

Lesson 05 – Iterations o Concepts: increment, decrement, iterations, test / exit conditions, loop control variables, definite loop, indefinite loop, iterator i.e. The range() function. Counting starts at zero for all sequence types (ranges, strings, lists).

o When to use a definite versus an indefinite loop, infinite loop pattern (While True w/break).

o Python: for, while, break , continue, range()

1. What is wrong with this Python loop:

```
n = 5
while n > 0:
    print n
print 'All done'
```

Options:

- a) This loop will run forever
 - b) There should be no colon on the while statement
 - c) The print 'All done' statement should be indented
 - d) while is not a Python reserved word
2. What does the break statement do?
- a) Jumps to the "top" of the loop and starts the next iteration
 - b) Resets the iteration variable to its initial value
 - c) Exits the program
 - d) Exits the currently executing loop
 - e) Both c & d
3. What does the continue statement do?
- a) Exits the currently executing loop
 - b) Jumps to the "top" of the loop and starts the next iteration
 - c) Exits the program
 - d) Resets the iteration variable to its initial value
4. What does the following Python program print out?
- ```
tot = 0
for i in [5, 4, 3, 2, 1] :
 tot = tot + 1
print tot
```
- a) 10 b) 5 c) 0 d) 15 e) 16

5. What is the iteration variable in the following Python code?

```
friends = ['Joseph', 'Glenn', 'Sally']
for friend in friends :
 print('Happy New Year:', friend)
print('Done!')
```

a)friends b)friend c)Glenn d)Joseph e) Joseph Glenn Sally

6. What will the following code print out?

```
smallest_so_far = -1
for the_num in [9, 41, 12, 3, 74, 15]:
 if the_num < smallest_so_far :
 smallest_so_far = the_num
print(smallest_so_far)
```

A) 3 B) -1 C) 9 D)74 E) ERROR

7.

```
i = 1
while True:
 if i%2 == 0:
 break
 print(i)
 i += 2
```

a)1  
b)12  
c)123456 .....  
d) 1 3 5 7 9 11 .....  
e) 2 4 6 8 10 12 .....

8.

```
i = 1
while True:
 if i%7 == 0:
 break
 print(i)
 i += 1
```

- a)123456
- b)1234567
- c)error
- d) none of the mentioned

Lesson 06 – Functions o Concepts: function call, function definition, argument, parameters, return values, module file, module object, fruitful function. Difference between a function and a method function. Functions as programs with inputs and output.

o Python: import, def, return, dir(), help()

o Why create functions? What are the benefits? Variables in the global scope versus the local scope.

9. Which Python keyword indicates the start of a function definition?

- a) break b) def c) return d) sweet

10. In Python, how do you indicate the end of the block of code that makes up the function?

- a) You put the colon character (:) in the first column of a line
- b) You de-indent a line of code to the same indent level as the def keyword
- c) You put a # character at the end of the last line of the function
- d) You add the matching curly brace that was used to start the function }

11. What does the following code print out?

```
def thing():
 print('Hello')
print('There')
```

- 1.Hello
- 2.There
- 3.Error
- 4.Hello There

12. In the following Python code, which of the following is an "argument" to a function?

```
x = 'banana'
y = max(x) |
print(y)
print(x)
```

- 1.y
- 2.banana
- 3.max
- 4.x
- 5.n banana

13. Which line of code is useless?

```
def stuff():
 print('Hello')
 return
 print('World')
stuff()|
```

- a) print 'Hello'
- b) def stuff():
- c) stuff()
- d) print 'World'
- e) return

14.

```
def greet(lang):
 if lang == 'es':
 return 'Hola'
 elif lang == 'fr':
 return 'Bonjour'
 else: return 'Hello'
print(greet('fr'),'Michael')
```

- a) Hola Michael
- b) def Michael
- c) Hola Bonjour Hello
- d) Bonjour Michael
- e) fr Michael

15.

```
def addtwo(a, b):
 added = a + b
 return a
x = addtwo(2, 7)
print(x)
```

- a)error b)9 c)2 d)7



Lesson 07 - Strings o Concepts: Strings are immutable sequences of characters. Zero based index. String tokenization and parsing.

Slice notation: There will be a few questions on slice notation. Learn it. Examples: x[:5], x[5:], x[1:4] x[3:-2], x[2]

Python: Know these string functions: len(), upper(), lower(), find(), count(), replace(), startswith(), endswith(), strip(), split()

16.

```
for i in x:
 x.append(i.upper())
print(x)
```

- a) ['AB', 'CD'].
- b) ['ab', 'cd', 'AB', 'CD'].
- c) ['ab', 'cd'].
- d) none of the mentioned
- e) infinite loop

17.

```
x = ['ab', 'cd']
for i in x:
 i.upper()
```

- a) ['ab', 'cd'].
- b) ['AB', 'CD'].
- c) [None, None].
- d) none of the mentioned

18. `print("xyyzxyzxxyy".endswith("xyy"))`

- a) 1
- b) True
- c) 3
- d) 9

19 What will variable y output?

x= 'Till'

y= x.upper().replace("l", "O")

A) TILL B) Toll C) TOLL D) ERROR E) TOLL

20. What is the output for the following:

```
text = 'Faith is laughing loudly'
for word in text.split(): |
 print(word[0])
```

A)Faith B)F C)Fill D)loudly

### **#Concept Revision: Tokenization**

*Tokenization is the process of breaking up a string into words, phrases, or symbols.*

- Tokenize a sentence into words.
- "mike is here" -> ['mike','is','here']

21.

```
data = 'From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008'
atpos = data.find('@')|
sppos = data.find(' ',atpos)
host = data[atpos+1:sppos]
print(host)
```

- A) uct.ac.za
- B)31
- C)@ uct.ac.za
- D)uct.ac.za Sat

### **#Concept Revision: Parsing**

Lesson 08 - Files o Concepts: files are a persistence layer, open file, file handle, reading, writing, file paths, file exception handling

o Python: with open(), read(), write(), close().

o Python code to: read from a file a line at a time, read from a file all at once, write to a file.

22.

```
f = open('test.txt', 'r', encoding = 'utf-8')
f.read()
```

- A. This program reads the content of test.txt file.
- B. If test.txt contains newline, read() will return newline as '\n'.
- C. You can pass an integer to the read() method.
- D. All of the above.

23. To open a file c:\scores.txt for writing, we use

- a) outfile = open("c:\scores.txt", "w")
- b) outfile = open("c:\\scores.txt", "w")

- c) `outfile = open(file = "c:\scores.txt", "w")`
- d) `outfile = open(file = "c:\\scores.txt", "w")`

24. The `readlines()` method returns

- a) `str`
- b) a list of lines
- c) a list of single characters
- d) a list of integers

25. What will happen if you try to open a file that doesn't exist?

- A) A new file is created.
- B) Nothing will happen.
- C) An exception is raised.
- D) None of the above.

27. Which of the following statements are true?

- a) When you open a file for reading, if the file does not exist, an error occurs
- b) When you open a file for writing, if the file does not exist, a new file is created
- c) When you open a file for writing, if the file exists, the existing file is overwritten with the new file
- d) All of the mentioned

28. Which of the following codes closes file automatically if exception occurs?

A.

```
with open("test.txt", encoding = 'utf-8') as f:
 # perform file operation
```

B.

```
try: f = open("test.txt", encoding = 'utf-8') # perform file operations
finally:
 f.close()
```

C.

None of the above

D.

Both of the above

Lesson 09 – Lists – o Lists will be emphasized on the next exam.

o For this exam, simply know what a Python list is, what it looks like in Python. and how it is different from a string



***How is list different from a String? //Open Question to all***

### SELF STUDY - PRACTICE QUESTIONS

1. What is a good description of the following bit of Python code?

```
zork = 0
for thing in [9, 41, 12, 3, 74, 15] :
 zork = zork + thing
print('After', zork)
```

- A) Count all of the elements in a list
- B) Sum all the elements of a list
- C) Find the largest item in a list
- D) Find the smallest item in a list

2. Which reserved word indicates the start of an "indefinite" loop in Python?

a) indef b) infinite d) while e)break f) none

3. How many times will the body of the following loop be executed?

```
n = 0
while n > 0 :
 print('Lather')
 print('Rinse')
print('Dry off!')
```

- a) 0 b) 1 c) infinite times d) will cause an error e) 5

4. What will the following Python code print out?

```
def func(x):
 print(x)
func(10)
func(20)
```

- a) 10   b)x   c)x   d) error  
20   20   x

5. 'Hello'.replace('l','e')

- a) Heeeo  
b) Heelo  
c) Heleo  
d) None

**References:**

**Sanfoundry.com**

**Google.com**

**Prof. Fudge's Study Guide**

**Python For Everybody**