NWEN 241

Python3 assignment (/50 marks) Due on: 7th June 2017, 23:59

Q1. [1 mark each] For each entry in Table 1, define the python3 data type.

No.	Data	Type
1	123	<class 'int'>
2	"this is a great course"	<class 'str'>
3	24.75	<class 'float'>
4	["try","me","out"]	<class 'list'>
5	{2, 'a', 3, "c"}	<class 'set'>

Q2.

(a) [3 marks] Write a python3 program that reads two strings as parameters on the command line and swaps the position of the two strings and prints them both. For example, it might be called by:

$ python3 swapstring.py my_str1 mystr2

```
import sys
if len(sys.argv) >= 2:
        s1 = sys.argv[1] #get the first string
        s2 = sys.argv[2] #get the second string
        print(s2+" "+s1)
else:
        print("you should at least input two string")
```

(b) [2 marks] What will be the output after the following Python3 snippet is executed?

```
str_of_words = ['Games' , "people", 'play']
str_of_new_words = 'The' + str_of_words[-1] + str_of_words[1] + str_of_words[0]
print(str_of_new_words)
```

TheplaypeopleGames

Q3.

(a) [2 marks] What will be the output after the following Python3 snippet is executed?

```
temp_list = []
for n in range(6):
        temp_list.append(n*3)

print(temp_list)
```

[0, 3, 6, 9, 12, 15]

(b) [3 marks] list three immutable data types and three mutable data types in python3.

immutable types: int, float, long, str, tuple

mutable types: list, dict, set

Q4.
(a) [2 marks] What does the python3 statement import os.path do ?
Import os.path as a module to be used

(b) [3 marks] Differentiate the effect of the statement import os and import os.path ?

Import os -→ import all the modules under the os. （more modules than under os.path）
Import os.path → import all the modules under the os. (less modules than under os.path）

Q5.
Functions are used to wrap/encapsulate a set of tasks for increased modularity and reusability. A function takes on arguments.
(a) [4 marks] Write an example of a function definition in python3 (no more than four lines).

```
def myPrint( str ):
        print (str)
        return
```

(b) [4 marks] Define positional arguments and keyword arguments in the python3.

keyword arguments: Call a function with parameters by passing arguments preceded by an identifier (e.g. `name=`) which should be same as the name of parameter in a function. Or passed as a value in a dictionary preceded by `**`.

Positional arguments: Call a function with parameters by passing arguments without preceding by an identifier (e.g. `name=`) in a function, or be passed as elements of an interable preceded by *.

(c) [2 marks] Give an example of calling a function via positional arguments and keyword arguments.

```
def foo(a,b):
        print (a+b)
        return
```

Example via keywords arguments,

```
foo (a = 1, b =2)
foo (**{'a':1, 'b':2})
```

Example via positional arguments,

```
foo (1, 2)
foo (*(1, 2))
```

Q6. [10 marks] Say you have three lists called lst1, lst2 and lst3. When you interleave three list perfectly, the first element would come from lst1, the second element from lst2 and the third element from lst3 (or any permutation of such). This pattern of three elements drawn from three different lists repeats until all elements in lst1, lst2 and lst3 are exhausted.

lst1	lst2	lst3	def perfect_interleave (lst1,lst2,lst3)
[4,6,8]	[3,5,7]	[11,12,13]	[4,3,11,6,5,12,8,7,13] or [8,7,13,6,5,12,4,3,11]
[a,b,c]	[d,e,f]	[x,y,z]	[a,d,x,b,e,y,c,f,z]

(a) [5 marks] Write the pseudocode or flowchart for perfect_interleave(lst1,lst2,lst3)

define the perfect_interleave(lst1, lst2, lst3) method

(b) [5 marks]Write a python3 **recursive** function that implements perfect_interleave(lst1,lst2,lst3)

```
def perfect_interleave(lst1,lst2,lst3):
        if len(lst1) != len(lst2) or len(lst1) != len(lst3):
                raise Exception('Should input same size list')
        elif len(lst1) == 0 or len(lst2) == 0 or len(lst3) == 0 :
                return []
        else:
                return [lst1[0],lst2[0],lst3[0]] + perfect_interleave(lst1[1:],lst2[1:],lst3[1:])
```

Q7.
(a) [2 marks] Find the error in the following python3 program.

```
line = raw_input("Type a word")
print "You typed", line
line = line + "h"
num = int(line)
print "You typed the number ", num
```

line = raw_input("Type a word") → should change to-→ line =input("Type a word")
because raw_input("something") is a method of python2 program

print "You typed", line → should change to-→ print ("You typed", line)
because print "" without **parenthesis** is a method of python2 program

num = int(line) does not work, because int()method cannot cast "non-number string" into a int type

print "You typed", line → should change to-→ print ("You typed the number %s" %num)
because print "" without **parenthesis** is a method of python2 program

(b) [3 marks] Explain the difference between a syntax error and semantic error in the context of computer programming languages. (Note that this is not python specific)

Syntax error was occurred when there was a piece of code breaking the rule of the kind of programming language.
Semantics error was occurred when there was a piece of code which does not implement the intended functionality, even though there is no syntax errors

Q8.
(a) [3 marks] Assuming num = 20, determine the value of each of the following Python expressions:

(i) num / 12
(ii) 123 % 100
(iii) 8 + 3 * 7
(iv) (0 == 1) and (2 < 3)
(v) not ((4.5 < 12.9) and (6 * 2 <= 13))
(vi) (0 ==1) or (2 < 3)

(i) 1.6666666666666667
(ii) 23
(iii) 29
(iv) False
(v) not ((4.5 < 12.9) and (6 * 2 <= 13))
(vi) True

(b) [2 marks] Consider the following exception:

TypeError: can only concatenate tuple (not "int") to tuple

Which of the following python3 snippets will throw this exception?
(A) tuple("LAN")+len("DO")
(B) tuple("LAN")[len("DO")]
(C) tuple("LAN")+tuple("DO")
(D) None of the other answers are correct
(E) "LAN"+[tuple("DO")]

A and E