Roll No. | | | | |

Internal Assessment Test 1 – July 2023

Sub:	Introduction to Python **Programming-Solutions and Scheme**					Sub Code:	BPLCK205B	Branch:	Chemistry Cycle
Date:	06-07-2023	Duration:	90 min's	Max Marks:	50	Sem / Sec:	II / Chemistry Cycle		OBE

	Answer any FIVE FULL QUESTIONS	MARKS	CO	RBT
1 (a)	Briefly discuss about different python data types with syntax and example ● Description of the same **3 Marks** ● Syntax and correct explanation **2 Marks** The integer (or int) data type indicates values that are whole numbers. Numbers with a decimal point, such as 3.14, are called floating-point numbers (or floats). Python programs can also have text values called strings, or strs. **Table 1-2:** Common Data Types Integers: -2, -1, 0, 1, 2, 3, 4, 5 Floating-point numbers: -1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25 Strings: 'a', 'aa', 'aaa', 'Hello!', '11 cats' Booleans that are having two different values i.e. True, False. Truth values that represent Yes/No. It is immutable. Lists data type which is represented as [1,2,3,4,5].It is a collection of data, sits between [ ] .It is mutable. Tuples data type which is represented as (1,2,3,4,5).It is a collection of data, sits between ( ). It is Immutable. Dictionaries are represented as {"a":1, "b":2, "c":3} which is a collection of data, sits between { } It is mutable.	[5]	CO1	L2
(b)	Write a Python program to find the largest of 'n' numbers inputted. ● Correct looping structure **2 Marks** ● Correct Equations **1 Marks** ● Correct Syntax **2 Marks** i = 0 N = int(input('How many numbers do you want to enter?: ')) n_maximum = int(input('Insert the first number: ')) while i < N-1: n=int(input('Insert a number: '))	[5]	CO1	L3

	if n > n_maximum:
	n_maximum = n;
	i += 1
	print('The maximum value is: ', n_maximum)
	
	Output:
	How many numbers do you want to enter?: 10
	Insert the first number: 12
	Insert a number: 34
	The maximum value is: 34
	Insert a number: 56
	The maximum value is: 56
	Insert a number: 78
	The maximum value is: 78			

2 (a)	"Consider below expressions and identify the type of errors that occur. Justify Your Answer. ● Description of the same **1 Marks** ● Syntax and correct explanation **1 Marks each(1*4 =4Marks)** a. >>> 45.30+ b. >>>'SPAM' + 983654 c. >>> 'SPAM'*'BACCON' d >>>'SPAM' * 7 **Output:** syntax error, incomplete input. b. >>>'SPAM' + 983654 TypeError: can only concatenate str (not "int") to str c. 'SPAM'*'BACCON' TypeError: can't multiply sequence by non-int of type 'str' d. 'SPAM' * 7	[5]	CO1	L3

	No error.It will print 'SPAM' seven times. 'SPAMSPAMSPAMSPAMSPAMSPAMSPAM'			

(b)	What is a modulo operator? Give an example. How would you implement modulo operation without using the modulo operator?
	
	● Correct looping structure **2 Marks**
	● Correct Equations **2 Marks**
	● Correct Syntax **2 Marks**
	
	The modulo operator, like the other arithmetic operators, can be used with the numeric types int and float. Basically, the Python modulo operation is used to get the remainder of a division.
	
	Example:
	
	>>> 17 % 12
	
	5
	
	>>> 12.5 % 5.5
	
	Implement Modulo operation without using modulo operator
	
	def getRemainder(num, divisor):
	
	return (num - divisor * (num // divisor))
	
	num = 50
	
	divisor = 7
	
	print(getRemainder(num, divisor))
	
	output:
	
	1
		[6]	CO1	L3

3 (a)	Explain the use of range() function with example code snippets ● Description of the same **2 Marks** ● Code and correct explanation **2 Marks** Syntax: range(start, stop, step) Parameter: › start: [optional] start value of the sequence › stop: next value after the end value of the sequence › step: [optional] integer value, denoting the difference between any two numbers in the sequence. Return: Returns a range type object. Eg. for i in range(0, 10, 2): print(i, end=",") Output : 0,2,4,6,8 ●	[4]	CO1	L2
(b)	Write a Python program to guess a number between 1 to 100. ● Correct looping structure **2 Marks**	[6]	CO1	L3

	• Correct Equations **2 Marks**
	• Correct Syntax **2 Marks**
	
	import random
	num = random.randint(1, 100)
	while True:
	  print('Guess a number between 1 and 100')
	  guess = input()
	  i = int(guess)
	  if i == num:
	  print('You won!!!')
	  break
	  elif i < num:
	  print('Try Higher Value')
	  elif i > num:
	  print('Try Lower Value')
	#Any recommendations for the game end
	print('if you guessed less than 6 times you WON the Game')			

4 (a)	Define the Scope of the variable. Differentiate local scope with global scope with example code snippets.
	• Definition/Description of the scope of a **variable [1 Marks]**
	• Differences with example code snippets **[4 Marks]**
	• A variable is only available from inside the region it is created. This is called **scope**. A variable created inside a function belongs to the *local scope* of that function, and can only be used inside that function.
	• while a variable that exists in the global scope is called a global variable.
	• A variable must be one or the other; it cannot be both local and global
	• There is only one global scope, and it is created when your program begins. When your program terminates, the global scope is destroyed, and all its variables are forgotten.
	
	Eg: def myfunc():
	  x = 300
	  print(x)
	  myfunc()
	• ***Local Variables Cannot Be Used in the Global Scope***
	• This code results in an error.
	def spam():
	  eggs = 31337
	spam()
	print(eggs)
	• ***Local Scopes Cannot Use Variables in Other Local Scopes***
	• ***Global Variables Can Be Read from a Local Scope-Example***
	def spam():
	  print(eggs)
	  eggs = 42
	spam()
	print(eggs)
		[5]	CO1	L2

	● It is acceptable to use the same variable name for a global variable and local variables in different scopes in Python
	`def spam():`
	`  ❶ eggs = 'spam local'`
	`  print(eggs)  # prints 'spam local'`
	`def bacon():`
	`  ❷ eggs = 'bacon local'`
	`  print(eggs)  # prints 'bacon local'`
	`  spam()`
	`  print(eggs)  # prints 'bacon local'`
	`❸ eggs = 'global'`
	`bacon()`
	`print(eggs)  # prints 'global'`
	Here the eggs defined with 'global' value is global variable.			

(b)	Differentiate the use of break and continue statement with example
	● Difference between the keywords(2 points)- **2 marks**
	● Example of the same – **3 Marks**
	The **break** keyword is used to break out a for loop, or a while loop, mostly when a condition is met.
	Eg: i = 1
	while i < 9:
	print(i)
	if i == 3:
	break
	i += 1
	Output : 1 2
	The **continue** keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration.
	Eg : i = 0
	while i < 6:
	i += 1
	if i == 3:
	continue
	print(i)
	Output : 1 2 4 5
		[5]	CO1	L2

5(a)	How to define and call functions in a python program? Illustrate with an example program ● Correct **logic [3 marks]** ● Correct **syntax [2 marks]** A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result. In Python a function is defined using the def keyword: def my_function(): print("Hello from a function") To call a function, use the function name followed by parenthesis: def my_function():	[5]	CO1	L2

	print("Hello from a function") my_function()			

(b)	Write a python to check whether the number inputted is in Fibonacci series or not. Hint: 0,1,1,2,3,5,8…. (Any number in the series is sum of the two previous numbers except first two).
	● Correct logic **[3 marks]**
	● Correct syntax**[2 marks]**
	n=int(input("Enter the number: "))
	c=0
	a=1
	b=1
	if n==0 or n==1:
	print("Yes")
	else:
	while c<n:
	c=a+b
	b=a
	a=c
	if c==n:
	print("Yes, It is a Fibonacci no")
	else:
	print("No, It is not a Fibonacci no ")
		[5]	CO1	L3

6 (a)	Define a function is_prime(n) to check n is prime or not. If 'n' is prime function should return True else it should return False.
	● Correct logic **[3 marks]**
	● Correct syntax**[2 marks]**
	. def is_prime(n):
	for i in range(2,n):
	if n%i == 0:
	return False
	return True
		[6]	CO1	L3

(b)	Explain about various logical operators with example.
	Boolean operators are called **logical operators.**
	Explanation with Truth Table [2 marks]
	Code snippets/Examples [2 marks]
	The three Boolean operators (and, or, and not) are used to compare Boolean values. Like comparison operators, they evaluate these expressions down to a Boolean value.
	Binary Boolean Operators:
	and operator
	>>> True and True
	True
	>>> True and False
	False
		[4]	CO1	L2

Table 2-2: The and Operator's Truth Table

Expression	Evaluates to...
True and True	True
True and False	False
False and True	False
False and False	False

Or operator

>>> False or True
True
>>> False or False
False

Table 2-3: The or Operator's Truth Table

Expression	Evaluates to...
True or True	True
True or False	True
False or True	True
False or False	False

The not Operator

>>> not True
False >>> not not not not True
True

Table 2-4: The not Operator's Truth Table

Expression	Evaluates to...
not True	False
not False	True

7 (a)	Define Exception. Explain with example how exceptions are handled in Python program.
	● Correct definition/**description [2 marks]**
	● Correct code and explanation **[3 marks]**
	Exceptions are raised when the program is syntactically correct, but the code resulted in an error. This error does not stop the execution of the program, however, it changes the normal flow of the program.
	try and **except** statements are used to catch and handle exceptions in Python.
	Statements that can raise exceptions are kept inside the try clause and the statements that handle the exception are written inside except clause.
	def AbyB(a , b):
	try:
	c = ((a+b) / (a-b))
	except ZeroDivisionError:
	print ("a/b result in 0")
	else:
	print (c)
		[5]	CO1	L2

(b)	Explain the following methods with example code snippets a)remove() b) append() c) insert()	[5]	CO1	L2

	At least 3 functions if they explain example give 3 marks (a) The remove() method is passed the value to be removed from the list it is called on. E.g.: spam = ['cat', 'bat', 'rat', 'elephant'] spam.remove('bat') spam o/p: ['cat', 'rat', 'elephant'] (b)To add new values to a list, use the append()methods. E.g. spam = ['cat', 'dog', 'bat'] spam.append('moose') spam o/p:['cat', 'dog', 'bat', 'moose'] (c) The insert() method can insert a value at any index in the list. The first argument to insert() is the index for the new value, and the second argument is the new value to be inserted. E.g. spam = ['cat', 'dog', 'bat'] spam.insert(1, 'chicken') spam o/p: ['cat', 'chicken', 'dog', 'bat']			

(Chief Course Instructor)