**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-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**  **void code()**  **{**  **glPointSize(10.0);**  **glLineWidth(3.5);**  **glPointSize(5.0);**  **glBegin(GL\_LINES);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-2.0f, -1.0f);**  **glVertex2f(2.0f, -1.0f);**  **glVertex2f(2.0f, -1.0f);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(-2.0f, 2.0f);**  **glVertex2f(-2.0f, 2.0f);**  **glVertex2f(-2.0f, -1.0f);**  **}**  **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)**  **code();**  **glEnd();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("LAB TASK"); // Create a window with the given title**  **glutInitWindowSize(800, 800); // Set the window's initial width & height**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **gluOrtho2D(-5,5,-5,5);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-2**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code:**  **#include <windows.h>**  **#include <GL/glut.h>**  **void code()**  **{**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-3.0f, -1.0f); // x, y**  **glVertex2f(3.0f, -1.0f);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(-2.0f, 2.0f);**  **}**  **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)**  **code();**  **glEnd();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("LAB TASK"); // Create a window with the given title**  **glutInitWindowSize(800, 800); // Set the window's initial width & height**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **gluOrtho2D(-8,8,-8,9);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| Octagon Shape | Area & Angles - Video & Lesson Transcript | Study.com**Question-3**  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 code()**  **{**  **glPointSize(5.0);**    **glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-1.0f, -3.0f); // x, y**  **glVertex2f(1.0f, -3.0f);**  **glVertex2f(3.0f, -1.0f);**  **glVertex2f(3.0f, 1.0f);**  **glVertex2f(1.0f, 3.0f);**  **glVertex2f(-1.0f, 3.0f);**  **glVertex2f(-3.0f, 1.0f);**  **glVertex2f(-3.0f, -1.0f);**  **}**  **/\* 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);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **code();**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitWindowSize(500,500);**  **glutInitWindowPosition(10,10);**  **glutCreateWindow("LAB TASK");**  **glutDisplayFunc(display);**  **gluOrtho2D(-8,8,-8,9);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **void code()**  **{**  **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(0.0f, -1.0f);**  **glVertex2f(2.0f, -2.0f);**  **glVertex2f(1.0f, 0.0f);**  **glVertex2f(3.0f, 1.0f);**  **glVertex2f(0.62f, 1.0f);**  **glVertex2f(0.0f, 3.0f);**  **glVertex2f(-0.610f, 1.0f);**  **glVertex2f(-3.0f, 1.0f);**  **glVertex2f(-1.178f, 0.002f);**  **glVertex2f(-2.0f, -2.0f);**  **}**  **/\* 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);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **code();**  **glEnd();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitWindowSize(500,500);**  **glutInitWindowPosition(10,10);**  **glutCreateWindow("LAB TASK");**  **glutDisplayFunc(display);**  **gluOrtho2D(-8,8,-8,9);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-5**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **void Axis()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_LINES);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-6.0f, 0.0f);**  **glVertex2f(6.0f, 0.0f);**  **glVertex2f(0.0f, 6.0f);**  **glVertex2f(0.0f, -6.0f);**  **}**  **void Square()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(-5.0f, 1.0f); // x, y**  **glVertex2f(-1.0f, 1.0f);**  **glVertex2f(-1.0f, 5.0f);**  **glVertex2f(-5.0f, 5.0f);**  **}**  **void Triangle\_Left()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(0.5f, 0.0f, 0.8f);**  **glVertex2f(-4.0f, -3.0f); // x, y**  **glVertex2f(-1.0f, -5.0f);**  **glVertex2f(-1.0f, -1.0f);**  **}**  **void Triangle\_Right()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(1.0f, -4.0f); // x, y**  **glVertex2f(5.0f, -4.0f);**  **glVertex2f(3.0f, -1.0f);**  **}**  **void Arrow()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 1.0f, 0.0f);**  **glVertex2f(1.0f, 3.0f);**  **glVertex2f(1.0f, 2.0f);**  **glVertex2f(3.0f, 2.0f);**  **glVertex2f(3.0f, 3.0f);**  **glEnd();**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 1.0f, 0.0f);**  **glVertex2f(4.42f, 2.58f);**  **glVertex2f(3.0f, 4.0f);**  **glVertex2f(2.96f, 0.96f);**  **}**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **Axis();**  **glEnd();**  **Square();**  **glEnd();**  **Triangle\_Left();**  **glEnd();**  **Triangle\_Right();**  **glEnd();**  **Arrow();**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitWindowSize(700,700);**  **glutInitWindowPosition(10,10);**  **glutCreateWindow("LAB TASK");**  **glutDisplayFunc(display); // 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)-** |

|  |
| --- |
| **Question-6**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **void Rhombus()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2f(-4.0f, 0.0f); // x, y**  **glVertex2f(0.0f, -2.0f);**  **glVertex2f(4.0f, 0.0f);**  **glVertex2f(0.0f, 2.0f);**  **}**  **void Axis()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_LINES);**  **glColor3f(1.0f, 0.5f, 0.0f);//Orange**  **glVertex2f(-2.0f, 2.0f);**  **glVertex2f(3.0f, -3.0f);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(-3.0f, -3.0f);**  **}**  **void Triangle()**  **{**  **glLineWidth(5.5);**  **glPointSize(5.0);**  **glBegin(GL\_POLYGON);**  **glColor4f(1.0f, 1.0f, 0.0f, 0.0f);//yellow**  **glVertex2f(-3.0f, -4.0f); // x, y**  **glVertex2f(3.0f, -4.0f);**  **glVertex2f(0.0f, -0.77f);**  **}**  **void square()**  **{**  **glLineWidth(4.0);**  **glPointSize(5.0);**  **glBegin(GL\_LINES);**  **glColor3ub(30, 144 ,255);**  **glVertex2f(-2.0f, 0.77f);**  **glVertex2f(2.0f, 0.77f);**  **glVertex2f(2.0f, 0.77f);**  **glVertex2f(2.0f, -0.77f);**  **glVertex2f(2.0f, -0.77f);**  **glVertex2f(-2.0f, -0.77f);**  **glVertex2f(-2.0f, -0.77f);**  **glVertex2f(-2.0f, 0.77f);**  **}**  **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)**  **Rhombus();**  **glEnd();**  **Axis();**  **glEnd();**  **Triangle();**  **glEnd();**  **square();**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitWindowSize(700,700);**  **glutInitWindowPosition(10,10);**  **glutCreateWindow("LAB TASK");**  **glutDisplayFunc(display); // 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)-** |