CSc 110, Spring 2018

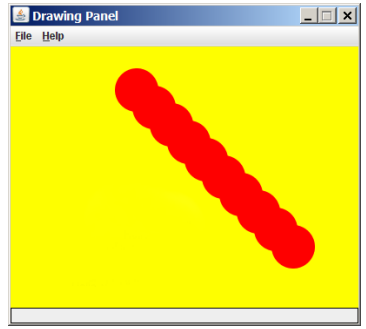
Lecture 10: More Graphics

Adapted from slides by Marty Stepp and Stuart Reges

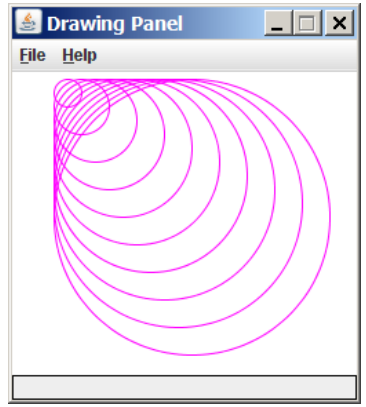
Drawing with loops

- The *x1*, *y1*, *w*, h expression can contain the loop counter, `i`.

```
panel = DrawingPanel(400, 300, background="yellow")

for i in range(1, 11):
    panel.fill_oval (100 + 20 * i, 5 + 20 * i,
                     50, 50, "red")
```

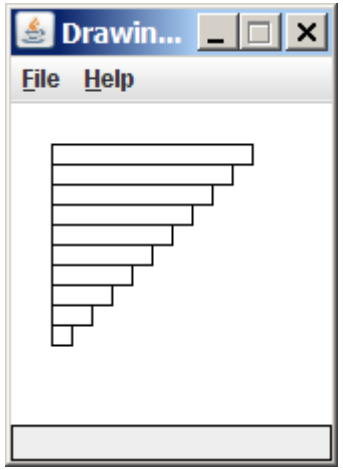



```
panel = DrawingPanel(250, 220)
for i in range(1, 11):
    panel.draw_oval (30, 5, 20 * i, 20 * i, "magenta")
```

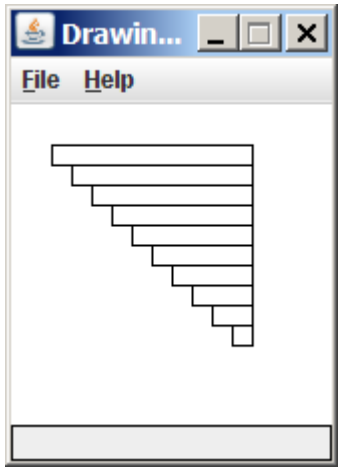


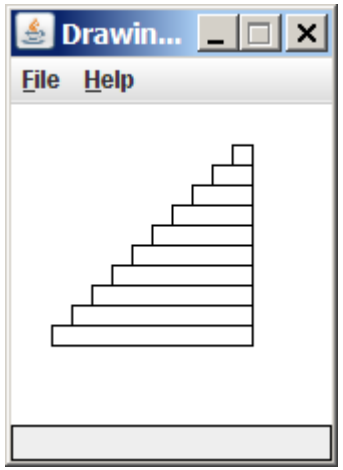
Drawing w/ loops questions

```
panel = DrawingPanel(160, 160)

for i in range(0, 10):
    panel.draw_rectangle (20, 20 + 10 * i,
                          100 - 10 * i, 10)
```



- Write variations of the above program that draw the figures at right as output.

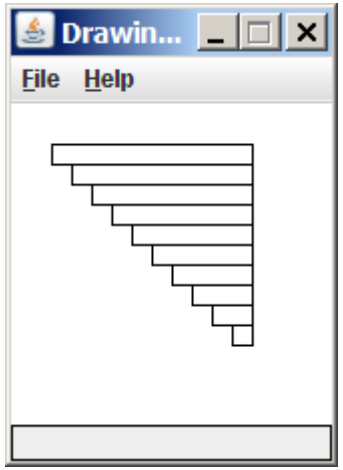




Drawing w/ loops answers

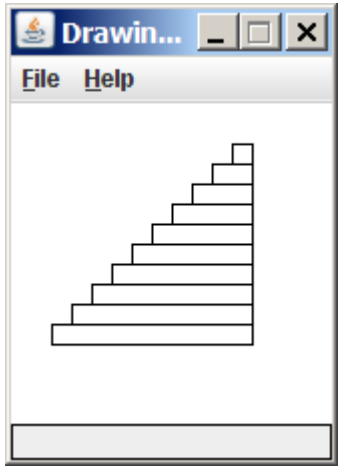
- Solution #1:

```
panel = DrawingPanel(160, 160)
for i in range(0, 10):
    panel.draw_rectangle (20 + 10 * i, 20 + 10 * i,
                          100 - 10 * i, 10)
```



- Solution #2:

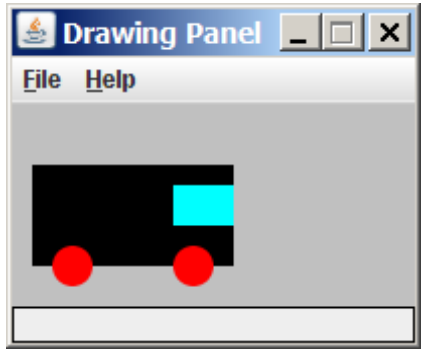
```
panel = DrawingPanel(160, 160)

for i in range(0, 10):
    panel.draw_rect(110 - 10 * i, 20 + 10 * i,
                    10 + 10 * i, 10)
```



Drawing with functions

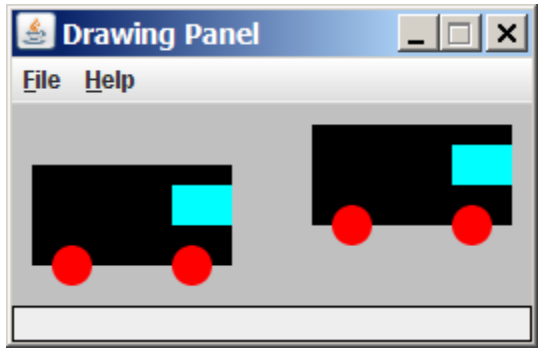
- To draw in multiple functions, you must pass `DrawingPanel`.

```
def main():
    panel = DrawingPanel(200, 100, background="light gray")
    draw_car(panel)

def draw_car(p):
    p.fill_rect(10, 30, 100, 50, "black")

    p.fill_oval(20, 70, 20, 20, "red")
    p.fill_oval(80, 70, 20, 20, "red")

    p.fill_rect(80, 40, 30, 20, "cyan")
```



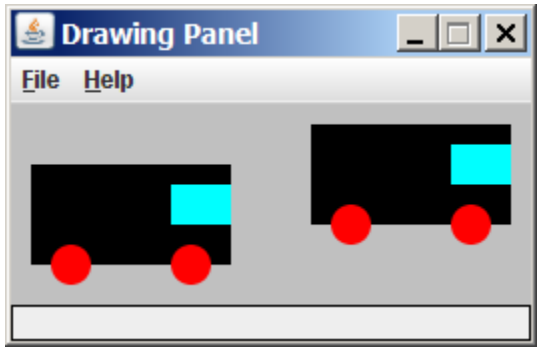
Parameterized figures

- Modify the car-drawing function so that it can draw many cars, such as in the following image.
  - Top-left corners: (10, 30), (150, 10)
  - Hint: We must modify our `draw_car` function to accept x/y coordinates as parameters.



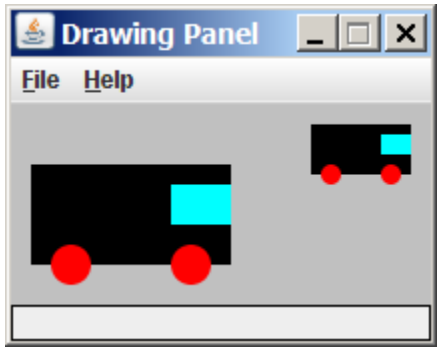
Parameterized answer

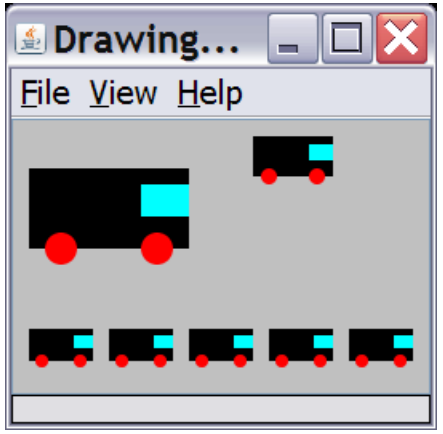
```
def main():
    panel = DrawingPanel(260, 100, background="light gray")
    draw_car(panel, 10, 30)
    draw_car(panel, 150, 10)

def draw_car(p, x, y):
    p.fill_rect(x, y, 100, 50, "black")

    p.fill_oval(x + 10, y + 40, 20, 20, "red")
    p.fill_oval(x + 70, y + 40, 20, 20, "red")

    p.fill_rect(x + 70, y + 10, 30, 20, "cyan")
```



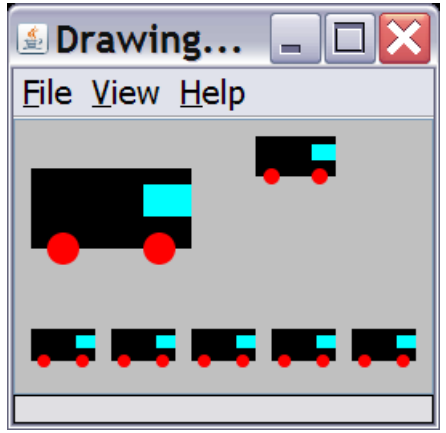
Drawing parameter question

- Modify `draw_car` to allow the car to be drawn at any size.
  - Existing car: size 100. Second car: (150, 10), size 50.
- Once you have this working, use a `for` loop with your function to draw a line of cars, like the picture at right.
  - Start at (10, 130), each size 40, separated by 50px.





Drawing parameter answer




```
def main():
    panel = DrawingPanel(260, 100, background="light gray")
    draw_car(panel, 10, 30, 100)
    draw_car(panel, 150, 10, 50)
    for i in range(0, 5):
        draw_car(panel, 10 + i * 50, 130, 40);

def draw_car(p, x, y, size):
    p.fill_rect(x, y, size, size / 2, "black")

    p.fill_oval(x + size / 10, y + size / 5 * 2, size / 5, size / 5, "red")
    p.fill_oval(x + size / 10 * 7, y + size / 5 * 2, size / 5, size / 5, "red")

    p.fill_rect(x + size / 10 * 7, y + size / 10, size / 10 * 3, size / 5, "cyan")
```

Animation with `sleep`

- `DrawingPanel`'s `sleep` function pauses your program for a given number of milliseconds.

- You can use `sleep` to create simple animations.

```
panel = DrawingPanel(250, 200)
for i in range(1, NUM_CIRCLES + 1):
    panel.draw_oval(15 * i, 15 * i, 30, 30)
    panel.sleep(500)
```

  - Try adding `sleep` commands to loops in past exercises in this chapter and watch the panel draw itself piece by piece.