| |  | | --- | | |
| --- | --- | --- |
| |  | | --- | |  | |  | |  | |  | |  | | Version 1.5 Documentation:  | Overview | Programming |  | | --- | --- | --- | | * [Setup Notes](http://docs.google.com/setup.html) * [Notes/Caveats](http://docs.google.com/notes.html) * [FAQs](http://docs.google.com/faqs.html) | * [Basics](http://docs.google.com/basics.html) * [Objects](http://docs.google.com/objects.html) * [Functions](http://docs.google.com/functions.html) * [Constants](http://docs.google.com/constants.html) * [Source](http://docs.google.com/source.html) |  |    Basics: Here is a very basic program:  window testWindow(600, 440, 5, 5);  testWindow.ChangeTitle("CMU Graphics Package Demo");   testWindow.SetPen(0.0, 1.0, 0.0, 5);   testWindow.DrawLine(5, 10, 5, 420);    WaitNClear(testWindow);   testWindow.SetPen(RED, 0);  testWindow.SetBrush(RED);  testWindow.DrawRectangle(30, 30, 200, 200, FILLED);    image testImage("images\\scs.jpg", JPEG);  if(testImage.Valid()) {  testWindow.DrawImage(testImage, 10, 10);  } else {  testWindow.SetPen(BLACK);  testWindow.DrawString(5, 5, "Could not load image \"images\\scs.jpg\"!");  }  Let's go through an examine everything that is going on:  window testWindow(600, 440, 5, 5);   testWindow.ChangeTitle("CMU Graphics Package Demo");  Here we are creating a window object, of size 600 pixels wide and 440 pixels high, 5 pixels down and 5 pixels to the right of the upper left corner of the screen. It also sets the title of the window.  testWindow.SetPen(0.0, 1.0, 0.0, 5);   testWindow.DrawLine(5, 10, 5, 420);  This sets the pen color to green and the pen thickness to 5. This variant of the SetPen function (see the function reference for a description of all the different flavors) takes 3 floating point numbers that each range from 0.0 to 1.0, each specifying the amount of red, green and blue, respectively. The last parameter sets the pen thickness. DrawLine draws a line from x1,y1 to x2,y2. Here it draws a line from (5,10) to (5,420).  WaitNClear(testWindow);  This waits for a mouse click and clears the window.  testWindow.SetPen(RED, 0);  testWindow.SetBrush(RED);  testWindow.DrawRectangle(30, 30, 200, 200, FILLED);  This demonstrates another variant of the SetPen function. There are a huge number of predefined color constants (see the constants section for details); almost any color you'd want to use is probably already a predefined constant. We also set the brush color. This is the color that is going to be used to fill in any objects that can be filled in. We then draw a rectangle and specify that we want it to be filled (we can also specify FRAME for no fill).  image testImage("images\\scs.jpg", JPEG);  Just to show off some of the very basic imaging features of the library, here we are loading a JPEG image into the object testImage.  if(testImage.Valid()) {  testWindow.DrawImage(testImage, 10, 10);  } else {  testWindow.SetPen(BLACK);  testWindow.DrawString(5, 5, "Could not load image \"images\\scs.jpg\"!");  }  We then call Valid() to check that the image was loaded properly. If it was, we use DrawImage to place the test image at window location (10,10). If the image was not loaded successfully, we draw some text on the screen with a black pen at location (5,5) to tell the user that the image failed to load.  To learn about all of the functionality available, see the Object and Function references. |