.0f, 0.0f, 0.0f); // Black color for text**  **glRasterPos2f(x, y); // Set the position for text**  **while (\*text) {**  **glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, \*text);**  **text++;**  **}**  **}**  **void task3(){**  **glClearColor(256.0f, 256.0f, 256.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **drawText("Octagon", -2.5f, 10.0f); // Draw the text above the octagon**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-4.0f, -8.0f); // x, y**  **glVertex2f(4.0f, -8.0f); // x, y**  **glVertex2f(4.0f, -8.0f); // x, y**  **glVertex2f(8.0f, -4.0f); // x, y**  **glVertex2f(8.0f, -4.0f); // x, y**  **glVertex2f(8.0f, 4.0f); // x, y**  **glVertex2f(8.0f, 4.0f); // x, y**  **glVertex2f(4.0f, 8.0f); // x, y**  **glVertex2f(0.4f, 8.0f); // x, y**  **glVertex2f(-0.4f, 8.0f); // x, y**  **glVertex2f(-4.0f, 8.0f); // x, y**  **glVertex2f(-8.0f, 4.0f); // x, y**  **glVertex2f(-8.0f, 4.0f); // x, y**  **glVertex2f(-8.0f, -4.0f); // x, y**  **glVertex2f(-8.0f, -4.0f); // x, y**  **glVertex2f(-4.0f, -8.0f); // x, y**  **glEnd();**  **glLineWidth(5.0f);**  **glBegin(GL\_LINE\_LOOP);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-4.0f, -8.0f); // x, y**  **glVertex2f(4.0f, -8.0f); // x, y**  **glVertex2f(4.0f, -8.0f); // x, y**  **glVertex2f(8.0f, -4.0f); // x, y**  **glVertex2f(8.0f, -4.0f); // x, y**  **glVertex2f(8.0f, 4.0f); // x, y**  **glVertex2f(8.0f, 4.0f); // x, y**  **glVertex2f(4.0f, 8.0f); // x, y**  **glVertex2f(0.4f, 8.0f); // x, y**  **glVertex2f(-0.4f, 8.0f); // x, y**  **glVertex2f(-4.0f, 8.0f); // x, y**  **glVertex2f(-8.0f, 4.0f); // x, y**  **glVertex2f(-8.0f, 4.0f); // x, y**  **glVertex2f(-8.0f, -4.0f); // x, y**  **glVertex2f(-8.0f, -4.0f); // x, y**  **glVertex2f(-4.0f, -8.0f); // x, y**  **glEnd();**  **glFlush();**  **}**  **/\* 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**  **glutReshapeWindow (720,720);**  **glutDisplayFunc(task3); // Register display callback handler for window re-paint**  **gluOrtho2D(-20,20,-20,20);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Draw the object- |
| **Graph Plot (Picture)-**  **A screen shot of a graph  Description automatically generated** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Program entry point \*/**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void task4()**  **{**  **glClearColor(256.0f, 256.0f, 256.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-6.0f, 4.0f); // E**  **glVertex2f(-2.0f, 2.0f); // D**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(-2.0f, 2.0f); // D**  **glVertex2f(-4.0f, -2.0f); // A**  **glVertex2f(-4.0f, -2.0f); // A**  **glVertex2f(0.0f, 0.0f); // H**  **glVertex2f(0.0f, 0.0f); // H**  **glVertex2f(4.0f, -2.0f); // C**  **glVertex2f(4.0f, -2.0f); // C**  **glVertex2f(2.0f, 2.0f); // F**  **glVertex2f(2.0f, 2.0f); // F**  **glVertex2f(6.0f, 4.0f); // G**  **glVertex2f(6.0f, 4.0f); // G**  **glVertex2f(1.8f, 4.0f); // J**  **glVertex2f(1.8f, 4.0f); // J**  **glVertex2f(0.0f, 8.0f); // B**  **glVertex2f(0.0f, 8.0f); // B**  **glVertex2f(-1.8f, 4.0f); // I**  **glVertex2f(-1.8f, 4.0f); // I**  **glVertex2f(-6.0f, 4.0f); // E**  **glVertex2f(-6.0f, 4.0f); // E**  **glVertex2f(-2.0f, 2.0f); // D**  **glEnd();**  **glLineWidth(3.0f);**  **glBegin(GL\_LINE\_LOOP);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-2.0f, 2.0f); // D**  **glVertex2f(-4.0f, -2.0f); // A**  **glVertex2f(-4.0f, -2.0f); // A**  **glVertex2f(0.0f, 0.0f); // H**  **glVertex2f(0.0f, 0.0f); // H**  **glVertex2f(4.0f, -2.0f); // C**  **glVertex2f(4.0f, -2.0f); // C**  **glVertex2f(2.0f, 2.0f); // F**  **glVertex2f(2.0f, 2.0f); // F**  **glVertex2f(6.0f, 4.0f); // G**  **glVertex2f(6.0f, 4.0f); // G**  **glVertex2f(1.8f, 4.0f); // J**  **glVertex2f(1.8f, 4.0f); // J**  **glVertex2f(0.0f, 8.0f); // B**  **glVertex2f(0.0f, 8.0f); // B**  **glVertex2f(-1.8f, 4.0f); // I**  **glVertex2f(-1.8f, 4.0f); // I**  **glVertex2f(-6.0f, 4.0f); // E**  **glEnd();**  **glFlush();**  **}**  **/\* 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**  **glutReshapeWindow (720,720);**  **glutDisplayFunc(task4); // Register display callback handler for window re-paint**  **gluOrtho2D(-20,20,-20,20);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screen shot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-5**  Draw the object- |
| **Graph Plot (Picture)-**  **A screenshot of a graph  Description automatically generated** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Program entry point \*/**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void task5()**  **{**  **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(2.5);**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glVertex2f(-9.5f, 0.0f); // x, y**  **glVertex2f(9.5f, 0.0f); // x, y**  **glVertex2f(0.0f, 9.5f); // x, y**  **glVertex2f(0.0f, -9.5f); // x, y**  **glEnd();**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-5.0, 1.0f); // x, y**  **glVertex2f(-1.0f, 1.0f); // x, y**  **glVertex2f(-1.0f, 1.0f); // x, y**  **glVertex2f(-1.0f, 5.0f); // x, y**  **glVertex2f(-1.0f, 5.0f); // x, y**  **glVertex2f(-5.0f, 5.0f); // x, y**  **glVertex2f(-5.0f, 1.0f); // x, y**  **glVertex2f(-1.0f, 1.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(0.5f, 0.0f, 0.5f); // Purple**  **glVertex2f(-5.0f, -3.0f); // N**  **glVertex2f(-1.0f, -1.0f); // L**  **glVertex2f(-1.0f, -1.0f); // L**  **glVertex2f(-1.0f, -5.0f); // M**  **glEnd();**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(0.5f, 0.0f, 0.5f); // Purple**  **glVertex2f(-1.0f, -5.0f); // M**  **glVertex2f(-5.0f, -0.3f); // N**  **glEnd();**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 1.0f, 0.0f); // Yellow**  **glVertex2f(0.1f, -0.45f); // x, y**  **glVertex2f(0.5f, -0.45f); // x, y**  **glVertex2f(0.5f, -0.45f); // x, y**  **glVertex2f(0.3f, -0.15f); // x, y**  **glVertex2f(0.3f, -0.15f); // x, y**  **glVertex2f(0.1f, -0.45f); // x, y**  **glEnd();**  **// SQUARE**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(0.0f, 1.0f, 0.0f); // Green**  **glVertex2f(1.5f, 2.0f); // x, y**  **glVertex2f(5.5f, 2.0f); // x, y**  **glVertex2f(5.5f, 2.0f); // x, y**  **glVertex2f(5.5f, 4.0f); // x, y**  **glVertex2f(5.5f, 4.0f); // x, y**  **glVertex2f(1.5f, 4.0f); // x, y**  **glVertex2f(1.5f, 4.0f); // x, y**  **glVertex2f(1.5f, 2.0f);**  **glEnd();**  **// TRIANGLE**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(0.0f, 1.0f, 0.0f); // Green**  **glVertex2f(5.5f, 1.0f); // x, y**  **glVertex2f(7.5f, 3.0f); // x, y**  **glVertex2f(7.5f, 3.0f); // x, y**  **glVertex2f(5.5f, 5.0f); // x, y**  **glVertex2f(5.5f, 5.0f); // x, y**  **glVertex2f(5.5f, 1.0f); // x, y**  **glEnd();**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 1.0f, 0.0f); // Yellow**  **glVertex2f(1.0f, -4.5f); // P**  **glVertex2f(5.0f, -4.5f); // Q**  **glVertex2f(5.0f, -4.5f); // Q**  **glVertex2f(3.0f, -1.5f); // O**  **glEnd();**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 1.0f, 0.0f); // Yellow**  **glVertex2f(3.0f, -1.5f); // O**  **glVertex2f(-1.0f, -4.5f); // P**  **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**  **glutReshapeWindow (720,720);**  **glutDisplayFunc(task5); // Register display callback handler for window re-paint**  **gluOrtho2D(-20,20,-20,20);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-6**  Draw the object- |
| **Graph Plot (Picture)-**  **A screenshot of a graph  Description automatically generated** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Program entry point \*/**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void task6()**  **{**  **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(2.5);**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 1.0f, 0.0f); // Yellow**  **glVertex2f(2.0f, -1.0f); // C**  **glVertex2f(0.0f, 2.0f); // A**  **glEnd();**  **//rectangle**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.0f, 0.0f); // red**  **glVertex2f(0.68f, 0.98f); // L**  **glVertex2f(4.0f, 3.0f); // J**  **glVertex2f(4.0f, 3.0f); // J**  **glVertex2f(0.0f, 5.0f); // I**  **glVertex2f(0.0f, 5.0f); // I**  **glVertex2f(-4.0f, 3.0f); // H**  **glVertex2f(-4.0f, 3.0f); // H**  **glVertex2f(-0.68f, 0.98f); // K**  **glVertex2f(-0.68f, 0.98f); // K**  **glVertex2f(0.68f, 0.98f); // L**  **glEnd();**  **//Line**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.5f, 0.0f); // Orange**  **glVertex2f(-3.0f, 0.0f); // x, y**  **glVertex2f(3.0f, 5.0f); // x, y**  **glVertex2f(3.0f, 0.0f); // x, y**  **glVertex2f(-3.0f, 5.0f); // x, y**  **glEnd();**  **//square**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3f(0.196078f, 0.6f, 0.8f); // sky blue**  **glVertex2f(-1.8f, 3.5f); // F**  **glVertex2f(-1.8f, 2.0f); // D**  **glVertex2f(-1.8f, 2.0f); // D**  **glVertex2f(1.8f, 2.0f); // E**  **glVertex2f(1.8f, 2.0f); // E**  **glVertex2f(1.8f, 3.5f); // G**  **glVertex2f(1.8f, 3.5f); // G**  **glVertex2f(-1.8f, 3.5f); // F**  **glEnd();**  **//triangle**  **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 1.0f, 0.0f); // Yellow**  **glVertex2f(0.0f, 2.0f); // A**  **glVertex2f(-2.0f, -1.0f); // B**  **glVertex2f(-2.0f, -1.0f); // B**  **glVertex2f(2.0f, -1.0f); // C**  **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**  **glutReshapeWindow (720,720);**  **glutDisplayFunc(task6); // Register display callback handler for window re-paint**  **gluOrtho2D(-10,10,-10,10);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |