**Intro. to Computing (810:051)** **Lab 11** **Name:_____**

**Objectives**:

- Gain experience using object-based programming using existing classes, objects, and methods
- Understand simple graphics operations to draw 2D shapes

For today's lab you'll need several files, so start the lab by downloading the following file to your desktop:
http://www.cs.uni.edu/~fienup/cs051s10/labs/lab11.zip
Extract this file to the Desktop (or your flash drive) by right-clicking on lab11.zip icon and selecting Extract All.

## Part A: Turtle Graphics

The Turtle methods from the Turtle Graphics module.

| Turtle Method | Description |
|---|---|
| `t = Turtle()` | Creates a Turtle object and opens a 200 x 200 pixel drawing area. |
| `t = Turtle(width, height)` | Creates a Turtle object and opens drawing area with the specified size. |
| `t.getWidth()` | Return's the width of t's drawing window. |
| `t.getHeight()` | Return's the height of t's drawing window. |
| `t.home()` | Move t to center pointing north without drawing any lines. |
| `t.setDirection(degrees)` | Points t in specified direction (0 is east, 90 is north, 180 is west, and 270 is south) |
| `t.turn(degrees)` | Adds the degrees to t's current direction (+ clockwise, neg. counter-clockwise) |
| `t.down()` | Puts t's tail/pen down |
| `t.up()` | Puts t's tail/pen up |
| `t.move(distance)` | Moves t forward the specified distance. |
| `t.move(x,y)` | Move t to the specified x-y coordinate |
| `t.setColor(r, g, b)` | Sets the color of the pen to specified RGB value |
| `t.setWidth(penWidth)` | Sets the width of the pen in pixels (default is 2) |

Define the following functions:

- drawRectangleSimple - this function expects a Turtle object, the coordinates of the upper-left, and the coordinates of the lower-right corners of a rectangle as arguments. Each coordinate argument is a tuple containing an x, y pair.
- drawRectangle - modify the drawRectangleSimple function to take an addition RGB value for the rectangles color and an optional Boolean value named fillOn as arguments. Its default fill value is False. If the fill value is True, the function should fill the rectangle in the given color. (Hint: you can use the line width to draw the filled rectangle.)

Demonstrate your functions by writing a program with a main function that:

- creates a Turtle with a window size of 400 by 400.
- draws a gray, filled rectangle with an upper-left corner at (0, 0) and a lower-right corner at (100, -175).
- draws a simple rectangle with an upper-left corner at (-150, 150) and a lower-right corner at (150, -150).

**After you have implemented the above program, raise your hand and demonstrate your code.**

## Part B: Image Processing

Table 7.4 contains the Image object methods. The below example converts a colored image into a gray-scaled image by weighting the red, green, and blue components of each pixel for best human perception. Notice the nested for-loops that processing each pixel. When the draw method is invoked, the window containing the image must be closed before execution is returned to the main function.

```
"""
File: testgrayscale.py
Tests a function for converting a color image to
grayscale.
"""

from images import Image

def grayscale(image):
    """Converts the argument image to grayscale."""
    for y in xrange(image.getHeight()):
        for x in xrange(image.getWidth()):
            (r, g, b) = image.getPixel(x, y)    # getPixel return tuple of RGB values
            r = int(r * 0.299)                  # weight each component for human
            g = int(g * 0.587)                  # perception
            b = int(b * 0.114)
            lum = r + g + b
            image.setPixel(x, y, (lum, lum, lum))

def main(filename = "smokey.gif"):
    image = Image(filename)
    print "Close the image window to continue. "
    image.draw()
    grayscale(image)
    print "Close the image window to quit. "
    image.draw()

main()
```

Write the following image processing functions:

- Copy the above testgrayscale.py program and modify it to perform a simplified gray-scaling calculation. For each pixel, calculate the average of the r, g, b values of the color image, and set all three r, g, b values to this average. Use the same main program to test your new function.
- Write an `invert` function that expects an image argument and returns a new image that's similar to a photographic negative. Each pixel in the new image is generated by subtracting each RGB component in the old image from 255.

Demonstrate your functions by writing a program with a main function that:

- Load the image from the file smokey.gif and draw it
- Call your simplified grayscale function and draw it
- RE-Load the image from the file smokey.gif and draw it
- Call your `invert` function and draw the resulting image.

**After you have implemented the above program, raise your hand and demonstrate your code.**

**If you do not get done today, then show me the completed lab in next lab period. Make sure that you log off the computer before you leave.**