```cpp
// $Id: hello-gl.cpp,v 1.16 2019-02-22 15:57:55-08 - - $

// Display text "Hello World" in a window.

#include <iostream>
#include <string>
using namespace std;

#include <GL/freeglut.h>
#include <libgen.h>

// Characteristics of the window.
struct window {
   string name;
   int width {256};
   int height {192};
} window;

void show (const char* func) {
   cout << window.name << ": " << func << " "
        << window.width << "x" << window.height << endl;
}

// Called by glutMainLoop to display window contents.
void display() {
   show (__PRETTY_FUNCTION__);

   // Glut strings use unsigned char instead of signed char.
   static const string hello {"Hello, World"};

   // Pointer to one of the bitmap fonts.
   void* font = GLUT_BITMAP_HELVETICA_18;

   // Width and height in pixels of the bitmap string.
   static auto u_str = reinterpret_cast<const GLubyte*> (hello.c_str());
   int str_width = glutBitmapLength (font, u_str);
   int str_height = glutBitmapHeight (font);

   // Set the background default color and clear the window.
   glClearColor (1.0, 1.0, 0.0, 1.0);
   glClear (GL_COLOR_BUFFER_BIT);

   // Set the color of the letters in the message.
   static const GLubyte BLUE[] = {0, 0, 255};
   glColor3ubv (BLUE);

   // Position (x,y) of the left end and base of the string.
   GLfloat xpos = window.width / 2.0 - str_width / 2.0;
   GLfloat ypos = window.height / 2.0 - str_height / 4.0;

   // Draw the bitmap in the window.
   glRasterPos2f (xpos, ypos);
   glutBitmapString (font, u_str);

   // Swap the passive and active buffers to display the window.
   glutSwapBuffers();
}

```

```

void reshape (int width, int height) {
   show (__PRETTY_FUNCTION__);
   window.width = width;
   window.height = height;
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D (0, window.width, 0, window.height);
   glMatrixMode (GL_MODELVIEW);
   glViewport (0, 0, window.width, window.height);
   glutPostRedisplay();
}

int main (int argc, char** argv) {
   window.name = basename (argv[0]);
   glutInit (&argc, argv);
   glutInitDisplayMode (GLUT_RGBA | GLUT_DOUBLE);
   glutInitWindowSize (window.width, window.height);
   glutCreateWindow (window.name.c_str());
   glutDisplayFunc (display);
   glutReshapeFunc (reshape);
   glutMainLoop();
   return 0;
}

//TEST// mkpspdf hello-gl.ps hello-gl.cpp*

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting hello-gl.cpp
checksource hello-gl.cpp
ident hello-gl.cpp
hello-gl.cpp:
     $Id: hello-gl.cpp,v 1.16 2019-02-22 15:57:55-08 - - $
cpplint.py.perl hello-gl.cpp
Done processing hello-gl.cpp
g++ -g -O0 -Wall -Wextra -Wpedantic -Wshadow -fdiagnostics-color=never -std=gnu++17 -Wold-style-cast hello-gl.cpp -o hello-gl -lm -lglut -lGLU -lGL -lX11 -ldrm -lm
rm -f hello-gl.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished hello-gl.cpp
```