Review
Multiple choice quiz

CS 1MD3 • Introduction to Programming
Winter 2018

Dr. Douglas Stebila

McMaster University

Question 1

What is the value of e after following operations?

```
a = [1, 2, 3]
b = a
c = [4, 5, b]
a[0] = c[1]
d = c[2]
e = d[0]
```

A. 1
B. 2
C. 3
D. 4
E. 5

a = [5,2,3]
b = a
c = [4,5,a]
d = c[2]
e = d[0]

Question 2

A sequence is **ascending** if every element is less than or equal to the next one. For example, [1, 4, 4, 5] but [1, 3, 2] is not. Which of the following returns True if s is ascending?

```
def ascendingA(s):
    for i in range(len(s)):
        if s[i] > s[i+1]:
            return False
    return True

def ascendingB(s):
    for i in range(len(s)):
        if s[i-1] > s[i]:
            return False
    return True

def ascendingC(s):
    for i in range(len(s)-1):
        if s[i-1] > s[i]:
            return False
    return True
```

```
def ascendingD(s):
    for i in range(1, len(s)):
        if s[i-1] > s[i]:
            return False
    return True

def ascendingE(s):
    for i in range(1, len(s)-1):
        if s[i] > s[i+1]:
            return False
    return True
```

Question 3

- The function allUnique(s) should return True if all characters of string s (or elements of list s) are unique, i.e. occur only once. For example, allUnique(''), allUnique('a'), allUnique('ab') are all True, but allUnique('aa') is False. Which of the following is **not** correct?

```
def allUniqueA(l):
    s = set()
    for e in l: s = s|{e}
    return len(l) == len(s)

def allUniqueB(l):
    s = set()
    for e in l: s.add(e)
    return len(l) == len(s)

def allUniqueC(l):
    return len(l)==len(set(l))
```

```
def allUniqueD(l):
    for i in range(len(l)):
        if l[i] not in l[:i]:
            return True
    return False

def allUniqueE(l):
    for i in range(len(l)):
        if l[i] in l[:i]:
            return False
    return True
```

Question 4

A **whole floating point number** is one like `3.0`, with `0` after the dot.

Which of the following is true in Python?

A. Any integer number can be exactly represented as a floating point number and any whole floating point number can be represented as an integer number

B. Any integer number can be exactly represented as a floating point number but not all whole floating point numbers can be represented as integer numbers

C. Not all integer numbers can be exactly represented as floating point numbers but any whole floating point number can be represented as an integer number

D. Not all integer numbers can be exactly represented as floating point numbers and not all whole floating point numbers can be represented as integer numbers

E. None of the above

Question 5

The algorithm for computing change is **greedy** because

A. It raises an exception if it cannot compute the exact change

B. It does not use recursion

C. It picks the most coins of one denomination at a time

D. It takes only as many steps as there are denominations

E. None of the above

Question 6

- How many steps does this program take to compute x**y? That is, how often is the body of the loop executed? Give an expression in terms of x and y!

```
def exp(x, y):
    r = 1
    while y > 0:
        r, y = r * x, y - 1
    return r
```

Question 7

```
def f(x, y):
  return x if y == 0 else f(x, y - 1) + 1
```

What does f(x, y) compute for integers x, y ≥ 0?

A. x + x
B. x * x
C. (x + 2) * (y – 1)
D. x ** y
E. None of the above

Question 8

- Pick the word below that best matches **exception**:

A. recursion

B. control flow

C. function

D. annotation

E. iteration

Question 9

Which of the following is true about **objects** in Python?

A. Neither functions nor integers are objects

B. Functions are not objects but integers are objects

C. Functions are objects but integers are not objects

D. Both functions and integers are objects

E. None of the above

Question 10

```
class Point:
    def __init__(self, x0, y0):
        self.x, self.y = x0, y0
    def set(self, x, y):
        self.x, self.y = x, y

class ColoredPoint(Point):
    def __init__(self, x0, y0, color):
        self.x, self.y = x0, y0
        self.color = color

c = ColoredPoint(1, 2) # A
c.set(3, 4)            # B
```

A. There is a syntax error in above code
B. The code will raise an exception when run because line A fails
C. The code will raise an exception when run because line B fails
D. The code will run and set `c.x` to `3` and `c.y` to `4`
E. None of the above