**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)-**  **A screenshot of a computer  Description automatically generated** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void Building(){  glBegin(GL\_QUADS);  glColor3f(0.4f, 0.2f, 0.1f);  // draw the building  glVertex2f(-0.8f, -0.7f);  glVertex2f(-0.0f, -0.7f);  glVertex2f(-0.0f, 0.8f);  glVertex2f(-0.8f, 0.8f);  glEnd();  // Lines  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, 0.5f);  glVertex2f(-0.0f, 0.5f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, 0.2f);  glVertex2f(-0.0f, 0.2f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, -0.1f);  glVertex2f(-0.0f, -0.1f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, -0.4f);  glVertex2f(-0.0f, -0.4f);  glEnd();  // Windows  glColor3f(0.8f, 0.8f, 0.8f);  glBegin(GL\_QUADS);  glVertex2f(-0.65f, 0.4f);  glVertex2f(-0.55f, 0.4f);  glVertex2f(-0.55f, 0.5f);  glVertex2f(-0.65f, 0.5f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.65f, 0.1f);  glVertex2f(-0.55f, 0.1f);  glVertex2f(-0.55f, 0.2f);  glVertex2f(-0.65f, 0.2f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.25f, 0.4f);  glVertex2f(-0.15f, 0.4f);  glVertex2f(-0.15f, 0.5f);  glVertex2f(-0.25f, 0.5f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.25f, 0.1f);  glVertex2f(-0.15f, 0.1f);  glVertex2f(-0.15f, 0.2f);  glVertex2f(-0.25f, 0.2f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.65f, -0.1f);  glVertex2f(-0.55f, -0.1f);  glVertex2f(-0.55f, -0.2f);  glVertex2f(-0.65f, -0.2f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.25f, -0.1f);  glVertex2f(-0.15f, -0.1f);  glVertex2f(-0.15f, -0.2f);  glVertex2f(-0.25f, -0.2f);  glEnd();  // Door  glColor3f(0.2f, 0.1f, 0.0f);  glBegin(GL\_QUADS);  glVertex2f(-0.45f, -0.7f);  glVertex2f(-0.35f, -0.7f);  glVertex2f(-0.35f, -0.5f);  glVertex2f(-0.45f, -0.5f);  glEnd();  }  void display() {  glClearColor(0.60f, 0.60f, 0.60f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);      Building();  glFlush();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("Building ");  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-**  **A screenshot of a computer  Description automatically generated** |
| **Code-**  void Tree(){  glBegin(GL\_QUADS);  glColor3ub(98.0, 83.0f, 83.0f);  glVertex2f(0.65f, 0.65f);  glVertex2f(0.65f, 0.0f);  glVertex2f(0.75f, 0.0f);  glVertex2f(0.75f, 0.65f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3f(00.0f, 01.0f, 0.0f);  glVertex2f(0.45f, 0.65f);  glVertex2f(0.95f, 0.65f);  glVertex2f(0.7f, 0.9f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3f(00.0f, 01.0f, 0.0f);  glVertex2f(0.45f, 0.5f);  glVertex2f(0.95f, 0.5f);  glVertex2f(0.7f, 0.75f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3f(00.0f, 01.0f, 0.0f);  glVertex2f(0.45f, 0.4f);  glVertex2f(0.95f, 0.4f);  glVertex2f(0.7f, 0.6f);  glEnd();  }  void display() {  glClearColor(0.60f, 0.60f, 0.60f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  Tree();    glFlush();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("Tree ");  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-**  A screenshot of a computer  Description automatically generated |

|  |
| --- |
| **Question- 3**  Draw a lamppost with black background |
| **Graph Plot (Picture)-**  **A graph with lines and dots** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void Lamp(){  glBegin(GL\_QUADS);  glColor3ub(220.0f, 220.0f, 8.0f);  glVertex2f(-0.60f, -0.6f);  glVertex2f(0.9f, -0.6f);  glVertex2f(0.6f, 0.5f);  glVertex2f(-0.3f, 0.5f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(0, 0, 139);  glVertex2f(-0.7f, -0.6f);  glVertex2f(-0.5f, -0.6f);  glVertex2f(-0.5f, 0.9f);  glVertex2f(-0.7f, 0.9f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(0, 0, 139);  glVertex2f(-0.5f, 0.6f);  glVertex2f(-0.15f, 0.6f);  glVertex2f(0.2f, 0.9f);  glVertex2f(-0.5f, 0.9f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3ub(142, 142, 142);  glVertex2f(-0.3f, 0.5f);  glVertex2f(0.6f, 0.5f);  glVertex2f(0.2f, 0.9f);  glEnd();  }  void display() {  glClearColor(0.00f, 0.0f, 0.0f, 0.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glLineWidth(5);  Lamp();  glFlush();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("Lamppost");  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-**  **A graph with lines and dots  Description automatically generated** |
| Code-  #include <windows.h>  #include <GL/glut.h>  void Bench(){  glBegin(GL\_LINES);  glColor3ub(100, 100, 100); // Dark gray color  glVertex2f(-0.4f, -0.2f); // Bottom left leg  glVertex2f(-0.4f, 0.5f); // Top left leg  glEnd();  glBegin(GL\_LINES);  glColor3ub(100, 100, 100); // Dark gray color  glVertex2f(0.4f, 0.5f); // Top right leg  glVertex2f(0.4f, -0.2f); // Bottom right leg  glEnd();  glBegin(GL\_LINES);  glColor3ub(100, 100, 100); // Dark gray color  glVertex2f(-0.4f, 0.1f);  glVertex2f(-0.6f, -0.1f);  glEnd();  glBegin(GL\_LINES);  glColor3ub(100, 100, 100); // Dark gray color  glVertex2f(0.4f, 0.1f);  glVertex2f(0.2f, -0.1f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(139, 69, 19); // Saddle Brown color  glVertex2f(-0.4f, 0.4f);  glVertex2f(0.4f, 0.4f);  glVertex2f(0.4f, 0.5f);  glVertex2f(-0.4f, 0.5f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(139, 69, 19); // Saddle Brown color  glVertex2f(-0.4f, 0.2f);  glVertex2f(0.4f, 0.2);  glVertex2f(0.4f, 0.3f);  glVertex2f(-0.4f, 0.3f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(160, 82, 45); // Sienna color  glVertex2f(-0.4f, 0.1f);  glVertex2f(-0.42f, 0.08f);  glVertex2f(0.38f, 0.08f);  glVertex2f(0.4f, 0.1f);  glVertex2f(0.4f, 0.13f);  glVertex2f(-0.4f, 0.13f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(139, 69, 19); // Saddle Brown color  glVertex2f(-0.5f, 0.0f);  glVertex2f(0.3f, 0.0f);  glVertex2f(0.35f, 0.05f);  glVertex2f(-0.45f, 0.05f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(139, 69, 19); // Saddle Brown color  glVertex2f(-0.6f, -0.1f);  glVertex2f(0.2f, -0.1f);  glVertex2f(0.25f, -0.05f);  glVertex2f(-0.55f, -0.05f);  glEnd();  glBegin(GL\_LINES);  glColor3ub(100, 100, 100); // Dark gray color  glVertex2f(-0.6f, -0.1f);  glVertex2f(-0.6f, -0.4f);  glEnd();  glBegin(GL\_LINES);  glColor3ub(100, 100, 100); // Dark gray color  glVertex2f(0.2f, -0.1f);  glVertex2f(0.2f, -0.4f);  glEnd();  }  void display() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black background  glClear(GL\_COLOR\_BUFFER\_BIT);  glLineWidth(5);  Bench();  glFlush();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("Bench ");  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-**  **A computer screen shot of a bench  Description automatically generated** |

|  |
| --- |
| **Question- 5**  Use the building, tree, lamppost and bench to create a scenario |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void Road(){  glBegin(GL\_QUADS);  glColor3f(0.0, 0.0f, 0.0f);  glVertex2f(-1.0f, -0.8f);  glVertex2f(-1.0f, -1.0f);  glVertex2f(1.0f, -1.0f);  glVertex2f(1.0f, -0.8f);  glEnd();  glBegin(GL\_QUADS);  glColor3f(0.0, 0.0f, 0.0f);  glVertex2f(0.2f, 1.0f);  glVertex2f(0.4f, 1.0f);  glVertex2f(0.4f, -0.8f);  glVertex2f(0.2f, -0.8f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3ub(173, 255, 47);  glVertex2f(-1.0f, -0.9f);  glVertex2f(1.0f, -0.9f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3ub(173, 255, 47);  glVertex2f(0.3f, -0.8f);  glVertex2f(0.3f, 1.0f);  glEnd();  }  void Building(){  glBegin(GL\_QUADS);  glColor3f(0.4f, 0.2f, 0.1f);  // draw the building  glVertex2f(-0.8f, -0.7f);  glVertex2f(-0.0f, -0.7f);  glVertex2f(-0.0f, 0.8f);  glVertex2f(-0.8f, 0.8f);  glEnd();  // Lines  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, 0.5f);  glVertex2f(-0.0f, 0.5f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, 0.2f);  glVertex2f(-0.0f, 0.2f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, -0.1f);  glVertex2f(-0.0f, -0.1f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.2f, 0.2f, 0.2f);  glVertex2f(-0.8f, -0.4f);  glVertex2f(-0.0f, -0.4f);  glEnd();  // Windows  glColor3f(0.8f, 0.8f, 0.8f);  glBegin(GL\_QUADS);  glVertex2f(-0.65f, 0.4f);  glVertex2f(-0.55f, 0.4f);  glVertex2f(-0.55f, 0.5f);  glVertex2f(-0.65f, 0.5f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.65f, 0.1f);  glVertex2f(-0.55f, 0.1f);  glVertex2f(-0.55f, 0.2f);  glVertex2f(-0.65f, 0.2f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.25f, 0.4f);  glVertex2f(-0.15f, 0.4f);  glVertex2f(-0.15f, 0.5f);  glVertex2f(-0.25f, 0.5f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.25f, 0.1f);  glVertex2f(-0.15f, 0.1f);  glVertex2f(-0.15f, 0.2f);  glVertex2f(-0.25f, 0.2f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.65f, -0.1f);  glVertex2f(-0.55f, -0.1f);  glVertex2f(-0.55f, -0.2f);  glVertex2f(-0.65f, -0.2f);  glEnd();  glBegin(GL\_QUADS);  glVertex2f(-0.25f, -0.1f);  glVertex2f(-0.15f, -0.1f);  glVertex2f(-0.15f, -0.2f);  glVertex2f(-0.25f, -0.2f);  glEnd();  // Door  glColor3f(0.2f, 0.1f, 0.0f);  glBegin(GL\_QUADS);  glVertex2f(-0.45f, -0.7f);  glVertex2f(-0.35f, -0.7f);  glVertex2f(-0.35f, -0.5f);  glVertex2f(-0.45f, -0.5f);  glEnd();  }  void Bench(){  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.78f,0.28f);  glVertex2f(0.78f,0.16f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.78f,0.22f);  glVertex2f(0.74f,0.18f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.74f,0.18f);  glVertex2f(0.74f,0.12f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.94f,0.28f);  glVertex2f(0.94f,0.16f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.94f,0.22f);  glVertex2f(0.9f,0.18f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.9f,0.18f);  glVertex2f(0.9f,0.12f);  glEnd();  glLineWidth(3);  glBegin(GL\_LINES);  glColor3f(0.0f,0.0f,0.0f);  glVertex2f(0.76f,0.2f);  glVertex2f(0.92f,0.2f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(111,11,22);  glVertex2f(0.78f,0.28f);  glVertex2f(0.78f,0.26f);  glVertex2f(0.94f,0.26f);  glVertex2f(0.94f,0.28f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(111,11,22);  glVertex2f(0.78f,0.24f);  glVertex2f(0.78f,0.22f);  glVertex2f(0.94f,0.22f);  glVertex2f(0.94f,0.24f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(111,11,22);  glVertex2f(0.78f,0.22f);  glVertex2f(0.76f,0.2f);  glVertex2f(0.92f,0.2f);  glVertex2f(0.94f,0.22f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(111,11,22);  glVertex2f(0.76f,0.196f);  glVertex2f(0.92f,0.196f);  glVertex2f(0.9f,0.18f);  glVertex2f(0.74f,0.18f);  glEnd();  }  void Tree(){  glBegin(GL\_QUADS);  glColor3ub(98.0, 83.0f, 83.0f);  glVertex2f(0.65f, 0.65f);  glVertex2f(0.65f, 0.2f);  glVertex2f(0.75f, 0.2f);  glVertex2f(0.75f, 0.65f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3f(00.0f, 01.0f, 0.0f);  glVertex2f(0.45f, 0.65f);  glVertex2f(0.95f, 0.65f);  glVertex2f(0.7f, 0.9f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3f(00.0f, 01.0f, 0.0f);  glVertex2f(0.45f, 0.5f);  glVertex2f(0.95f, 0.5f);  glVertex2f(0.7f, 0.75f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3f(00.0f, 01.0f, 0.0f);  glVertex2f(0.45f, 0.4f);  glVertex2f(0.95f, 0.4f);  glVertex2f(0.7f, 0.6f);  glEnd();  }  void Lamp(){  glBegin(GL\_QUADS);  glColor3ub(0, 0, 139);  glVertex2f(0.14f, -0.63f);  glVertex2f(0.14f, -0.8f);  glVertex2f(0.16f, -0.8f);  glVertex2f(0.16f, -0.63f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(0, 0, 139);  glVertex2f(0.16f, -0.63f);  glVertex2f(0.16f, -0.65f);  glVertex2f(0.19f, -0.65f);  glVertex2f(0.19f, -0.63f);  glEnd();  glBegin(GL\_TRIANGLES);  glColor3ub(0, 0, 139);  glVertex2f(0.1775f, -0.65f);  glVertex2f(0.165f, -0.675f);  glVertex2f(0.19f, -0.675f);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(173, 255, 47);  glVertex2f(0.1775f, -0.675f);  glVertex2f(0.16f, -0.72f);  glVertex2f(0.16f, -0.8f);  glVertex2f(0.24f, -0.8f);  glEnd();  }  void display() {  glClearColor(0.60f, 0.60f, 0.60f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  Road();  Building();  Tree();  Bench();  Lamp();  glFlush();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("Scenario”);  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-**  **A computer screen shot of a video game  Description automatically generated** |