Roll No. | | | | |

Internal Assessment Test 1 – Nov. 2023

Sub:	Introduction to Python Programming--**Solutions and Scheme**					Sub Code:	BPLCK105B	Branch:	Chemistry Cycle
Date:	03-11-2023	Duration:	90 min's	Max Marks:	50	Sem / Sec:	I / Chemistry Cycle		OBE

	Answer any FIVE FULL QUESTIONS	MARKS	CO	RBT
1 (a)	Explain the usage of following functions with example code i) input() ii) print() iii) range() **(1+1) =2M x 3=6 Marks** ● Python **input()** function is used to take user input. By default, it returns the user input in form of a string. Eg : color = input("What color is rose?: ") print(color) ● The **print()** function prints the specified message to the screen, or other standard output device. The message can be a string, or any other object, the object will be converted into a string before written to the screen ● Eg : print("Hello", "how are you?") Hello how are you? ● The **range()** function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number. Eg: x = range(3, 5, 1) for n in x: print(n) 3 4	[6]	CO1	L2
(b)	Predict the output 1. print(2**3 + (5 + 6)**(1 + 1))**(1Mark)** 129 **2.** print(type(5 / 2)) **(0.5 Each)** print(type(5 // 2)) **float and int (2.5 and 2)** What will be the data type for var **(2 marks)** 3. var = 10 print(type(var)) var ="Hello" print(type(var)) **int and str**	[4]	CO1	L3
2 (a)	Which of the following is valid and invalid variable. Justify your answer**.(Each correct answer 0.5)** a) _a=1 b) __a=1 c) __str__=1 d) 1_string=2 e) stud id=5 f) foo Valid variable: a, b, c, f	[3]	CO1	L3

	Invalid variables:d,e You can name a variable anything as long as it obeys the following three rules: It can be only one word. It can use only letters, numbers, and the underscore (_) character. It can't begin with a number.			

(b)	Write a python program using ladder if statement to display University grade based on marks as following:
	
	>=90 A+
	>=80 and <90. A
	>=70 and <80. B
	>=60 and <70. C
	>=50 and <60. D
	>=40 and <50. E
	<40. F
	
	● Correct logic **[3 marks]**
	● Correct syntax**[4 marks]**
	
	print("Enter Marks Obtained by student: ")
	marks = int(input())
	if marks>=90 :
	print("Your Grade is A+")
	elif marks>=80 and marks<90:
	print("Your Grade is A")
	elif marks>=70 and marks<80:
	print("Your Grade is B")
	elif marks>=60 and marks<70:
	print("Your Grade is C")
	elif marks>=50 and marks<60:
	print("Your Grade is D")
	elif marks>=40 and marks<50:
	print("Your Grade is E")
	elif marks<40:
	print("Your Grade is F")
	else:
	print("Invalid Input!")
		[7]	CO1	L3

3 (a)	Explain string concatenation and replication with example. ● String concatenate explanation with example (3 marks) ● Replication explanation with example (3 marks) **Concatenate Strings in Python** String Concatenation is the technique of **combining two strings**. String Concatenation can be done using the '+' Operator This operator can be used to add multiple strings together. However, the arguments must be a string. var1 = "Hello " var2 = "Geek" var3 = var1 + var2 print(var3) Here, The + Operator combines the string that is stored in the **var1** and **var2** and stores in another variable **var3**. **String replication**	[4]	CO1	L2

	The * operator is used for multiplication when it operates on two integer or floating-point values. But when the * operator is used on one string value and one integer value; it becomes the string replication operator. 3*3 >>> 'Alice' * 5 'AliceAliceAliceAliceAlice' The expression evaluates down to a single string value that repeats the original a number of times equal to the integer value.			

(b)	What is a Keyword argument? Explain the use of 'sep' and 'end' argument in print() function with an example?
	● Explanation (3 Marks)
	● Example(each 1 mark*3)
	The separator between the arguments to print() function in Python is space by default , which can be modified and can be made to any character, integer or string as per our choice
	However, rather than through their position, keyword arguments are identified by the keyword put before them in the function call. Keyword arguments are often used for optional parameters.
	The print() function has the optional parameters end and sep to specify what should be printed at the end of its arguments and between its arguments (separating them), respectively.
	Example:
	print('Hello')
	print('World')
	O/p :
	Hello
	World
	
	print('Hello', end=' ')
	print('World')
	Hello World
	
	>>> print('cats', 'dogs', 'mice')
	cats dogs mice
	>>> print('cats', 'dogs', 'mice', sep=',')
	cats,dogs,mice
		[6]	CO1	L3

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. Eg: def myfunc(): x = 300 print(x) myfunc() ● ***Local Variables Cannot Be Used in the Global Scope*** ● This code results error. def spam(): eggs = 31337 spam()	[5]	CO1	L2

	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) ● It is acceptable to use the same variable name for a global variable and local variables in different scopes in Python			

(b)	Differentiate the use of break and continue statements 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 declare and call functions in a python program? Illustrate with an example script. ● 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(): print("Hello from a function") my_function()	[5]	CO1	L2
(b)	Write a python function to check whether three given numbers can form the sides of a triangle. Hint: Three numbers can be the sides of a triangle if none of the numbers are greater than or equal to the sum of the other two numbers.	[5]	CO1	L3

	• Correct logic **[3 marks]**
	• Correct syntax**[2 marks]**
	def triangle(a,b,c):
	if a<b+c and b<a+c and c<a+b:
	return True
	else:
	return False			

6 (a)	Develop a program to generate Fibonacci sequence of length (N). Read N from the console. • Correct **logic [3 marks]** • Correct **syntax [3 marks]** Input: Enter number Output: Fibonacci series n=int(input("enter the number")) a=0 b=1 sum=0 i=0 print("fibonacci series") while(i<=n): print(sum) a=b b=sum sum=a+b i=i+1 N=5 Fibonacci sequence =0 1 1 2 3	[6]	CO1	L3
(b)	Explain Boolean type of python with examples of relational operators. • Explanation(2 Marks) • Example (2 Marks) The three Boolean operators (and, or, and not) are used to compare Boolean values. Comparison operator's also known as relational operator. Compare two values and evaluate down to a single Boolean value. These operators evaluate to true or false depending upon the values given to them. • >>> 2 + 2 == 4 and not 2 + 2 == 5 and 2 * 2 == 2 + 2 4==4and not 4==5 and 4==4 • True and not(False)and(True) • True and True and True • True and True • True	[4]	CO1	L2

7 (a)	Explain Exception Handling in python with an example.
	
	● 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)	Write a python program to check whether a year is a leap year or not.
	
	● Correct **logic [3 marks]**
	● Correct **syntax [2 marks]**
	
	Input: Enter Year
	Output: Leap year or Not
	
	Program:
	year =int(input("enter year"))
	if (year % 400 == 0) and (year % 100 == 0):
	print("{0} is a leap year".format(year))
	elif (year % 4 ==0) and (year % 100 != 0):
	print("{0} is a leap year".format(year))
	else:
	print("{0} is not a leap year".format(year))
		[5]	CO1	L3

CI **CCI** **HOD**