Homework Turnin

Name:	Riley H Taylor
Email:	rileytaylor@email.arizona.edu
Student ID:	23183089
Section:	2J
Course:	CS 110 17sp
Assignment:	hw3
Receipt ID:	4bc7f4f73aa171063404de7055c05967

Turnin Successful!

The following file(s) were received:

cafewall.py (3123 bytes)

```
# Author: Riley Taylor
# Course: CSC 110, Section 2J, Spring 2017
# Program: Cafe Wall
#
# This draws several rows of black and white boxes, as well as several
# examples of the Cafe Wall illusion.

from drawingpanel import *


MARGIN = 2


def main():
    panel = DrawingPanel(650, 400, background="gray")

    draw_grid(panel, 1, 4, 0, 0, 20, 0)
    draw_grid(panel, 1, 5, 50, 70, 30, 0)
    draw_grid(panel, 8, 4, 10, 150, 25, 0)
    draw_grid(panel, 6, 3, 250, 200, 25, 10)
    draw_grid(panel, 10, 5, 425, 180, 20, 10)
    draw_grid(panel, 4, 2, 400, 20, 35, 35)

# ---------------------------------------------------------------------
# box() draws a box of a particular color. If it is a black box it
# will also add a blue 'X'
#
# PARAMETERS: panel -- a variable. References the DrawingPanel
#             x -- an int. The starting x coordinate
#             y -- an int. The starting y coordinate
#             size  -- an int. The size of the box
#             color -- a string. The color of the box
# ---------------------------------------------------------------------
def box(panel, x, y, size, color):
    panel.canvas.create_rectangle(x, y, x + size, y + size,
                                  fill=color, outline=color)
    if (color == "black"):
        panel.canvas.create_line(x, y, x + size, y + size, fill="blue")
        panel.canvas.create_line(x, y + size, x + size, y, fill="blue")
```

```
# --------------------------------------------------------------------
# draw_row() draws a row of box pairs.
#
# PARAMETERS: panel -- a variable. References the DrawingPanel
#             count -- an in. How many box pairs to render
#             x -- an int. The starting x coordinate
#             y -- an int. The starting y coordinate
#             size  -- an int. The size of the box
#             color -- a string. The color of the box
# --------------------------------------------------------------------
def draw_row(panel, count, x, y, size, offset):
    for pair in range(0, count):
        box(panel, x, y, size, "black")
        box(panel, x + size + 1, y, size, "white")
        x = x + size * 2

# --------------------------------------------------------------------
# draw_grid() draws a grid of rows
#
# PARAMETERS: panel -- a variable. References the DrawingPanel
#             rows -- an int. How many rows to render
#             row_length -- an int. How many box pairs per row
#             x_start -- an int. The starting x coordinate
#             y_start -- an int. The starting y coordinate
#             box_size  -- an int. The size of the box
#             offset -- an int. The offset amount for even rows
# --------------------------------------------------------------------
def draw_grid(panel, rows, row_length, x_start, y_start, box_size,
              offset):
    for row in range(1, rows + 1):
        if ((row % 2 == 0) and offset > 0):
            x_start = x_start + offset
        else:
            x_start = x_start - offset
        draw_row(panel, row_length, x_start, y_start, box_size,
                 offset)
        y_start = y_start + box_size + MARGIN

main()
```

doodle.py (1059 bytes)

```
# Author: Riley Taylor
# Course: CSC 110, Section 2J, Spring 2017
# Program: Doodle
#
# This draws a git logo using the DrawingPanel library

from drawingpanel import *


def main():
    panel = DrawingPanel(400, 400, background="white")

    # draw the background for the logo
    panel.canvas.create_polygon(200, 15, 385, 200,
                                200, 385, 15, 200,
                                fill="orangered", outline="white")

    # Draw the lines and ovals for the git logo
    panel.canvas.create_line(130, 85, 260, 200,
                             fill="white", width=8)
    panel.canvas.create_line(200, 150, 200, 280,
                             fill="white", width=8)
    panel.canvas.create_oval(180, 130, 220, 170,
                             fill="white", outline="white", width=4)
    panel.canvas.create_oval(240, 180, 280, 220,
                             fill="white", outline="white", width=4)
    panel.canvas.create_oval(180, 260, 220, 300,
                              fill="white", outline="white", width=4)

main()
```