```cpp
// $Id: hello-gl.cpp,v 1.29 2014-07-22 16:38:07-07 - - $

// 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;

// Called by glutMainLoop to display window contents.
void display() {
   cout << __func__ << "()" << endl;

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

   // Pointer to one of the bitmap fonts.
   void* font = GLUT_BITMAP_TIMES_ROMAN_24;

   // Width and height in pixels of the bitmap string.
   int str_width = glutBitmapLength (font, (GLubyte*) hello.c_str());
   int str_height = glutBitmapHeight (font);

   // Set the background default color and clear the window.
   glClearColor (0.25, 0.25, 0.25, 1.0);
   glClear (GL_COLOR_BUFFER_BIT);

   // Set the color of the letters in the message.
   const GLubyte GREEN[] = {0x00, 0xFF, 0x00};
   glColor3ubv (GREEN);

   // Position (x,y) of the left end and base of the string.
   float xpos = window.width / 2.0 - str_width / 2.0;
   float ypos = window.height / 2.0 - str_height / 4.0;
   glRasterPos2f (xpos, ypos);

   // Draw the bitmap in the window.
   //glutBitmapString (font, hello.c_str());
   for (auto ch: hello) glutBitmapCharacter (font, ch);

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

```

```

void reshape (int width, int height) {
   cout << __func__ << "(" << width << "," << height << ")" << endl;
   window.width = width;
   window.height = height;
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity();
   glOrtho (0, window.width, 0, window.height, -1, +1);
   glMatrixMode (GL_MODELVIEW);
   glViewport (0, 0, window.width, window.height);
   glutPostRedisplay();
}

void close() {
   cout << __func__ << "()" << endl;
}

void entry (int state) {
   cout << __func__ << "(";
   switch (state) {
      case GLUT_LEFT: cout << "GLUT_LEFT"; break;
      case GLUT_ENTERED: cout << "GLUT_ENTERED"; break;
      default: cout << state; break;
   }
   cout << ")" << endl;
}

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);
   glutEntryFunc (entry);
   glutCloseFunc (close);
   glutMainLoop();
   return 0;
}

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting hello-gl.cpp
hello-gl.cpp:
     $Id: hello-gl.cpp,v 1.29 2014-07-22 16:38:07-07 - - $
g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++14 hello-gl.cpp
        -o hello-gl -lglut -lGLU -lGL -lX11 -lrt -lm
rm -f hello-gl.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished hello-gl.cpp
```