# Building Programs Using Classes

- Top Down Design
- Problem 1
- An image class
- gplot class revisited
- Top Down Design for Problem 1
- Problem 2

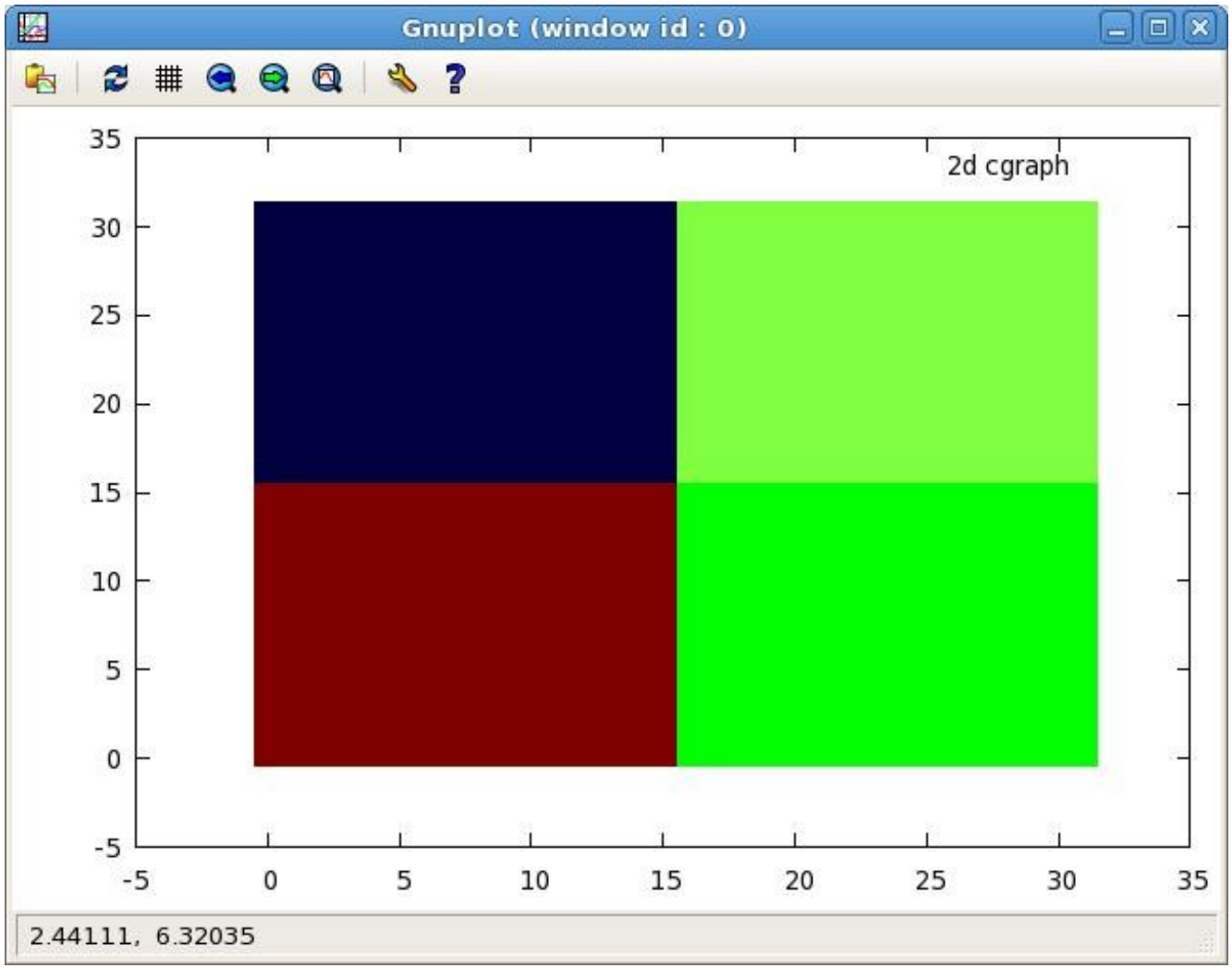
# Top Down Design

- A top-down approach is essentially breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in yet greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements.

# The Problem

- Write a program to draw a checkerboard pattern of red, green, and blue blocks with prescribed color values and a block which shows the color that results from combining the red, green, and blue values with their prescribed intensities.
- The program should read in the width and height of the blocks, the red, green, and blue intensities.

# The Problem


Gnuplot (window id : 0)
2d cgraph
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
2.44111, 6.32035

# The Class

- gimage.h
- class gimage{

```
friend class gplot;
public:
int width,height;

//allocate memory for a width x height image
gimage(int pwidth,int pheight);

//add/change a pixel with value c to the image at position (x,y) ;
void addpixel(int x, int y, unsigned char c[]);

//get the value of a pixel c[3] in the image at position (x,y);
void getpixel(int x, int y, unsigned char c[]);

private:
unsigned char *image;
};
```

# gplot class revisited

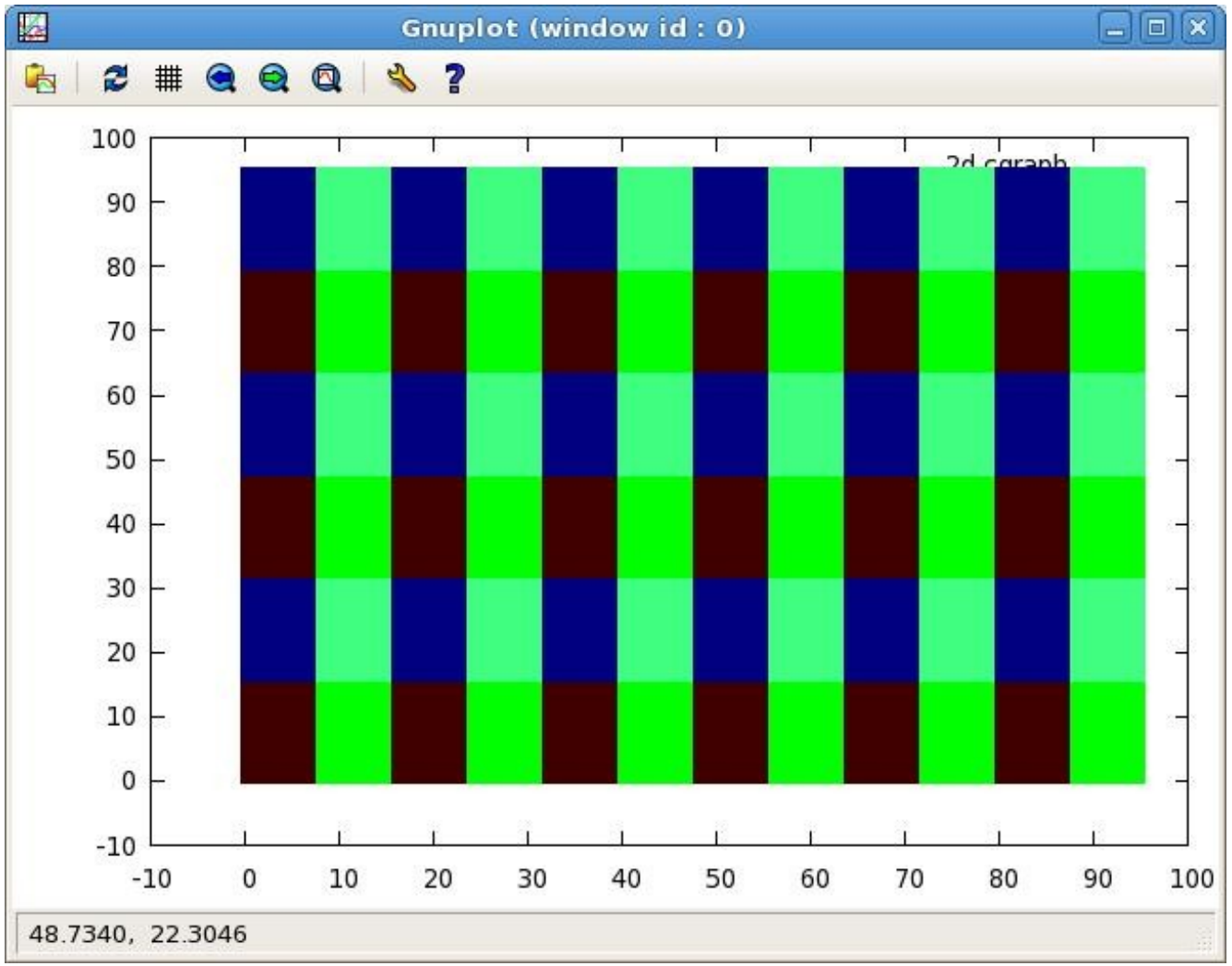
- gplot.h
- /*Display picture store in image */
  void rgbimage(gimage image);

- Image contains height and width, but these are private so gplot must be a friend class.

# Exercise

- Design a program to draw 2d array of checkerboard patterns of red, green, and blue blocks with prescribed color values and a block which shows the color that results from combining the red, green, and blue values with their prescribed intensities.
- The program should read in the width and height of the blocks, the red, green, and blue intensities, and the number of horizontal an vertical block to be drawn.

# Exercise


Gnuplot (window id : 0)
2d cgraph
100
90
80
70
60
50
40
30
20
10
0
-10
-10
0
10
20
30
40
50
60
70
80
90
100
48.7340, 22.3046