**Lab Taks-3**

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 time that will be discussed in class VUES to the section named Lab Tak-3
* Must include resources for all the section in the table

|  |
| --- |
| **Question- 1**  Draw five storied building with windows and a front door |
| **Graph Plot (Picture)-**  **task3(a)** |
| **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(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(7.0);**  **// Draw a Red 1x1 Square centered at origin**  **glBegin(GL\_QUADS);**  **glColor3ub(255,140,0);**  **glVertex2d(70,90);**  **glVertex2d(-70,90);**  **glVertex2d(-70,-90);**  **glVertex2d(70,-90);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-70,60);**  **glVertex2d(70,60);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-70,30);**  **glVertex2d(70,30);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-70,0);**  **glVertex2d(70,0);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-70,-30);**  **glVertex2d(70,-30);**  **glEnd();**  **//windows corner**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(60,85);**  **glVertex2d(40,85);**  **glVertex2d(40,65);**  **glVertex2d(60,65);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(60,55);**  **glVertex2d(40,55);**  **glVertex2d(40,35);**  **glVertex2d(60,35);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(60,25);**  **glVertex2d(40,25);**  **glVertex2d(40,5);**  **glVertex2d(60,5);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(60,-5);**  **glVertex2d(40,-5);**  **glVertex2d(40,-25);**  **glVertex2d(60,-25);**  **glEnd();**  **//middle windows**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(20,85);**  **glVertex2d(0,85);**  **glVertex2d(0,65);**  **glVertex2d(20,65);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(20,55);**  **glVertex2d(0,55);**  **glVertex2d(0,35);**  **glVertex2d(20,35);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(20,25);**  **glVertex2d(0,25);**  **glVertex2d(0,5);**  **glVertex2d(20,5);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(20,-5);**  **glVertex2d(0,-5);**  **glVertex2d(0,-25);**  **glVertex2d(20,-25);**  **glEnd();**  **//Front border**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-30,90);**  **glVertex2d(-70,90);**  **glVertex2d(-70,-90);**  **glVertex2d(-30,-90);**  **glEnd();**  **//Door**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-10,-90);**  **glVertex2d(-10,-40);**  **glVertex2d(50,-40);**  **glVertex2d(50,-90);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(255,255,0);**  **glVertex2d(10,-90);**  **glVertex2d(10,-50);**  **glVertex2d(40,-50);**  **glVertex2d(40,-90);**  **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**  **glutInitWindowSize(520, 520); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **gluOrtho2D(-100,100,-100,100);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **Task3(a)** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-task3(b)** |
| **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(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(7.0);**  **//Making a tree**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(70, 117, 1);**  **glVertex2d(0,50);**  **glVertex2d(-35,0);**  **glVertex2d(35,0);**  **glEnd();**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(70, 117, 1);**  **glVertex2d(0,30);**  **glVertex2d(-35,-15);**  **glVertex2d(35,-15);**  **glEnd();**  **//giving root for tree**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(-5,-15);**  **glVertex2d(-5,-60);**  **glVertex2d(5,-60);**  **glVertex2d(5,-15);**  **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**  **glutInitWindowSize(520, 520); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **gluOrtho2D(-100,100,-100,100);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  task3(b) |

|  |
| --- |
| **Question- 3**  Draw a lamppost with black background |
| **Graph Plot (Picture)-**  **task3(c)** |
| **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(0.0f, 0.0f, 0.0f, 0.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(7.0);**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(-10,-20);**  **glVertex2d(-15,-25);**  **glVertex2d(15,-25);**  **glVertex2d(10,-20);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(-15,-25);**  **glVertex2d(-15,-55);**  **glVertex2d(15,-55);**  **glVertex2d(15,-25);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(-20,-55);**  **glVertex2d(-20,-65);**  **glVertex2d(20,-65);**  **glVertex2d(20,-55);**  **glEnd();**  **//middle of the lamp post**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(-4,-20);**  **glVertex2d(4,-20);**  **glVertex2d(4,50);**  **glVertex2d(-4,50);**  **glEnd();**  **//Head of the lamp post**  **glBegin(GL\_POLYGON);**  **glColor3ub(255,255,0);**  **glVertex2d(-4,50);**  **glVertex2d(4,50);**  **glVertex2d(20,75);**  **glVertex2d(0,95);**  **glVertex2d(-20,75);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(0,0,0);**  **glVertex2d(-20,75);**  **glVertex2d(20,75);**  **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**  **glutInitWindowSize(520, 520); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **gluOrtho2D(-100,100,-100,100);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **task3(c)** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-**  task3(d) |
| **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(0.0f, 0.0f, 0.0f, 0.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(7.0);**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(10,80);**  **glVertex2d(10,75);**  **glVertex2d(80,75);**  **glVertex2d(80,80);**  **glEnd();**  **//straight quads**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(15,85);**  **glVertex2d(15,50);**  **glVertex2d(20,50);**  **glVertex2d(20,85);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(70,85);**  **glVertex2d(70,50);**  **glVertex2d(75,50);**  **glVertex2d(75,85);**  **glEnd();**  **//Seat**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(10,50);**  **glVertex2d(5,40);**  **glVertex2d(85,40);**  **glVertex2d(80,50);**  **glEnd();**  **//Front Legs**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(10,40);**  **glVertex2d(10,10);**  **glVertex2d(15,10);**  **glVertex2d(15,40);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(80,40);**  **glVertex2d(75,40);**  **glVertex2d(75,10);**  **glVertex2d(80,10);**  **glEnd();**  **//Back legs**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(20,40);**  **glVertex2d(20,20);**  **glVertex2d(25,20);**  **glVertex2d(25,40);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(65,40);**  **glVertex2d(65,20);**  **glVertex2d(70,20);**  **glVertex2d(70,40);**  **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**  **glutInitWindowSize(550,550); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **gluOrtho2D(-100,100,-100,100);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **task3(d)** |

|  |
| --- |
| **Question- 5**  Use the building, tree, lamppost and bench to create a scenario |
| **Graph Plot (Picture)-**  task3(e) |
| **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(0.0f, 0.0f, 0.0f, 0.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(7.0);**  **//Giving a green background in the lower area**  **glBegin(GL\_QUADS);**  **glColor3ub(0,128,0);**  **glVertex2d(-200,-90);**  **glVertex2d(-200,-200);**  **glVertex2d(200,-200);**  **glVertex2d(200,-90);**  **glEnd();**  **//Lower corner green background**  **glBegin(GL\_QUADS);**  **glColor3ub(0,128,0);**  **glVertex2d(0,10);**  **glVertex2d(0,-200);**  **glVertex2d(200,-200);**  **glVertex2d(200,10);**  **glEnd();**  **//Bench**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(10,80);**  **glVertex2d(10,75);**  **glVertex2d(80,75);**  **glVertex2d(80,80);**  **glEnd();**  **//straight quads**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(15,85);**  **glVertex2d(15,50);**  **glVertex2d(20,50);**  **glVertex2d(20,85);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(70,85);**  **glVertex2d(70,50);**  **glVertex2d(75,50);**  **glVertex2d(75,85);**  **glEnd();**  **//Seat**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(10,50);**  **glVertex2d(5,40);**  **glVertex2d(85,40);**  **glVertex2d(80,50);**  **glEnd();**  **//Front Legs**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(10,40);**  **glVertex2d(10,10);**  **glVertex2d(15,10);**  **glVertex2d(15,40);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(253, 54, 9);**  **glVertex2d(80,40);**  **glVertex2d(75,40);**  **glVertex2d(75,10);**  **glVertex2d(80,10);**  **glEnd();**  **//House**  **glBegin(GL\_QUADS);**  **glColor3ub(255,140,0);**  **glVertex2d(-30,90);**  **glVertex2d(-190,90);**  **glVertex2d(-190,-90);**  **glVertex2d(-30,-90);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-190,60);**  **glVertex2d(-30,60);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-190,30);**  **glVertex2d(-30,30);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-190,0);**  **glVertex2d(-30,0);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3ub(221, 221, 13);**  **glVertex2d(-190,-30);**  **glVertex2d(-30,-30);**  **glEnd();**  **//windows corner**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-180,85);**  **glVertex2d(-160,85);**  **glVertex2d(-160,65);**  **glVertex2d(-180,65);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-180,55);**  **glVertex2d(-160,55);**  **glVertex2d(-160,35);**  **glVertex2d(-180,35);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-180,25);**  **glVertex2d(-160,25);**  **glVertex2d(-160,5);**  **glVertex2d(-180,5);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-180,-5);**  **glVertex2d(-160,-5);**  **glVertex2d(-160,-25);**  **glVertex2d(-180,-25);**  **glEnd();**  **//middle windows**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-140,85);**  **glVertex2d(-120,85);**  **glVertex2d(-120,65);**  **glVertex2d(-140,65);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-140,55);**  **glVertex2d(-120,55);**  **glVertex2d(-120,35);**  **glVertex2d(-140,35);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-140,25);**  **glVertex2d(-120,25);**  **glVertex2d(-120,5);**  **glVertex2d(-140,5);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-140,-5);**  **glVertex2d(-120,-5);**  **glVertex2d(-120,-25);**  **glVertex2d(-140,-25);**  **glEnd();**  **//Back border**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-30,90);**  **glVertex2d(-70,90);**  **glVertex2d(-70,-90);**  **glVertex2d(-30,-90);**  **glEnd();**  **//Door**  **glBegin(GL\_QUADS);**  **glColor3ub(149, 43, 3);**  **glVertex2d(-130,-90);**  **glVertex2d(-130,-40);**  **glVertex2d(-180,-40);**  **glVertex2d(-180,-90);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(255,255,0);**  **glVertex2d(-140,-90);**  **glVertex2d(-140,-50);**  **glVertex2d(-170,-50);**  **glVertex2d(-170,-90);**  **glEnd();**  **//Making a tree**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(70, 117, 1);**  **glVertex2d(175,150);**  **glVertex2d(140,115);**  **glVertex2d(200,115);**  **glEnd();**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(70, 117, 1);**  **glVertex2d(180,130);**  **glVertex2d(135,100);**  **glVertex2d(200,100);**  **glEnd();**  **//giving root for tree**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(180,100);**  **glVertex2d(180,10);**  **glVertex2d(160,10);**  **glVertex2d(160,100);**  **glEnd();**  **//Making lamp post**  **//lower body**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(104,30);**  **glVertex2d(104,10);**  **glVertex2d(116,10);**  **glVertex2d(116,30);**  **glEnd();**  **//middle body**  **glBegin(GL\_QUADS);**  **glColor3ub(147, 50, 3);**  **glVertex2d(108,90);**  **glVertex2d(108,30);**  **glVertex2d(111,30);**  **glVertex2d(111,90);**  **glEnd();**  **// lamp post light**  **glBegin(GL\_QUADS);**  **glColor3ub(255,255,0);**  **glVertex2d(95,90);**  **glVertex2d(125,90);**  **glVertex2d(115,110);**  **glVertex2d(105,110);**  **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**  **glutInitWindowSize(550,550); // Set the window's initial width & height**  **glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title**  **gluOrtho2D(-200,200,-200,200);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **task3(e)** |