Python

Module 1 Lesson 12

Name: Date:

Instructions:

1. Students are given 1 hour to complete this test.
2. For the duration of the test, teachers are not allowed to help the students with the answer.
3. Students are to score at least 70% on the test to pass. If they fail, they will have to redo the test again in the next lesson.

Section A – MCQ	/ 10
Section B – Debugging	/ 10
Section C – Short Coding Question	/ 10
Section D – Open ended Question	/ 20
	/ 50

Python

Module 1 Lesson 12

Name: Date:

Section A: (10 marks)

This is a multiple-choice answer section. Write your answer is the bottom right of each question.

Question 1:

What would the output of the following code be?

Code	
1	X = 'Elephant'
2	print(X[::-1])

A) Elephant

B) tnahpelE

C) tnahpel

D) nahpelE

Question 2:

What is the final value of count?

Code	
1	count = 0
2	def welcome(name):
3	print("Hello", name)
4	global count
5	count += 2
6	welcome("Thomas")
7	welcome("Leon")

A) 1

B) 2

C) 3

D) 4

Name: Date:

Section A: (10 marks)

Question 3:

How many lines of output will there be for the following code?

Code	
1	def funct(x):
2	print(x*2)
3	
4	print(funct('Hello'))

A) 1 line

B) 2 lines

C) 3 lines

D) No output

Question 4:

What is the difference between return and print?

A) Return changes the value of a function, while print creates an output

B) Return creates an output, while print changes the value of a function

C) Both change the value of a function

D) Both create an output

Name: Date:

Section A: (10 marks)

Question 5:

What does X represent in the following function?

Code	
1	X = 5
2	def add(a,b):
3	Y = a + b
4	global X
5	X += 1
6	add(1,2)

A) Local Variable

B) Argument

C) Parameter

D) Global Variable

Question 6:

For function parameters, which kind of brackets do we use?

A) Round Brackets “( )”

B) Square Brackets “[ ]”

C) Curly Brackets “{ }”

D) Angle Brackets “< >”

Name: Date:

Section A: (10 marks)

Question 7:

What is the difference between an argument and a parameter?

A) An argument must only be an integer, while a parameter can have any datatype.

B) An argument can be called outside a function, while a parameter cannot.

C) An argument has no name restrictions, while a parameter can only be 1 character long.

D) An argument is from the calling side, while a Parameter is from the Function side.

Question 8:

What is the difference between % and //?

A) % refers to floor division while // refers to modulus

B) // refers to floor division while % refers to modulus

C) // gives me the remainder while % gives me the quotient

D) // and % are the same operator

Name: Date:

Section A: (10 marks)

Question 9:

How many arguments can there be in a function?

A) 1

B) 2

C) 5

D) Infinite

Question 10:

How many types of arguments are there?

A) One, Required Arguments

B) Two, Required Arguments and Keyword Arguments

C) Three, Required Arguments, Keyword Arguments and Default Arguments

D) Four, Required Arguments, Keyword Arguments, Default Arguments and Important Arguments

Name: Date:

Section B: (10 marks)

This is the debugging section. In the next few questions, there are **bugs in the code giving an incorrect output**. **The scenarios are shown in each question. Read the requirements carefully.**

Identify the bugs and correct them in the table on the right. Each correction is worth 2 marks.

Question 11: (4 marks)

The function *add()* is supposed to take in 2 arguments *x & y* **and print** the sum of the 2 arguments *x + y*.

Find the **2 mistakes** and correct them.

add(1,2) should print 3

add(2,3) should print 5

Faulty Code	
1	create add(x,y):
2	ans = x, y
3	print(ans)
4	
5	add(1, 2)
6	add(2, 3)

Corrected Code	
1	
2	
3	
4	
5	
6	

Question 12: (6 marks)

The function *minus()* is supposed to take in 2 arguments *x & y*, **return** *x -y*

Find the **3 mistakes** and correct them.

minus(3,2) should **return** 1

minus(5,3) should **return** 2

Faulty Code	
1	def minus(x,y)
2	ans = x, y
3	print(ans)
4	
5	print(minus(3, 2))
6	print(minus(5, 3))

Corrected Code	
1	
2	
3	
4	
5	
6	

Name: Date:

Section C: (10 marks)

This section is a short coding question section. Write the python function as stated in the questions.

Each function is worth 5 marks.

Question 13: (5 marks)

James was given 4 bags of cherries. He noticed that in each bag, there were some ripe and unripe cherries mixed inside. He counted the number of ripe cherries and wrote his findings below. He also recorded down the total number of cherries per bag below.

Write a **python function** that prints the **number** of unripe cherries in each bag.

Bag A: 25 ripe cherries out of 100

Bag B: 29 ripe cherries out of 89

Bag C: 87 ripe cherries out of 92

Bag D: 67 ripe cherries out of 102

The function should be called *getUnripeCherries()* with the parameters – *name, ripe, total*

Sample Function Calls	**Sample Output**
getUnripeCherries('A', ripe=25, total=100)	Bag **A** has **75** unripe cherries
getUnripeCherries('B', ripe=29, total=89)	Bag **B** has **60** unripe cherries
getUnripeCherries('C', ripe=87, total=92)	Bag **C** has **5** unripe cherries
getUnripeCherries('D', ripe=67, total=102)	Bag **D** has **35** unripe cherries

Name: Date:

Section C: (10 marks)

Question 14: (5 marks)

Tom is given the assignment to write a python function to add numbers together. However, the numbers may not always come in 5. Sometimes, there will only be 2 numbers.

Write a **python function** that prints the **sum** of the given numbers. You may assume that there will at least be 2 numbers and at most 5 numbers.

Case A: 1 + 1 + 1 + 1 + 1 = 5

Case B: 1.2 + 2.2 + 3.2 + 4.2 + 5.2 = 16.0

Case C: 100 + 25 = 125

The function should be called *getSum()* with the parameters – *name, a, b, c, d, e*

Sample Function Calls	**Sample Output**
getSum('A', 1, 1, 1, 1, 1)	Case **A** has the sum of **5**
getSum('B', 1.2, 2.2, 3.2, 4.2, 5.2)	Case **B** has the sum of **16.0**
getSum('C', 100, 25)	Case **C** has the sum of **125**

Name: Date:

Section D: (20 marks)

This section is a long coding question section.

Marks are allocated in the question.

Question 15: (20 marks)

Farmer Jones grows 3 crops: Carrots, Corns and Turnips.

He organizes each crop in groups of 2 tonnes, 5 tonnes and 10 tonnes respectively (1 set of carrots = 2 tonnes of carrots. 1 set of corn = 5 tonnes of corns. 1 set of turnip = 10 tonnes of turnips).

Farmer Jones tracks the number of crops he grows each year. He tabulates the data as shown below. He sells to the same market at the price of $10,000 per set of crops.

Year	Carrots (tonnes)	Corns (tonnes)	Turnips (tonnes)
2008	297	345	219
2009	303	357	243
2010	289	332	276

Write a **python code** to calculate the following. **You do not need to write a function.**

(i) How many sets of carrots, corns, turnips – 10 marks
(ii) Total number of tonnes of carrots, corns, turnips leftover – 5 marks
(iii) How much he earns in the given year – 5 marks

Sample Inputs	Sample Output
year = 2008 carrots = 297 corns = 345 turnips = 219	There are **148** sets of carrots, **69** sets of corns and **21** sets of turnips The total number of tonnes leftover is **10** In year **2008**, Farmer Jones will earn $**2380000**
year = 2009 carrots = 303 corns = 357 turnips = 243	There are **151** sets of carrots, **71** sets of corns and **24** sets of turnips The total number of tonnes leftover is **6** In year **2009**, Farmer Jones will earn $**2460000**
year = 2010 carrots = 289 corns = 332 turnips = 276	There are **144** sets of carrots, **66** sets of corns and **27** sets of turnips The total number of tonnes leftover is 9 In year **2010**, Farmer Jones will earn $**2370000**