

15-112 Fall 2014 Quiz 3

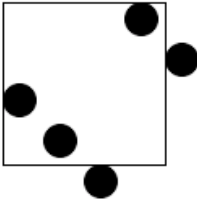
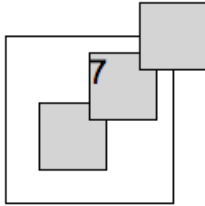
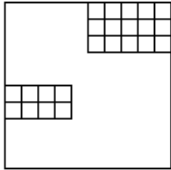
- * 20 minutes. No calculators, no notes, no books, no computers.
- * You may not discuss any portion of this quiz with anyone until after 5pm today.
- * No lists or recursion! * SHOW YOUR WORK, CIRCLE YOUR ANSWERS.

1. **Code Tracing** [20 pts]: This code prints 4 lines with 10 values. 2 values are provided for you. Fill in the missing 8.

<pre>def f(x, y, delta): for z in xrange(x, 0, -3): print "z", z while (y > z): print y, y -= delta delta += 1 print f(9, 8, 1)</pre>	<pre>z _____ _____ _____ z _____ _____</pre>
--	--

2. **Identify the graphics error** [30 pts]

Each problem below includes Python graphics code along with an image of the resulting window. In some cases the image contains an error. For each problem, if there is an error, circle it and briefly describe what is wrong ("missing a square", "this should be gray", etc). If there is no error, write and circle the word "ok" next to the image.

<pre>canvas.create_rectangle(50, 50, 150, 150) y = 100 for x in xrange(50, 150, 25): canvas.create_oval(x, y, x+20, y+20, fill="black") y = (y + 25) if (y < 150) else 50</pre>	
<pre>canvas.create_rectangle(50, 50, 150, 150) counter = 1 while (counter < 10): counter += 3 (cx, cy, r) = (50+10*counter, 150-10*counter, 20) canvas.create_rectangle(cx-r, cy-r, cx+r, cy+r, fill="gray") if (counter % 5 == 2): canvas.create_text(cx-r, cy+r, text=str(counter), anchor=SW, font="Arial 20")</pre>	
<pre>def q(canvas, rows, cols, x, y): for row in xrange(rows): for col in xrange(cols): d = 10 x0 = x + d*col y0 = y + d*row canvas.create_rectangle(x0, y0, x0+d, y0+d) canvas.create_rectangle(50, 50, 150, 150) q(canvas, 3, 5, 100, 50) q(canvas, 2, 4, 50, 100)</pre>	

3. **Free Response: dotAt2oclock** [25 pts]

Say that a circle representing a clock face is drawn on the screen centered at (cx, cy) with radius r. Write a few lines of code that draw a small solid black dot on that circle precisely where the hour hand would point at 2 o'clock. Assume that canvas is already created for you.

4. **Matching Functions** [25 pts]

Each function from f1 through f5 is supposed to be identical to f0 for non-negative integer values of x. However, some of those functions have bugs. For each of function f1 through f5, either identify an argument such that the function differs from f0 at that value of x, or write and circle the word "ok".

```
def f0(x):
    result = 0
    for y in xrange(1,x,2): result += y
    return result
```

```
def f1(x):
    result = 0
    while (x > 0):
        if (x % 2 == 1): result += x
        x -= 1
    return result
```

```
def f2(x):
    result = 0
    for z in xrange(x): result += (z%2)*z
    return result
```

```
def f3(x):
    return sum(xrange(1,x,2))
```

```
def f4(x):
    return sum(xrange(x)) - sum(xrange(0,x,2))
```

```
def f5(x):
    result = 0
    while (x%2 == 0): x -= 1
    while (x >= x%2):
        result += x
        x -= 2
    return result
```

5. **Bonus/Optional:** What will this print? [4 pts]

```
def bonus(x=25): print x, ; return 2*bonus(x/2+bonus(x/4)) if (x>10) else x
print bonus() # hint: this runs without errors
```