### Final Sample 1 Solutions

1. While Loop Simulation

|  |  |
| --- | --- |
| Function Call | Output |
| mystery(5, 0)  mystery(3, 2)  mystery(16, 5)  mystery(80, 9)  mystery(1600, 40) | 5  1 0 1  3 2 1 0 1  8 4 2 1 2 0 2  40 19 2 9 0 4 0 |

**2. Inheritance Mystery**

b

c 1

a 2 c 1

b

c 1

b 2 c 2

c

c 1

c 2

b

d 1 b 2 c 2

b 2 c 2

**3. Collections Mystery**

Function Call Contents of Set Returned

-----------------------------------------------

mystery(grid, 2, 2) [6, 7]

mystery(grid, 0, 2) [1, 2, 5, 8]

mystery(grid, 3, 3) [1, 2, 3, 7, 9]

**4. List Programming**

def is\_unique(list):

for i in range(1, len(list)):

for j in range(i, len(list)):

if (list[i - 1] == list[j]):

return False

return True

**5. Collections Programming**

def count\_in\_area\_code(numbers, area\_code):

unique\_numbers = set()

for name, phone in numbers.items():

if (phone[0:3] == area\_code):

unique\_numbers.add(phone)

return len(unique\_numbers)

**6. Programming**

def same\_pattern(s1, s2):

if (len(s1) != len(s2)):

return False

for i in range(0, len(s1)):

for j in range(i + 1, len(s1)):

if (s1[i] == s1[j] and s2[i] != s2[j]):

return False

if (s2[i] == s2[j] and s1[i] != s1[j]):

return False

return True

**7. 2d Lists**

def find\_max(lis):

max\_sum = 0

max\_row = 0

for i in range(0, len(lis)):

cur\_sum = 0

cur\_row = i

for j in range(0, len(lis[i])):

cur\_sum += lis[i][j]

if cur\_sum > max\_sum:

max\_sum = cur\_sum

max\_row = cur\_row

return max\_row

**8. Critters**

class Ostrich(Critter):

def \_\_init\_\_(self):

super(Ostrich, self).\_\_init\_\_()

self.\_\_hiding = True

self.\_\_steps = 0

self.\_\_west = randint(0, 1) == 0

def get\_color(self):

if (self.\_\_hiding):

return "cyan"

else:

return "white"

def get\_move(self):

if (self.\_\_steps == 10):

self.\_\_steps = 0 # Pick a new direction and re-set the steps counter

self.\_\_hiding = not self.\_\_hiding

self.\_\_west = randint(0, 1) == 0

self.\_\_steps += 1

if (self.\_\_hiding):

return DIRECTION\_CENTER

elif (self.\_\_west):

return DIRECTION\_WEST

else:

return DIRECTION\_EAST

**9. Classes**

def compare(other):

if (self.\_\_month < other.\_\_month or (self.\_\_month == other.\_\_month and

self.\_\_day < other.\_\_day)):

return -1

elif (self.\_\_month == other.\_\_month and self.\_\_day == other.\_\_day):

return 0

else:

return 1