**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)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **void building()**  **{**  **//Building**  **glBegin(GL\_QUADS);**  **glColor3f(0, 0, 255); //orange**  **glVertex2f(-0.5f, -0.7f);**  **glVertex2f(0.5f, -0.7f);**  **glVertex2f(0.5f, 0.7f);**  **glVertex2f(-0.5f, 0.7f);**  **glEnd();**  **//Windows**  **//5**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, 0.5f);**  **glVertex2f(-0.1f, 0.5f);**  **glVertex2f(-0.1f, 0.6f);**  **glVertex2f(-0.2f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, 0.5f);**  **glVertex2f(-0.3f, 0.5f);**  **glVertex2f(-0.3f, 0.6f);**  **glVertex2f(-0.4f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, 0.5f);**  **glVertex2f(0.3f, 0.5f);**  **glVertex2f(0.3f, 0.6f);**  **glVertex2f(0.4f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, 0.5f);**  **glVertex2f(0.1f, 0.5f);**  **glVertex2f(0.1f, 0.6f);**  **glVertex2f(0.2f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, 0.1f);**  **glVertex2f(-0.1f, 0.1f);**  **glVertex2f(-0.1f, 0.2f);**  **glVertex2f(-0.2f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, 0.1f);**  **glVertex2f(-0.3f, 0.1f);**  **glVertex2f(-0.3f, 0.2f);**  **glVertex2f(-0.4f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, 0.3f);**  **glVertex2f(-0.3f, 0.3f);**  **glVertex2f(-0.3f, 0.4f);**  **glVertex2f(-0.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, 0.3f);**  **glVertex2f(-0.1f, 0.3f);**  **glVertex2f(-0.1f, 0.4f);**  **glVertex2f(-0.2f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, 0.1f);**  **glVertex2f(0.1f, 0.1f);**  **glVertex2f(0.1f, 0.2f);**  **glVertex2f(0.2f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, 0.1f);**  **glVertex2f(0.3f, 0.1f);**  **glVertex2f(0.3f, 0.2f);**  **glVertex2f(0.4f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, 0.3f);**  **glVertex2f(0.3f, 0.3f);**  **glVertex2f(0.3f, 0.4f);**  **glVertex2f(0.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, 0.3f);**  **glVertex2f(0.1f, 0.3f);**  **glVertex2f(0.1f, 0.4f);**  **glVertex2f(0.2f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, -0.1f);**  **glVertex2f(-0.1f, -0.1f);**  **glVertex2f(-0.1f, -0.2f);**  **glVertex2f(-0.2f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, -0.1f);**  **glVertex2f(-0.3f, -0.1f);**  **glVertex2f(-0.3f, -0.2f);**  **glVertex2f(-0.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, -0.3f);**  **glVertex2f(-0.3f, -0.3f);**  **glVertex2f(-0.3f, -0.4f);**  **glVertex2f(-0.4f, -0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, -0.3f);**  **glVertex2f(-0.1f, -0.3f);**  **glVertex2f(-0.1f, -0.4f);**  **glVertex2f(-0.2f, -0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, -0.1f);**  **glVertex2f(0.1f, -0.1f);**  **glVertex2f(0.1f, -0.2f);**  **glVertex2f(0.2f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, -0.1f);**  **glVertex2f(0.3f, -0.1f);**  **glVertex2f(0.3f, -0.2f);**  **glVertex2f(0.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, -0.3f);**  **glVertex2f(0.3f, -0.3f);**  **glVertex2f(0.3f, -0.4f);**  **glVertex2f(0.4f, -0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, -0.3f);**  **glVertex2f(0.1f, -0.3f);**  **glVertex2f(0.1f, -0.4f);**  **glVertex2f(0.2f, -0.4f);**  **glEnd();**  **//Door**  **glBegin(GL\_QUADS);**  **glColor3f(1.f, 1.0f, 1.0f);**  **glVertex2f(-0.2f, -0.695f);**  **glVertex2f(0.2f, -0.695f);**  **glVertex2f(0.2f, -0.5f);**  **glVertex2f(-0.2f, -0.5f);**  **glEnd();**  **glFlush();**  **}**  **void display() {**  **glClearColor(224, 224, 224, 1.0f); // Set background color to white**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **building();**  **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);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **void tree()**  **{**  **//Tree**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 255, 0); //green**  **glVertex2f(0.7f, 0.0f);**  **glVertex2f(0.8f, 0.3f);**  **glVertex2f(0.9f, 0.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 255, 0); //green**  **glVertex2f(0.7f, -0.1f);**  **glVertex2f(0.8f, 0.1f);**  **glVertex2f(0.9f, -0.1f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 255, 0); //green**  **glVertex2f(0.7f, -0.2f);**  **glVertex2f(0.8f, 0.0f);**  **glVertex2f(0.9f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 0, 0); //black**  **glVertex2f(0.79f, -0.7f);**  **glVertex2f(0.83f, -0.7f);**  **glVertex2f(0.83f, -0.2f);**  **glVertex2f(0.79f, -0.2f);**  **glEnd();**  **glFlush();**  **}**  **void display() {**  **glClearColor(255, 255, 255, 1.0f); // Set background color to white**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **tree();**  **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);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 3**  Draw a lamppost with black background |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **void lamppost()**  **{**  **//lamppost**  **glLineWidth(8);**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3f(0, 0, 0); // black**  **glVertex2f(0.7f, -0.28f);**  **glVertex2f(0.72f, -0.28f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 0, 0); //black**  **glVertex2f(0.66f, -0.7f);**  **glVertex2f(0.7f, -0.7f);**  **glVertex2f(0.7f, -0.25f);**  **glVertex2f(0.66f, -0.25f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0); //yellow**  **glVertex2f(0.7f, -0.3f);**  **glVertex2f(0.72f, -0.3f);**  **glVertex2f(0.72f, -0.35f);**  **glVertex2f(0.7f, -0.35f);**  **glEnd();**  **glFlush();**  **}**  **void display() {**  **glClearColor(224, 224, 224, 1.0f); // Set background color to white**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **lamppost();**  **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);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **void bench()**  **{**  **glLineWidth(5);**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.63f, -0.6f);**  **glVertex2f(-0.63f, -0.7f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.65f, -0.5f);**  **glVertex2f(-0.65f, -0.68f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.81f, -0.6f);**  **glVertex2f(-0.81f, -0.7f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.85f, -0.5f);**  **glVertex2f(-0.85f, -0.68f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.62f, -0.6f);**  **glVertex2f(-0.65f, -0.55f);**  **glVertex2f(-0.85f, -0.55f);**  **glVertex2f(-0.82f, -0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(0, 200, 200);**  **glVertex2f(-0.85f, -0.5f);**  **glVertex2f(-0.65f, -0.5f);**  **glVertex2f(-0.65f, -0.55f);**  **glVertex2f(-0.85f, -0.55f);**  **glEnd();**  **glFlush();**  **}**  **void display() {**  **glClearColor(224, 224, 224, 1.0f); // Set background color to white**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **bench();**  **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);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 5**  Use the building, tree, lamppost and bench to create a scenario |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **void building()**  **{**  **//Building**  **glBegin(GL\_QUADS);**  **glColor3f(0, 0, 255); //orange**  **glVertex2f(-0.5f, -0.7f);**  **glVertex2f(0.5f, -0.7f);**  **glVertex2f(0.5f, 0.7f);**  **glVertex2f(-0.5f, 0.7f);**  **glEnd();**  **//Windows**  **//5**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, 0.5f);**  **glVertex2f(-0.1f, 0.5f);**  **glVertex2f(-0.1f, 0.6f);**  **glVertex2f(-0.2f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, 0.5f);**  **glVertex2f(-0.3f, 0.5f);**  **glVertex2f(-0.3f, 0.6f);**  **glVertex2f(-0.4f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, 0.5f);**  **glVertex2f(0.3f, 0.5f);**  **glVertex2f(0.3f, 0.6f);**  **glVertex2f(0.4f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, 0.5f);**  **glVertex2f(0.1f, 0.5f);**  **glVertex2f(0.1f, 0.6f);**  **glVertex2f(0.2f, 0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, 0.1f);**  **glVertex2f(-0.1f, 0.1f);**  **glVertex2f(-0.1f, 0.2f);**  **glVertex2f(-0.2f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, 0.1f);**  **glVertex2f(-0.3f, 0.1f);**  **glVertex2f(-0.3f, 0.2f);**  **glVertex2f(-0.4f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, 0.3f);**  **glVertex2f(-0.3f, 0.3f);**  **glVertex2f(-0.3f, 0.4f);**  **glVertex2f(-0.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, 0.3f);**  **glVertex2f(-0.1f, 0.3f);**  **glVertex2f(-0.1f, 0.4f);**  **glVertex2f(-0.2f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, 0.1f);**  **glVertex2f(0.1f, 0.1f);**  **glVertex2f(0.1f, 0.2f);**  **glVertex2f(0.2f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, 0.1f);**  **glVertex2f(0.3f, 0.1f);**  **glVertex2f(0.3f, 0.2f);**  **glVertex2f(0.4f, 0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, 0.3f);**  **glVertex2f(0.3f, 0.3f);**  **glVertex2f(0.3f, 0.4f);**  **glVertex2f(0.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, 0.3f);**  **glVertex2f(0.1f, 0.3f);**  **glVertex2f(0.1f, 0.4f);**  **glVertex2f(0.2f, 0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, -0.1f);**  **glVertex2f(-0.1f, -0.1f);**  **glVertex2f(-0.1f, -0.2f);**  **glVertex2f(-0.2f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, -0.1f);**  **glVertex2f(-0.3f, -0.1f);**  **glVertex2f(-0.3f, -0.2f);**  **glVertex2f(-0.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.4f, -0.3f);**  **glVertex2f(-0.3f, -0.3f);**  **glVertex2f(-0.3f, -0.4f);**  **glVertex2f(-0.4f, -0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(-0.2f, -0.3f);**  **glVertex2f(-0.1f, -0.3f);**  **glVertex2f(-0.1f, -0.4f);**  **glVertex2f(-0.2f, -0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, -0.1f);**  **glVertex2f(0.1f, -0.1f);**  **glVertex2f(0.1f, -0.2f);**  **glVertex2f(0.2f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, -0.1f);**  **glVertex2f(0.3f, -0.1f);**  **glVertex2f(0.3f, -0.2f);**  **glVertex2f(0.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.4f, -0.3f);**  **glVertex2f(0.3f, -0.3f);**  **glVertex2f(0.3f, -0.4f);**  **glVertex2f(0.4f, -0.4f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(255,255,0);**  **glVertex2f(0.2f, -0.3f);**  **glVertex2f(0.1f, -0.3f);**  **glVertex2f(0.1f, -0.4f);**  **glVertex2f(0.2f, -0.4f);**  **glEnd();**  **//Door**  **glBegin(GL\_QUADS);**  **glColor3f(1.f, 1.0f, 1.0f);**  **glVertex2f(-0.2f, -0.695f);**  **glVertex2f(0.2f, -0.695f);**  **glVertex2f(0.2f, -0.5f);**  **glVertex2f(-0.2f, -0.5f);**  **glEnd();**  **glFlush();**  **}**  **void tree()**  **{**  **//Tree**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 255, 0); //green**  **glVertex2f(0.7f, 0.0f);**  **glVertex2f(0.8f, 0.3f);**  **glVertex2f(0.9f, 0.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 255, 0); //green**  **glVertex2f(0.7f, -0.1f);**  **glVertex2f(0.8f, 0.1f);**  **glVertex2f(0.9f, -0.1f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 255, 0); //green**  **glVertex2f(0.7f, -0.2f);**  **glVertex2f(0.8f, 0.0f);**  **glVertex2f(0.9f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 0, 0); //black**  **glVertex2f(0.79f, -0.7f);**  **glVertex2f(0.83f, -0.7f);**  **glVertex2f(0.83f, -0.2f);**  **glVertex2f(0.79f, -0.2f);**  **glEnd();**  **glFlush();**  **}**  **void lamppost()**  **{**  **//lamppost**  **glLineWidth(8);**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3f(0, 0, 0); // black**  **glVertex2f(0.7f, -0.28f);**  **glVertex2f(0.72f, -0.28f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 0, 0); //black**  **glVertex2f(0.66f, -0.7f);**  **glVertex2f(0.7f, -0.7f);**  **glVertex2f(0.7f, -0.25f);**  **glVertex2f(0.66f, -0.25f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0); //yellow**  **glVertex2f(0.7f, -0.3f);**  **glVertex2f(0.72f, -0.3f);**  **glVertex2f(0.72f, -0.35f);**  **glVertex2f(0.7f, -0.35f);**  **glEnd();**  **glFlush();**  **}**  **void bench()**  **{**  **glLineWidth(5);**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.63f, -0.6f);**  **glVertex2f(-0.63f, -0.7f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.65f, -0.5f);**  **glVertex2f(-0.65f, -0.68f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.81f, -0.6f);**  **glVertex2f(-0.81f, -0.7f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.85f, -0.5f);**  **glVertex2f(-0.85f, -0.68f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0, 0, 0);**  **glVertex2f(-0.62f, -0.6f);**  **glVertex2f(-0.65f, -0.55f);**  **glVertex2f(-0.85f, -0.55f);**  **glVertex2f(-0.82f, -0.6f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3f(0, 200, 200);**  **glVertex2f(-0.85f, -0.5f);**  **glVertex2f(-0.65f, -0.5f);**  **glVertex2f(-0.65f, -0.55f);**  **glVertex2f(-0.85f, -0.55f);**  **glEnd();**  **glFlush();**  **}**  **void road()**  **{**  **//road**  **glBegin(GL\_POLYGON);**  **glColor3f(0 ,0 ,0); //black**  **glVertex2f(-0.999f, -0.7f);**  **glVertex2f(-0.999f, -0.9999f);**  **glVertex2f(0.999f, -0.9999f);**  **glVertex2f(0.999f, -0.7f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255 ,255 ,255); //white**  **glVertex2f(-0.7f, -0.9f);**  **glVertex2f(-0.2f, -0.9f);**  **glVertex2f(-0.2f, -0.85f);**  **glVertex2f(-0.7f, -0.85f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255 ,255 ,255); //white**  **glVertex2f(0.7f, -0.9f);**  **glVertex2f(0.2f, -0.9f);**  **glVertex2f(0.2f, -0.85f);**  **glVertex2f(0.7f, -0.85f);**  **glEnd();**  **glFlush();**  **}**  **void display() {**  **glClearColor(224, 224, 224, 1.0f); // Set background color to white**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **building();**  **tree();**  **lamppost();**  **bench();**  **road();**  **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);**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |