**Lab Taks-1**

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 the given deadline in VUES to the section named Lab Tak-1
* Must include resources for all the section in the table

|  |
| --- |
| **Question-**  Draw the object- |
| **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);  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(7.5);  // Draw a Red 1x1 Square centered at origin  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(0.2f, 0.2f); // x, y  glVertex2f(0.8f, 0.2f);  glVertex2f(0.8f, 0.2f); // x, y  glVertex2f(0.8f, 0.6f);  glVertex2f(0.8f, 0.6f);  glVertex2f(0.2f, 0.6f);  glVertex2f(0.2f, 0.6f);  glVertex2f(0.2f, 0.2f);  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("OpenGL Setup"); // Create a window with the given title  glutInitWindowSize(1000, 1000); // 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-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  /\*  \* GL02Primitive.cpp: Vertex, Primitive and Color  \* Draw Simple 2D colored Shapes: quad, triangle and polygon.  \*/  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  /\* Initialize OpenGL Graphics \*/  void initGL() {  // Set "clearing" or background color  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  }  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glBegin(GL\_POLYGON);  glColor3f(1.0f, 0.0f, 0.0f);  glVertex2f(0.3f, 0.6f);  glVertex2f(0.2f, 0.2f);  glVertex2f(0.8f, 0.2f);  glVertex2f(0.7f, 0.6f);  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("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutInitWindowPosition(50, 50); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  /\*  \* GL02Primitive.cpp: Vertex, Primitive and Color  \* Draw Simple 2D colored Shapes: quad, triangle and polygon.  \*/  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  /\* Initialize OpenGL Graphics \*/  void initGL() {  // Set "clearing" or background color  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Black and opaque  }  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color  glBegin(GL\_POLYGON); // These vertices form a closed polygon  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(0.9f, 0.4f);  glVertex2f(0.7f, 0.6f);  glVertex2f(0.7f, 0.5f);  glVertex2f(0.3f, 0.5f);  glVertex2f(0.3f, 0.3f);  glVertex2f(0.7f, 0.3f);  glVertex2f(0.7f, 0.2f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 0.0f, 0.0f); // Red  glVertex2f(-0.8f, 0.6f); // x, y  glVertex2f(-0.8f, 0.3f);  glVertex2f(-0.3f, 0.3f);  glVertex2f(-0.3f, 0.6f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 0.0f, 1.0f);  glVertex2f(-0.3f, -0.3f); // x, y  glVertex2f(-0.6f, -0.4f);  glVertex2f(-0.3f, -0.6f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(0.5f, -0.2f); // x, y  glVertex2f(0.3f, -0.5f);  glVertex2f(0.7f, -0.5f);  glEnd();  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(-1.0f, 0.0f); // x, y  glVertex2f(1.0f, 0.0f);  glVertex2f(0.0f, 0.6f); // x, y  glVertex2f(0.0f, -0.6f);  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("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |