**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-** |
| **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  #include <math.h>  void circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void tree(){  glBegin(GL\_POLYGON); //red  glColor3ub(106, 86, 25);  glVertex2f(-6.0, -35.0);  glVertex2f(6.0, -35.0);  glVertex2f(6.0, -35.0);  glVertex2f(4.0, 4.0);  glVertex2f(4.0, 4.0);  glVertex2f(-4.0, 4.0);  glVertex2f(-4.0, 4.0);  glVertex2f(-6.0, -35.0);  glEnd();  }  /\* 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, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(7.5);  tree();  circle(10.0,-8, 0, 96, 198, 34);  circle(10.0, 8, 0, 96, 198, 34);  circle(10.0, -10,11, 96, 198, 34);  circle(10.0, 10,11, 96, 198, 34);  circle(10.0, 0, 15, 96, 198, 34);  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(980, 720);// Set the window's initial width & height // Initialize GLUT  glutCreateWindow("OpenGL Setup Test");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-100.0, 100.0, -100.0, 100.0);  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  #include <math.h>  void circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void lamppost (){  glBegin(GL\_POLYGON); //red  glColor3ub(255, 255, 255);  glVertex2f(60.0, -35.0);  glVertex2f(63.0, -35.0);  glVertex2f(63.0, -35.0);  glVertex2f(63.0, 35.0);  glVertex2f(63.0, 35.0);  glVertex2f(60.0, 35.0);  glVertex2f(60.0, 35.0);  glVertex2f(60.0, -35.0);  glEnd();  glBegin(GL\_POLYGON); //red  glColor3ub(255, 255, 255);  glVertex2f(47.0, 44.0);  glVertex2f(66.0, 31.0);  glVertex2f(66.0, 31.0);  glVertex2f(66.0, 34.0);  glVertex2f(64.0, 34.0);  glVertex2f(51.0, 44.0);  glVertex2f(51.0, 44.0);  glVertex2f(47.0, 44.0);  glEnd();  }  /\* 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, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(7.5);  circle(4.0, 50, 40, 234, 221, 14);  circle(3.0, 50, 40, 255, 240, 0);  lamppost ();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(980, 720);// Set the window's initial width & height // Initialize GLUT  glutCreateWindow("OpenGL Setup Test");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-100.0, 100.0, -100.0, 100.0);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-** |
| **Code-** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 5**  Use the building, tree, lamppost and bench to create a scenario |
| **Graph Plot (Picture)-** |
| **Code-** |
| **Output Screenshot (Full Screen)-** |