**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**  **/\* 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);**  **//glClearColor(0.1f, 0.0f, 0.0f, 0.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(10.0);**  **// Draw a Red 1x1 Square centered at origin**  **//For the building frame structure**  **glColor3ub(0, 125, 250);**  **glBegin(GL\_QUADS);**  **glVertex2f(35.0f,20.0f);**  **glVertex2f(130.0f,20.0f);**  **glVertex2f(130.0f,125.0f);**  **glVertex2f(35.0f,125.0f);**  **glEnd();**  **//For 2D view**  **glColor3ub(125, 250, 0);**  **glBegin(GL\_QUADS);**  **glVertex2f(15.0f,20.0f);**  **glVertex2f(35.0f,20.0f);**  **glVertex2f(35.0f,125.0f);**  **glVertex2f(15.0f,105.0f);**  **glEnd();**  **//For the building door**  **glColor3ub(125, 0, 250);**  **glBegin(GL\_QUADS);**  **glVertex2f(75.0f,20.0f);**  **glVertex2f(85.0f,20.0f);**  **glVertex2f(85.0f,35.0f);**  **glVertex2f(75.0f,35.0f);**  **glEnd();**  **//LEFT SIDE WINDOW**  **//Window of first floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(45.0f,30.0f);**  **glVertex2f(65.0f,30.0f);**  **glVertex2f(65.0f,40.0f);**  **glVertex2f(45.0f,40.0f);**  **glEnd();**  **//for black line above the first floor window**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(35.0f,45.0f);**  **glVertex2f(130.0f,45.0f);**  **glEnd();**  **//Window of second floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(45.0f,50.0f);**  **glVertex2f(65.0f,50.0f);**  **glVertex2f(65.0f,60.0f);**  **glVertex2f(45.0f,60.0f);**  **glEnd();**  **//for black line above the second floor window**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(35.0f,65.0f);**  **glVertex2f(130.0f,65.0f);**  **glEnd();**  **//Window of third floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(45.0f,70.0f);**  **glVertex2f(65.0f,70.0f);**  **glVertex2f(65.0f,80.0f);**  **glVertex2f(45.0f,80.0f);**  **glEnd();**  **//for black line above the third floor window**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(35.0f,85.0f);**  **glVertex2f(130.0f,85.0f);**  **glEnd();**  **//Window of fourth floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(45.0f,90.0f);**  **glVertex2f(65.0f,90.0f);**  **glVertex2f(65.0f,100.0f);**  **glVertex2f(45.0f,100.0f);**  **glEnd();**  **//for black line above the fourth floor window**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(35.0f,105.0f);**  **glVertex2f(130.0f,105.0f);**  **glEnd();**  **//Window of fifth floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(45.0f,110.0f);**  **glVertex2f(65.0f,110.0f);**  **glVertex2f(65.0f,120.0f);**  **glVertex2f(45.0f,120.0f);**  **glEnd();**  **//for black line above the fifth floor window**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(35.0f,125.0f);**  **glVertex2f(130.0f,125.0f);**  **glEnd();**  **//RIGHT SIDE WINDOW**  **//Window of first floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(95.0f,30.0f);**  **glVertex2f(115.0f,30.0f);**  **glVertex2f(115.0f,40.0f);**  **glVertex2f(95.0f,40.0f);**  **glEnd();**  **//Window of second floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(95.0f,50.0f);**  **glVertex2f(115.0f,50.0f);**  **glVertex2f(115.0f,60.0f);**  **glVertex2f(95.0f,60.0f);**  **glEnd();**  **//Window of third floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(95.0f,70.0f);**  **glVertex2f(115.0f,70.0f);**  **glVertex2f(115.0f,80.0f);**  **glVertex2f(95.0f,80.0f);**  **glEnd();**  **//Window of fourth floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(95.0f,90.0f);**  **glVertex2f(115.0f,90.0f);**  **glVertex2f(115.0f,100.0f);**  **glVertex2f(95.0f,100.0f);**  **glEnd();**  **//Window of fifth floor**  **glColor3ub(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(95.0f,110.0f);**  **glVertex2f(115.0f,110.0f);**  **glVertex2f(115.0f,120.0f);**  **glVertex2f(95.0f,120.0f);**  **glEnd();**  **//for black line under ground**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(15.0f,15.0f);**  **glVertex2f(130.0f,15.0f);**  **glVertex2f(130.0f,20.0f);**  **glVertex2f(15.0f,20.0f);**  **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**  **glutCreateWindow("Five Storied Building"); // Create a window with the given title**  **glutInitWindowSize(2000, 1600);**  **gluOrtho2D(0.0,150.0,0.0,150.0); //resize the axis size**  **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  /\* 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(10.0);  // Draw a Red 1x1 Square centered at origin  //For the background  glColor3ub(0,191,255);  glBegin(GL\_QUADS);  glVertex2f(0.0f,0.0f);  glVertex2f(120.0f,0.0f);  glVertex2f(120.0f,100.0f);  glVertex2f(0.0f,100.0f);  glEnd();  //for base green grass line  glColor3f(0.0f,0.0f,0.0f);  glBegin(GL\_QUADS);  glVertex2f(0.0f,0.0f);  glVertex2f(130.0f,0.0f);  glVertex2f(130.0f,5.0f);  glVertex2f(0.0f,5.0f);  glEnd();  //for Tree trunk  glColor3ub(184,134,11);  glBegin(GL\_QUADS);  glVertex2f(55.0f,5.0f);  glVertex2f(65.0f,5.0f);  glVertex2f(62.0f,40.0f);  glVertex2f(58.0f,40.0f);  glEnd();  //for first layer Tree leaf  glColor3ub(250, 0, 187);  glBegin(GL\_QUADS);  glVertex2f(45.0f,40.0f);  glVertex2f(75.0f,40.0f);  glVertex2f(67.0f,45.0f);  glVertex2f(53.0f,45.0f);  glEnd();  //for second layer Tree leaf  glColor3ub(250, 0, 187);  glBegin(GL\_QUADS);  glVertex2f(49.0f,45.0f);  glVertex2f(71.0f,45.0f);  glVertex2f(63.0f,50.0f);  glVertex2f(57.0f,50.0f);  glEnd();  //for third layer Tree leaf  glColor3ub(250, 0, 187);  glBegin(GL\_POLYGON);  glVertex2f(53.0f,50.0f);  glVertex2f(67.0f,50.0f);  glVertex2f(63.0f,55.0f);  glVertex2f(57.0f,55.0f);  glEnd();  //for fourth layer Tree leaf  glColor3ub(250, 0, 187);  glBegin(GL\_POLYGON);  glVertex2f(57.0f,55.0f);  glVertex2f(63.0f,55.0f);  glVertex2f(60.0f,60.0f);  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  glutCreateWindow("Tree"); // Create a window with the given title  glutInitWindowSize(320, 320);  gluOrtho2D(0.0,120,0.0,100); //resize the axis size  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>**  **#include <GL/glut.h>**  **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(10.0);**  **//For the background**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(350.0f,0.0f);**  **glVertex2f(350.0f,300.0f);**  **glVertex2f(0.0f,300.0f);**  **glEnd();**  **glColor3ub(173, 255, 47);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(300.0f,0.0f);**  **glVertex2f(300.0f,5.0f);**  **glVertex2f(0.0f,5.0f);**  **glEnd();**  **//for white line**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(185.0f,2.0f);**  **glVertex2f(220.0f,2.0f);**  **glVertex2f(220.0f,5.0f);**  **glVertex2f(185.0f,5.0f);**  **glEnd();**  **//for lamppost pillar**  **glColor3ub(255,0 ,0);**  **glBegin(GL\_POLYGON);**  **glVertex2f(170.0f,5.0f);**  **glVertex2f(175.0f,5.0f);**  **glVertex2f(175.0f,55.0f);**  **glVertex2f(170.0f,55.0f);**  **glEnd();**  **//for lamp holder**  **glColor3ub(128,128,128);**  **glBegin(GL\_QUADS);**  **glVertex2f(170.0f,55.0f);**  **glVertex2f(200.0f,55.0f);**  **glVertex2f(200.0f,60.0f);**  **glVertex2f(170.0f,60.0f);**  **glEnd();**  **//for lamp**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(185.0f,50.5f);**  **glVertex2f(200.0f,50.5f);**  **glVertex2f(200.0f,57.0f);**  **glVertex2f(185.0f,57.0f);**  **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**  **glutCreateWindow("Lamp post"); // Create a window with the given title**  **glutInitWindowSize(1500, 1200);**  **gluOrtho2D(0.0,250.0,0.0,150.0); //resize the axis size**  **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  /\* 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(10.0);  // Draw a Red 1x1 Square centered at origin  //For the background  glColor3ub(0,191,255);  glBegin(GL\_QUADS);  glVertex2f(0.0f,0.0f);  glVertex2f(130.0f,0.0f);  glVertex2f(130.0f,100.0f);  glVertex2f(0.0f,100.0f);  glEnd();  //for base green grass line  glColor3ub(124,252,0);  glBegin(GL\_QUADS);  glVertex2f(0.0f,0.0f);  glVertex2f(130.0f,0.0f);  glVertex2f(130.0f,5.0f);  glVertex2f(0.0f,5.0f);  glEnd();  //for bench  //first paya  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(90.0f,5.0f);  glVertex2f(92.5f,5.0f);  glVertex2f(92.5f,18.0f);  glVertex2f(90.0f,18.0f);  glEnd();  //second paya  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(94.0f,5.0f);  glVertex2f(96.0f,5.0f);  glVertex2f(96.0f,18.0f);  glVertex2f(94.0f,18.0f);  glEnd();  //bench seat position  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(88.0f,18.0f);  glVertex2f(125.0f,18.0f);  glVertex2f(125.0f,23.0f);  glVertex2f(88.0f,23.0f);  glEnd();  //fourth paya  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(123.0f,5.0f);  glVertex2f(120.5f,5.0f);  glVertex2f(120.5f,18.0f);  glVertex2f(123.0f,18.0f);  glEnd();  //third paya  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(119.0f,5.0f);  glVertex2f(117.0f,5.0f);  glVertex2f(117.0f,18.0f);  glVertex2f(119.0f,18.0f);  glEnd();  //backside first portion  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(96.0f,23.0f);  glVertex2f(98.0f,23.0f);  glVertex2f(98.0f,40.0f);  glVertex2f(96.0f,40.0f);  glEnd();  //backside second portion  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(117.0f,23.0f);  glVertex2f(115.0f,23.0f);  glVertex2f(115.0f,40.0f);  glVertex2f(117.0f,40.0f);  glEnd();  //backside third portion  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(90.0f,29.0f);  glVertex2f(123.0f,29.0f);  glVertex2f(123.0f,32.0f);  glVertex2f(90.0f,32.0f);  glEnd();  //backside fourth portion  glColor3ub(139,69,19);  glBegin(GL\_QUADS);  glVertex2f(90.0f,34.0f);  glVertex2f(123.0f,34.0f);  glVertex2f(123.0f,37.0f);  glVertex2f(90.0f,37.0f);  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  glutCreateWindow("Bench"); // Create a window with the given title  glutInitWindowSize(320, 320);  gluOrtho2D(50.0,130.0,0.0,100.0); //resize the axis size  glutDisplayFunc(display); // Register display callback handler for window re-paint  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |