Practice Quiz

15-112 Fundamentals of Programming

You Got dis shiz

Code Tracing (20)

# ct1 recursion 	
def ct1(n, d=0):
  def f(s, n, d): return '%s%d' % (s, 10*n+d)
  if (n <= 1):
  return [f('A', n, d)]
  else:
  return ct1(n-1, d+1) + [f('B', n, d)] + ct1(n-2, d+1)
print(ct1(3)) # prints a list of 5 strings	

# ct2() uses class B 	
 	
class B(object):
  def __init__(self, board, row, col, target):
  self.board = board
  self.row = row
  self.col = col
  self.target = target
  def f(self, other):
  target = other.target
  board = self.board
  (rows, cols) = (len(board), len(board[0]))
  for row in range(self.row, rows):
  for col in range(self.col, cols):
  if (board[row][col] == target):
  return (row, col)
  return None
 	
def ct2():
  board = [([0]*10) for row in range(20)]
  (rows, cols) = (len(board), len(board[0]))
  for row in range(rows):	

```
      for col in  range(cols):
          board[row][col] =   row+col
   b1  =   B(board,    8,  4,  0)
   b2  =   B(None, 13, 14, 15)
   return  b1.f(b2)
print(ct2())
```

Reasoning Over Code (15)

```
def rc2Helper(L,    depth=0):

   if  (sum(L) >   500):
      return  (L[-1], depth)
   else:
      L2  =   [val*10 for val in  L]
      return  rc2Helper(L2,   depth+1)

def rc2(L):
   assert(L    ==  sorted(L))
   return  (rc2Helper(L)   ==  (500,   1))
```

Short Answer (15)

1. How are static methods used and why are they useful?

2. How is writing __hash__ in a class useful?

3. What is a class attribute?

Free Response (50)

 # Recursion FR (20)

Write the function KACHOW that takes in a list L and a power p and returns the sum of all the elements of L to the power p. For example if L = [1,2,3] and p = 2, the answer should be 1^2 + 2^2 + 3^2 = 14.

Note : you **cannot** use inbuilt math.pow function - write a **recursive** helper function POW which takes in an integer and a non-negative power and returns the required answer. You can assume that all elements in L are integers and that the power p is a non negative integer.

If you use the math.pow function you will get **no credit**. Yes, you will write a recursive function KAPOW which uses another recursive function POW to get the answer. Godspeed.

#OOPy animation FR (30)

Assuming our run() function is already written, write an animation where each time the user clicks in the canvas, even if that click is inside an existing square, an additional square is added, centered on the click, with a label set to the next available letter. So the first square is labeled "a", the next "b", and so on. After 26 squares are created, future clicks are simply ignored.

Also, when created, each square is assigned a randomly-chosen size, as small as 5x5 up to as large as 20x20. Once created, squares sweep across the screen from left-to-right, and when they go completely offscreen on the right, they reappear on the left side and continue sweeping left-to-right.

Finally, if the user presses the letter corresponding to a square's label, that square increases its velocity by one, so it sweeps faster to the right, unless its velocity reaches 5x normal, in which case it resets to 1x normal.

To do this, you must create a Square class, where each square is an instance of the Square class. Your Square class must include at least a constructor, a moveRight method, and a draw method, all of which must be used properly by your event handlers and redrawAll function. You may add additional methods as you wish.