**Image processing using the *image* module**

The *image* module is similar to the one used in our text but has at least one important difference: it does not have a getPixels method. That means we can’t get all the pixels in one long list – we have to use nested loops to modify all the pixels.This document describes how to use *image* with IDLE or other IDEs.

This module should work with gif images under any version of Python. If you’re using Anaconda Python you should also be able to use jpeg images. Either way, **image.py must be in the same folder as your program.**

**Installation**

Download image.py from Blackboard and copy it to the same folder as your program.

**image Functions** (The following assumes you have executed from image import \*)

FileImage(filename) Loads an image from the file and returns an image object

ImageWin(width, height, title) Returns a display window

EmptyImage(width, height) Returns a blank image

Pixel(r, g, b) Returns a pixel object with specified red, green, and blue values

**Image Methods** (The following assumes the name of the image is img)

img.get\_width() Returns the width of img

img.get\_height() Returns the height of img

img.get\_pixel(x, y) Returns a pixel from the x, y position in img

img.set\_pixel(x, y, pxl) Sets a pixel at the x, y position

**Note:** pxl must be a Pixel object; you can create one using Pixel(r, g, b)

img.draw(win) Displays img in the display window specified by win

img.clone() Returns a copy of img

img.save() Saves the image to its filename if it has one; **not recommended**

img.save(filename) Saves the image to specified file

**Note:** The file type is determined by the extension (.png, .gif, etc); default is .jpg

**Pixel Methods** (The following assumes the name of the pixel is p)

p.get\_red() Gets the red channel from pixel p

p.get\_green() Gets the green channel from pixel p

p.get\_blue() Gets the blue channel from pixel p