**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  void display() {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glLineWidth(0.5);  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f); //  glVertex2f(0.0f, 0.0f); // x, y  glVertex2f(0.5f, 0.0f);  glVertex2f(0.5f, 0.0f);  glVertex2f(0.5f, 0.5f);  glVertex2f(0.5f, 0.5f);  glVertex2f(0.0f, 0.5f);  glVertex2f(0.0f, 0.0f);  glVertex2f(0.0f, 0.5f);  glVertex2f(0.0f, 0.0f);  glEnd();  glFlush();  }  /\* 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 Test"); // Create a window with the given title  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-**  Draw the object- |
| **Graph Plot (Picture)-** |
| Code-#include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  void display() {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glBegin(GL\_QUADS);  glColor3f(1.0f, 0.0f, 0.0f); // Red  glVertex2f(0.0f, 0.0f); // x, y  glVertex2f(0.5f, 0.0f);//-----------------------QUADS----------------------------------  glVertex2f(0.4f, 0.5f);  glVertex2f(0.1f, 0.5f);  glEnd();  glFlush();  }  /\* 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 Test"); // Create a window with the given title  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-**  Draw the object-  Octagon Shape | Area & Angles - Video & Lesson Transcript | Study.com |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h>  // for MS Windows #include <GL/glut.h>  // GLUT, include glu.h and gl.h  void display() {   glClearColor(1.0f, 1.0f, 1.0f, 1.0f);   glClear(GL\_COLOR\_BUFFER\_BIT);    glBegin(GL\_POLYGON); glColor3f(1.0f, 0.0f, 0.0f); // Red    glVertex2f(0.2f, 0.5f);   glVertex2f(0.5f, 0.2f);   glVertex2f(0.5f, 0.2f);   glVertex2f(0.5f, -0.2f);     glVertex2f(0.5f, -0.2f);   glVertex2f(0.2f, -0.5f);   glVertex2f(0.2f, -0.5f);   glVertex2f(-0.2f, -0.5f);   glVertex2f(-0.2f, -0.5f);   glVertex2f(-0.5f, -0.2f);   glVertex2f(-0.5f, -0.2f);   glVertex2f(-0.5f, 0.2f);   glVertex2f(-0.5f, 0.2f);   glVertex2f(-0.2f, 0.5f);   glVertex2f(-0.2f, 0.5f);   glVertex2f(0.2f, 0.5f); glEnd();   glFlush(); }  /\* 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 Test"); // Create a window with the given title   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-**  Draw the object- |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  void \_axis() {  glColor3ub(0, 0, 0);  glBegin(GL\_LINES);  glVertex2f(-1.0, 0.0);glVertex2f(1.0, 0.0);  glVertex2f(0.0, 1.0); glVertex2f(0.0, -1.0);  glEnd();  }  void \_rectangle() {  glColor3ub(255, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-0.8, 0.2);glVertex2f(-0.3, 0.2);  glVertex2f(-0.3, 0.2);glVertex2f(-0.3, 0.6);  glVertex2f(-0.3, 0.6);glVertex2f(-0.8, 0.6);  glVertex2f(-0.8, 0.6);glVertex2f(-0.8, 0.2);  glEnd();  }  void \_triangle3rdQuad() {  glColor3ub(117, 52, 255);  glBegin(GL\_POLYGON);  glVertex2f(-0.8, -0.45);glVertex2f(-0.3, -0.7);  glVertex2f(-0.3, -0.7);glVertex2f(-0.3, -0.2);  glVertex2f(-0.3, -0.2);glVertex2f(-0.8, -0.45);  glEnd();  }  void \_arrow() {  glColor3ub(0, 255, 0);  glBegin(GL\_POLYGON);  glVertex2f(0.2, 0.3);glVertex2f(0.5, 0.3);  glVertex2f(0.5, 0.3);glVertex2f(0.5, 0.5);  glVertex2f(0.5, 0.5);glVertex2f(0.2, 0.5);  glVertex2f(0.2, 0.5);glVertex2f(0.2, 0.3);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(0.5, 0.2);glVertex2f(0.8, 0.4);  glVertex2f(0.8, 0.4);glVertex2f(0.5, 0.6);  glVertex2f(0.5, 0.6);glVertex2f(0.5, 0.2);  glEnd();  }  void \_triangle4thQuad() {  glColor3ub(255, 255, 0);  glBegin(GL\_POLYGON);  glVertex2f(0.4, -0.2);glVertex2f(0.1, -0.6);  glVertex2f(0.1, -0.6);glVertex2f(0.7, -0.6);  glVertex2f(0.7, -0.6);glVertex2f(0.4, -0.2);  glEnd();  }  void display() {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  \_axis();  \_rectangle();  \_triangle3rdQuad();  \_arrow();  \_triangle4thQuad();  glFlush();  } /\* 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 Test"); // Create a window with the given title  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)-** |