**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>  #include<GL/glut.h>  void building()  {  //building ara  glBegin(GL\_POLYGON);  glColor3ub(218,160,109);  glVertex2f(3,0);  glVertex2f(3,7);  glVertex2f(7,7);  glVertex2f(7,0);  glEnd();  //side are  glBegin(GL\_POLYGON);  glColor3ub(128,0,32);  glVertex2f(3,0);  glVertex2f(3,7);  glVertex2f(2,6);  glVertex2f(2,0);  glEnd();  //door  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(4.5,0);  glVertex2f(5.5,0);  glVertex2f(5.5,1);  glVertex2f(4.5,1);  glEnd();  //1st window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,2);  glVertex2f(3.5,1.5);  glVertex2f(4.5,1.5);  glVertex2f(4.5,2);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,1.5);  glVertex2f(6.5,1.5);  glVertex2f(6.5,2);  glVertex2f(5.5,2);  //2nd window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,3);  glVertex2f(3.5,2.5);  glVertex2f(4.5,2.5);  glVertex2f(4.5,3);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,3);  glVertex2f(5.5,2.5);  glVertex2f(6.5,2.5);  glVertex2f(6.5,3);  //3rd window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,4);  glVertex2f(3.5,3.5);  glVertex2f(4.5,3.5);  glVertex2f(4.5,4);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,4);  glVertex2f(5.5,3.5);  glVertex2f(6.5,3.5);  glVertex2f(6.5,4);  //4th window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,5);  glVertex2f(3.5,4.5);  glVertex2f(4.5,4.5);  glVertex2f(4.5,5);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,5);  glVertex2f(5.5,4.5);  glVertex2f(6.5,4.5);  glVertex2f(6.5,5);  //5th window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,6);  glVertex2f(3.5,5.5);  glVertex2f(4.5,5.5);  glVertex2f(4.5,6);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,6);  glVertex2f(5.5,5.5);  glVertex2f(6.5,5.5);  glVertex2f(6.5,6);  glEnd();  }  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)  building();  tree();  glFlush(); // Render now  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-** |
| **Code-** #include<windows.h>  #include<GL/glut.h>  void tree()  { glBegin(GL\_POLYGON);  glColor3ub(165,42,42);  glVertex2f(0.8,0);  glVertex2f(1.2,0);  glVertex2f(1.2,2.5);  glVertex2f(0.8,2.5);  glEnd();  //leaf  glBegin(GL\_POLYGON);  glColor3f(0,1,0);  glVertex2f(1,5);  glVertex2f(0,3);  glVertex2f(2,3);  glBegin(GL\_POLYGON);  glColor3f(0,1,0);  glVertex2f(1,3.8);  glVertex2f(2,2.5);  glVertex2f(0,2.5);  glEnd(); }  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)  tree();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Lab Task 3"); // Create a window with the given title  gluOrtho2D(0,20,-2,15);  glutInitWindowSize(320, 320); // Set the window's initial width & height  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 lamppost()  {  glBegin(GL\_POLYGON);  glColor3ub(82,86,78);  glVertex2f(8.1,0.2);  glVertex2f(8,0);  glVertex2f(8.4,0);  glVertex2f(8.3,0.2);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(82,86,78);  glVertex2f(8.1,0.2);  glVertex2f(8.3,0.2);  glVertex2f(8.3,3);  glVertex2f(8.1,3);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,255,78);  glVertex2f(8.1,3);  glVertex2f(8.3,3);  glVertex2f(8.4,3.6);  glVertex2f(8,3.6);  glEnd();  } void display()  {glClearColor(0.0f, 0.0f, 0.0f, 0.0f);  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  lamppost();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Lab Task 3"); // Create a window with the given title  gluOrtho2D(0,20,-2,15);    glutInitWindowSize(320, 320); // Set the window's initial width & height  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>  #include<GL/glut.h>  void bench()  {  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10,1.1);  glVertex2f(11.6,1.1);  glVertex2f(11.6,1);  glVertex2f(10,1);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10,0.9);  glVertex2f(11.6,0.9);  glVertex2f(11.6,0.8);  glVertex2f(10,0.8);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10.1,0.7);  glVertex2f(10,0.55);  glVertex2f(11.6,0.55);  glVertex2f(11.5,0.7);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10,0.55);  glVertex2f(11.6,0.55);  glVertex2f(11.6,0.52);  glVertex2f(10,0.52);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.04,0.25);  glVertex2f(10.1,0.25);  glVertex2f(10.1,0.52);  glVertex2f(10.04,0.52);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.45,0.52);  glVertex2f(11.5,0.52);  glVertex2f(11.5,0.25);  glVertex2f(11.45,0.25);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,0.52);  glVertex2f(11.4,0.52);  glVertex2f(11.4,0.4);  glVertex2f(11.36,0.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,0.52);  glVertex2f(10.24,0.52);  glVertex2f(10.24,0.4);  glVertex2f(10.2,0.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,0.8);  glVertex2f(10.24,0.8);  glVertex2f(10.24,0.7);  glVertex2f(10.2,0.7);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,1);  glVertex2f(10.24,1);  glVertex2f(10.24,0.9);  glVertex2f(10.2,0.9);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,1.15);  glVertex2f(10.24,1.15);  glVertex2f(10.24,1.1);  glVertex2f(10.2,1.1);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,0.8);  glVertex2f(11.4,0.8);  glVertex2f(11.4,0.7);  glVertex2f(11.36,0.7);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,1);  glVertex2f(11.4,1);  glVertex2f(11.4,0.9);  glVertex2f(11.36,0.9);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,1.15);  glVertex2f(11.4,1.15);  glVertex2f(11.4,1.1);  glVertex2f(11.36,1.1);  glEnd();  }  void display()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to white  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  bench();  glFlush(); // Render now  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Lab Task 3"); // Create a window with the given title  gluOrtho2D(0,20,-2,15);  glutInitWindowSize(320, 320); // Set the window's initial width & height  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>  #include<GL/glut.h>  void building()  {  //building ara  glBegin(GL\_POLYGON);  glColor3ub(218,160,109);  glVertex2f(3,0);  glVertex2f(3,7);  glVertex2f(7,7);  glVertex2f(7,0);  glEnd();  //side are  glBegin(GL\_POLYGON);  glColor3ub(128,0,32);  glVertex2f(3,0);  glVertex2f(3,7);  glVertex2f(2,6);  glVertex2f(2,0);  glEnd();  //door  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(4.5,0);  glVertex2f(5.5,0);  glVertex2f(5.5,1);  glVertex2f(4.5,1);  glEnd();  //1st window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,2);  glVertex2f(3.5,1.5);  glVertex2f(4.5,1.5);  glVertex2f(4.5,2);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,1.5);  glVertex2f(6.5,1.5);  glVertex2f(6.5,2);  glVertex2f(5.5,2);  //2nd window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,3);  glVertex2f(3.5,2.5);  glVertex2f(4.5,2.5);  glVertex2f(4.5,3);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,3);  glVertex2f(5.5,2.5);  glVertex2f(6.5,2.5);  glVertex2f(6.5,3);  //3rd window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,4);  glVertex2f(3.5,3.5);  glVertex2f(4.5,3.5);  glVertex2f(4.5,4);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,4);  glVertex2f(5.5,3.5);  glVertex2f(6.5,3.5);  glVertex2f(6.5,4);  //4th window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,5);  glVertex2f(3.5,4.5);  glVertex2f(4.5,4.5);  glVertex2f(4.5,5);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,5);  glVertex2f(5.5,4.5);  glVertex2f(6.5,4.5);  glVertex2f(6.5,5);  //5th window  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(3.5,6);  glVertex2f(3.5,5.5);  glVertex2f(4.5,5.5);  glVertex2f(4.5,6);  glBegin(GL\_QUADS);  glColor3ub(128,0,32);  glVertex2f(5.5,6);  glVertex2f(5.5,5.5);  glVertex2f(6.5,5.5);  glVertex2f(6.5,6);  glEnd();  }  void tree()  { //root  glBegin(GL\_POLYGON);  glColor3ub(165,42,42);  glVertex2f(0.8,0);  glVertex2f(1.2,0);  glVertex2f(1.2,2.5);  glVertex2f(0.8,2.5);  glEnd();  //leaf  glBegin(GL\_POLYGON);  glColor3f(0,1,0);  glVertex2f(1,5);  glVertex2f(0,3);  glVertex2f(2,3);  glBegin(GL\_POLYGON);  glColor3f(0,1,0);  glVertex2f(1,3.8);  glVertex2f(2,2.5);  glVertex2f(0,2.5);  glEnd();  }  void lamppost()  {  glBegin(GL\_POLYGON);  glColor3ub(82,86,78);  glVertex2f(8.1,0.2);  glVertex2f(8,0);  glVertex2f(8.4,0);  glVertex2f(8.3,0.2);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(82,86,78);  glVertex2f(8.1,0.2);  glVertex2f(8.3,0.2);  glVertex2f(8.3,3);  glVertex2f(8.1,3);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,255,78);  glVertex2f(8.1,3);  glVertex2f(8.3,3);  glVertex2f(8.4,3.6);  glVertex2f(8,3.6);  glEnd();  }  void bench()  {  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10,1.1);  glVertex2f(11.6,1.1);  glVertex2f(11.6,1);  glVertex2f(10,1);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10,0.9);  glVertex2f(11.6,0.9);  glVertex2f(11.6,0.8);  glVertex2f(10,0.8);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10.1,0.7);  glVertex2f(10,0.55);  glVertex2f(11.6,0.55);  glVertex2f(11.5,0.7);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(234,161,83);  glVertex2f(10,0.55);  glVertex2f(11.6,0.55);  glVertex2f(11.6,0.52);  glVertex2f(10,0.52);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.04,0.25);  glVertex2f(10.1,0.25);  glVertex2f(10.1,0.52);  glVertex2f(10.04,0.52);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.45,0.52);  glVertex2f(11.5,0.52);  glVertex2f(11.5,0.25);  glVertex2f(11.45,0.25);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,0.52);  glVertex2f(11.4,0.52);  glVertex2f(11.4,0.4);  glVertex2f(11.36,0.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,0.52);  glVertex2f(10.24,0.52);  glVertex2f(10.24,0.4);  glVertex2f(10.2,0.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,0.8);  glVertex2f(10.24,0.8);  glVertex2f(10.24,0.7);  glVertex2f(10.2,0.7);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,1);  glVertex2f(10.24,1);  glVertex2f(10.24,0.9);  glVertex2f(10.2,0.9);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(10.2,1.15);  glVertex2f(10.24,1.15);  glVertex2f(10.24,1.1);  glVertex2f(10.2,1.1);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,0.8);  glVertex2f(11.4,0.8);  glVertex2f(11.4,0.7);  glVertex2f(11.36,0.7);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,1);  glVertex2f(11.4,1);  glVertex2f(11.4,0.9);  glVertex2f(11.36,0.9);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(143,139,129);  glVertex2f(11.36,1.15);  glVertex2f(11.4,1.15);  glVertex2f(11.4,1.1);  glVertex2f(11.36,1.1);  glEnd();  }  void land() {  glColor3ub(47, 98, 60);  glBegin(GL\_POLYGON);  glVertex2f(0, -2);glVertex2f(20, -2);  glVertex2f(20, -2);glVertex2f(20, 1);  glVertex2f(20, 1);glVertex2f(0, 1);  glVertex2f(0, 1);glVertex2f(0, -2);  glEnd();  }  void display()  {glClearColor(0.0f, 0.0f, 0.0f, 0.0f);  //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)  land();  building();  tree();  lamppost();  bench();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Lab Task 3"); // Create a window with the given title  gluOrtho2D(0,20,-2,15);  glutInitWindowSize(200, 50); // Set the window's initial width & height  glutDisplayFunc(display); // Register display callback handler for window re-paint  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |