```
 1: // $Id: konnichiwa-sekai.cpp,v 1.2 2019-05-15 17:16:28-07 - - $
 2:
 3: // Draw line from (0,0) to (1,1).
 4:
 5: #include <cmath>
 6: #include <iostream>
 7: #include <vector>
 8: using namespace std;
 9:
10: #include <GL/freeglut.h>
11: #include <libgen.h>
12:
13: struct {
14:    int width = 384;
15:    int height = 256;
16: } window;
17:
18: const GLubyte WHITE[] {255, 255, 255};
19: const GLubyte CRIMSON_GLORY[] {188, 0, 45};
20: constexpr GLfloat aspect_ratio = 2.0 / 3.0;
21:
22: void draw_japanese_flag() {
23:    glBegin (GL_POLYGON);
24:    glColor3ubv (WHITE);
25:    glVertex2f (0, 0);
26:    glVertex2f (window.width, 0);
27:    glVertex2f (window.width, window.height);
28:    glVertex2f (0, window.height);
29:    glEnd();
30:    glBegin (GL_POLYGON);
31:    glColor3ubv (CRIMSON_GLORY);
32:    const GLfloat delta = 2 * M_PI / 64;
33:    const GLfloat radius = window.height * 3.0 / 10.0;
34:    const GLfloat xoffset = window.width / 2.0;
35:    const GLfloat yoffset = window.height / 2.0;
36:    for (GLfloat theta = 0; theta < 2 * M_PI; theta += delta) {
37:       GLfloat xpos = radius * cos (theta) + xoffset;
38:       GLfloat ypos = radius * sin (theta) + yoffset;
39:       glVertex2f (xpos, ypos);
40:    }
41:    glEnd();
42: }
43:
44: void display() {
45:    glClearColor (0.0, 0.0, 0.0, 0.0);
46:    glClear (GL_COLOR_BUFFER_BIT);
47:    draw_japanese_flag();
48:    glutSwapBuffers();
49: }
50:
```

```
51:
52: void adjust_aspect (int width, int height) {
53:    if (window.width != width) {
54:       height = width * aspect_ratio;
55:    }else if (window.height != height) {
56:       width = height / aspect_ratio;
57:    }else {
58:       return;
59:    }
60:    window.width = width;
61:    window.height = height;
62:    glutReshapeWindow (window.width, window.height);
63: }
64:
65: void reshape (int width, int height) {
66:    adjust_aspect (width, height);
67:    window.width = width;
68:    window.height = height;
69:    glMatrixMode (GL_PROJECTION);
70:    glLoadIdentity();
71:    gluOrtho2D (0, window.width, 0, window.height);
72:    glMatrixMode (GL_MODELVIEW);
73:    glViewport (0, 0, window.width, window.height);
74:    glutPostRedisplay();
75: }
76:
77: int main (int argc, char** argv) {
78:    glutInit (&argc, argv);
79:    glutInitDisplayMode (GLUT_RGBA | GLUT_DOUBLE);
80:    glutInitWindowSize (window.width, window.height);
81:    glutCreateWindow (basename (argv[0]));
82:    glutDisplayFunc (display);
83:    glutReshapeFunc (reshape);
84:    glutMainLoop();
85:    return 0;
86: }
87:
88: //TEST// mkpspdf konnichiwa-sekai.cpp.ps konnichiwa-sekai.cpp*
89:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting konnichiwa-sekai.cpp
    2: checksource konnichiwa-sekai.cpp
    3: ident konnichiwa-sekai.cpp
    4: konnichiwa-sekai.cpp:
    5:      $Id: konnichiwa-sekai.cpp,v 1.2 2019-05-15 17:16:28-07 - - $
    6: cpplint.py.perl konnichiwa-sekai.cpp
    7: Done processing konnichiwa-sekai.cpp
    8: g++ -g -O0 -Wall -Wextra -Wpedantic -Wshadow -fdiagnostics-color=never -
std=gnu++2a -Wold-style-cast konnichiwa-sekai.cpp -o konnichiwa-sekai -lm -lglu
t -lGLU -lGL -lX11 -ldrm -lm
    9: rm -f konnichiwa-sekai.o
   10: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished konnichiwa-sekai.cpp
```