

## CS1026 Computer Science Fundamentals Quiz 1

For the following questions, choose the best answer from the options below.

- 1) If you entered the following in Python,

```
"M" < "Z"
```

what would the result be?

- a) **True**
  - b) False
  - c) Yes
  - d) Error, you cannot do math with letters
- 2) Which of the following makes the expression not x <= y and z > x evaluate to True?
- a) x = 10, y = 15, z = 10
  - b) x = 15, y = 20, z = 10
  - c) x = 20, y = 20, z = 10
  - d) **x = 15, y = 10, z = 20**
  - e) None of the above.
- 3) Which of the following is equivalent to  $4+12/(7-1)*2$ ?
- a)  $(4+12)/7 - (1 * 2)$
  - b)  $4+12/(7-1 * 2)$
  - c)  $(4+12)/7 - 1 * 2$
  - d)  $4+12/7 - 1 * 2$
  - e) **None of the above**
- 4) What is  $13//4$ ? – NOTE ... the choices on the handout were incorrect ... this is the correct choices (and answers)
- a) 3.0
  - b) 3.25
  - c) **3**
  - d) 4
  - e) Syntax error.
- 5) What output is generated by the following code snippet?
- ```
firstName = "John"
middleName = "Clark".upper()
lastName = "Thomas"
print(firstName[1], middleName[len(middleName)//2], lastName[5])
```
- a) J a s
  - b) **o A s**
  - c) J A A
  - d) j A s
  - e) o L s
- 6) What does the following code snippet print if the input is “elephant”?
- ```
position = 0
str = input("Enter a string: ")
while position < len(str) and str[position] != 'e' :
    position = position + 1
print(position)
```
- a) **0**
  - b) 1
  - c) 2
  - d) 3
  - e) 4

- 7) The following function is supposed to find and return the sum of odd numbers between **low** and **high** inclusive. The function has lines that contain 3 logic errors. Identify the logic errors and correct the lines. Note that there may be more than one logic error on a line.

```
def addOne(low, high) :
    sum = 0
    for i in range(low,high) :
        if i % 3 == 0 :
            sum = sum + i
    return sum
```

**Solution:**

```
def addOne(low, high) :
    sum = 0
    for i in range(low,high+1) : # Error: changed high+1 to high
        if i % 2 == 1 : # Error: changed 2 to 3; AND changed the 1 to a 0
            sum = sum + i
    return sum
```

- 8) The following function is supposed to remove vowels from a string (assume all is lower case) and return a string. The function has lines with 4 logic errors. Identify the logic errors and correct the lines. Note that there may be more than one logic error on a line.

```
VOWELS = "aeiou"
def removeVowels (s1) :
    sr = ""
    for i in range(1,len(s1)-1) :
        if s1[i] in VOWELS:
            sr = sr + s1[i]
    return s1
```

**Solution:**

```
def removeVowels(s1):
    sr = ""
    for i in range(0,len(s1)): # Error: changed 0 to 1; AND changed len(s1) to len(s1)-1
        if s1[i] not in VOWELS: # Error: changed "not in" to "in"
            sr = sr + s1[i]
    return sr # Error: changed sr to s1
```

- 9) The following function is supposed to replace an integer in the first list (lst1) by the integer in the same position of the second (lst2) **IF** the integer in the second list is larger. The function has lines with 4 logic errors. Identify the logic errors and correct the lines. Note that there may be more than one logic error on a line.

```
def swapLowHigh (lst1,lst2) :
    for i in range(1,len(lst2)) :
        if lst1[i] < lst2[i] and I < len(lst2):
            tmp = lst1[i]
            lst2[i] = lst1[i]
            lst1[i] = tmp
```

**Solution:**

```
def swapLowHigh(lst1,lst2):
    for i in range(0,len(lst1)): # Error: changed 0 to 1
        if i < len(lst2) and lst1[i] < lst2[i]:
            # Error in above line: changed condition
            # to: lst1[i] < lst2[i] and i < len(lst2)
            # - this will fail with index out of bounds
            tmp = lst1[i]
            lst1[i] = lst2[i] # Error: changed lst1[i] = lst2[i] to lst2[i] = lst1[i]
            lst2[i] = tmp # Error: changed lst2[i] = tmp to to lst1[i] = tmp
```

- 10) Write a function that will count the multiples of a given integer in a range of numbers. The header for the function is given below. The function should return the number of multiples of n between low and high, inclusively. For example, if the input is (2,12,4), then the result is 3, since there are 3 multiples of 4 (4, 8 and 12) in the range.

```
def countMultiples(low,high,n):
    # your code goes here
```

```
def countMultiples(low,high,n):
    c = 0
    for i in range(low,high+1):
        if i % n == 0:
            c += 1
    return c
```

- 11) Write a function shuffle(lst1,lst2) that will create a new list which contains the elements of lst1 mixed with the elements of lst2 by alternating the elements from the lists. For example, shuffle([0,0,0],[1,1,1]) would produce [0,1,0,1,0,1]. If the lists are not the same length, then an empty list should be returned.

```
def shuffle(lst1,lst2):
    alst = []
    if len(lst1) == len(lst2):
        for i in range(0,len(lst1)):
            alst.append(lst1[i])
            alst.append(lst2[i])
    return alst
```