**Lab Taks-1**

Submission Guidelines-

* Rename the file with your serial number only
* 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>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void rectangle(){**  **glBegin(GL\_LINE\_LOOP); // Each set of 4 vertices form a quad**  **glColor3f(0.0f, 0.0f, 0.0f); // Red**  **glVertex2d(-2, 2);**  **glVertex2d(2, 2);**  **glVertex2d(2, -1);**  **glVertex2d(-2, -1);**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1.5);**  **rectangle();**  **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(720, 720);**  **glutCreateWindow("Rectangle"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-5,5,-5,5);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void object(){**  **glBegin(GL\_QUADS); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2d(-1, 2);**  **glVertex2d(1, 2);**  **glVertex2d(2, -1);**  **glVertex2d(-2, -1);**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1.5);**  **object();**  **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(720, 720);**  **glutCreateWindow("Object"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-5,5,-5,5);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code -**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void Square(){**  **glBegin(GL\_QUADS); // Each set of 4 vertices form a quad**  **glColor3f(1.0f, 0.0f, 0.0f); // Red**  **glVertex2d(-3, 3);**  **glVertex2d(-1, 3);**  **glVertex2d(-1, 1);**  **glVertex2d(-3, 1);**  **glEnd();**  **}**  **void Triangle1(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(113, 55, 197);**  **glVertex2d(-1, -1);**  **glVertex2d(-1, -3);**  **glVertex2d(-3, -2);**  **glEnd();**  **}**  **void Triangle2(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 0);**  **glVertex2d(3, -1);**  **glVertex2d(5, -3);**  **glVertex2d(1, -3);**  **glEnd();**  **}**  **void Direction(){**  **glBegin(GL\_POLYGON);**  **glColor3ub( 68, 176, 36);**  **glVertex2d(1,2.48);**  **glVertex2d(3,2.48);**  **glVertex2d(3,1.48);**  **glVertex2d(1,1.48);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub( 68, 176, 36);**  **glVertex2d(3,3);**  **glVertex2d(5,2);**  **glVertex2d(3,1);**  **glEnd();**  **}**  **void Line(){**  **glBegin(GL\_LINES);**  **glColor3ub( 0,0,0);**  **glVertex2d(0,4.9);**  **glVertex2d(0,-4.9);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub( 0,0,0);**  **glVertex2d(-4.9,0);**  **glVertex2d(4.9,0);**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1.5);**  **Square();**  **Triangle1();**  **Triangle2();**  **Direction();**  **Line();**  **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(720, 720);**  **glutCreateWindow("4 shapes"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-5,5,-5,5);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |