#Draw Sample Window (White)

```
#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)


        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 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;
}
```

//Draw Points

```
#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(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)
        glPointSize(5.0);
        // Draw a Red 1x1 Square centered at origin
        glBegin(GL_POINTS);             // Each set of 4 vertices form a quad
        glColor3f(1.0f, 0.0f, 0.0f); // Red
        glVertex2f(-0.0f, -0.0f);    // x, y

        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 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;
}
```

//Draw Line

```
#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(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);
	// Draw a Red 1x1 Square centered at origin
	glBegin(GL_LINES);		// Each set of 4 vertices form a quad
	glColor3f(1.0f, 0.0f, 0.0f); // Red
	glVertex2f(0.0f, 0.0f);    // x, y
	glVertex2f(1.0f, 0.0f);    // x, y

	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(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;
}
```

Draw X, Y Axis

```
#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(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(.5);
	// Draw a Red 1x1 Square centered at origin
	glBegin(GL_QUADS);              // Each set of 4 vertices form a quad
	glColor3f(1.0f, 0.0f, 0.0f); // Red

	glVertex2f(0.0f, 0.0f);    // x, y
	glVertex2f(1.0f, 0.0f);    // x, y

	glVertex2f(0.0f, 0.0f);    // x, y
	glVertex2f(0.0f, 1.0f);    // x, y
	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(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;
}
```

Draw Ploygon

```
/*
* 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(0.0f, 0.0f, 0.0f, 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(1.0f, 1.0f, 0.0f); // Yellow
	glVertex2f(0.4f, 0.2f);
	glVertex2f(0.6f, 0.2f);
	glVertex2f(0.7f, 0.4f);
	glVertex2f(0.6f, 0.6f);
	glVertex2f(0.4f, 0.6f);
	glVertex2f(0.3f, 0.4f);
	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;
}
```

4 Object in 4 axis

```
/*
* 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(0.0f, 0.0f, 0.0f, 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(1.0f, 1.0f, 0.0f); // Yellow

        glVertex2f(0.4f, 0.2f);
        glVertex2f(0.6f, 0.2f);
        glVertex2f(0.7f, 0.4f);
        glVertex2f(0.6f, 0.6f);
        glVertex2f(0.4f, 0.6f);
        glVertex2f(0.3f, 0.4f);
        glEnd();

                // Draw a Red 1x1 Square centered at origin
        glBegin(GL_TRIANGLES);            // Each set of 4 vertices form a quad
        glColor3f(1.0f, 0.0f, 0.0f); // Red

        glVertex2f(-0.9f, 0.3f);    // x, y
        glVertex2f(-0.5f, 0.3f);
```

```
            glVertex2f(-.7f,  0.6f);

            glEnd();

    glBegin(GL_QUADS);            // Each set of 4 vertices form a quad
            glColor3f(0.0f, 1.0f, 0.0f); // Red

            glVertex2f(-0.8f, -0.8f);    // x, y
            glVertex2f(-0.5f, -0.8f);
            glVertex2f(-0.5f, -0.5f);    // x, y
            glVertex2f(-0.8f, -0.5f);

            glEnd();

    glBegin(GL_TRIANGLES);//
    glColor3ub(232, 133, 20);//rgb color picker

    glVertex2f(+.5f, -.8f);    // x, y
            glVertex2f(+0.7f,-.8f);
            glVertex2f(+.6f, -0.4f);
            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;
}
```

Assignment:

1. Rainbow Flag
2. AIUB Text