
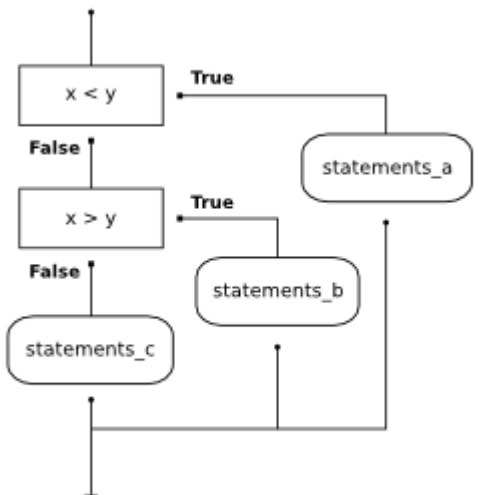


CMR INSTITUTE OF TECHNOLOGY		USN <input type="text"/>						 <small>CELEBRATING 30 YEARS</small> CMRIT <small>CMR INSTITUTE OF TECHNOLOGY, BANGALURU.</small> <small>ACCREDITED WITH A++ GRADE BY NAAC</small>	
Internal Assessment Test - II									
Sub:	Introduction to Python, Data and Control Systems						Code:	22MBABA303	
Date:	06-03-2024	Duration:	90 mins	Max Marks:	50	Sem:	III	Branch:	MBA
SET- III – Answer Key									
							Marks	OBE	
								CO	RBT
Part A - Answer Any Two Full Questions (2* 20 = 40 marks)									
1 (a)	Define the term ‘boolean values’ and write example. In programming you often need to know if an expression is True or False . You can evaluate any expression in Python, and get one of two answers, True or False . When you compare two values, the expression is evaluated and Python returns the Boolean answer: Example: <pre>print(10 > 9) print(10 == 9) print(10 < 9)</pre> Output: <pre>True False False</pre>						[03]	CO3	L1
(b)	How can you use the "while" loop to create an infinite loop in Python? Python While Loop is used to execute a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the line immediately after the loop in the program is executed. Infinite while Loop in Python Here, the value of the condition is always True. Therefore, the body of the loop is run infinite times until the memory is full. Example: <pre>age = 28 # the test condition is always True while age > 19: print('Infinite Loop')</pre> An infinite loop does not have an explicit end (no loop termination condition is specified by the programmer). Example: <pre>while True: print("Scaler Topics")</pre> Output: <pre>Scaler Topics Scaler Topics Scaler Topics . . .</pre>						[07]	CO3	L2

<p>(c)</p>	<p>Analyze the concept of a chained conditional in Python with a practical example.</p> <p>Python provides an alternative way to write nested selection such as the one shown in the previous section. This is sometimes referred to as a chained conditional.</p> <pre> if x < y: print("x is less than y") elif x > y: print("x is greater than y") else: print("x and y must be equal") </pre> <p>The flow of control can be drawn in a different orientation but the resulting pattern is identical to the one shown above.</p>  <pre> graph TD Start(()) --> Cond1{x < y} Cond1 -- True --> StatementsA([statements_a]) Cond1 -- False --> Cond2{x > y} Cond2 -- True --> StatementsB([statements_b]) Cond2 -- False --> StatementsC([statements_c]) StatementsA --> Exit(()) StatementsB --> Exit StatementsC --> Exit </pre> <p>elif is an abbreviation of else if. Again, exactly one branch will be executed. There is no limit of the number of elif statements but only a single (and optional) final else statement is allowed and it must be the last branch in the statement.</p>	<p>[10]</p>	<p>CO3</p>	<p>L4</p>
<p>2 (a)</p>	<p>Define the term tuple and write example.</p> <p>Tuples are used to store multiple items in a single variable.</p> <p>Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are <u>List</u>, <u>Set</u>, and <u>Dictionary</u>, all with different qualities and usage.</p> <p>A tuple is a collection which is ordered and unchangeable.</p> <p>Tuples are written with round brackets.</p> <p>Create a Tuple:</p> <pre> thistuple = ("apple", "banana", "cherry") print(thistuple) </pre> <p>Output:</p> <pre> ('apple', 'banana', 'cherry') </pre>	<p>[03]</p>	<p>CO4</p>	<p>L1</p>
<p>(b)</p>	<p>How can you access elements of an array in Python? Write example.</p> <p>Arrays:</p> <p>Arrays are used to store multiple values in one single variable:</p> <p>Create an array containing car names:</p> <pre> cars = ["Ford", "Volvo", "BMW"] </pre> <p>Access the Elements of an Array</p> <p>You refer to an array element by referring to the <i>index number</i>.</p> <p>Example</p> <p>Get the value of the first array item:</p> <pre> cars = ["Ford", "Volvo", "BMW"] x = cars[0] print(x) </pre>	<p>[07]</p>	<p>CO4</p>	<p>L3</p>

	<p>Output: Ford</p> <p>Example Modify the value of the first array item: cars = ["Ford", "Volvo", "BMW"] cars[0] = "Toyota" print(cars)</p> <p>Output: ['Toyota', 'Volvo', 'BMW']</p>			
<p>(c)</p>	<p>Analyze the role of fruitful functions in writing modular and reusable code. Fruitful Python functions are those that yield a result after computation. Unlike void functions, which perform a job without returning a result, productive functions add to a program's overall logic by generating an output. These routines include reusable pieces of code, which improves code modularity and maintainability. In Python, a fruitful function is like a powerhouse that not only performs a task but also hands back the required piece of information when it's done – like a loop that keeps on giving. Picture it as a function that not only does its job but also brings back a result that you can use elsewhere in your code. A function that returns a value is called fruitful function. Example: Root=sqrt(25) Example: def add(): a=10 b=20 c=a+b return c c=add() print(c)</p> <p>Void Function A function that perform action but don't return any value. Example: print("Hello") Example: def add(): a=10 b=20 c=a+b print(c) add()</p> <p>Return values: return keywords are used to return the values from the function. example: return a – return 1 variable return a,b– return 2 variables return a,b,c– return 3 variables return a+b– return expression return 8– return value</p>	<p>[10]</p>	<p>CO3</p>	<p>L4</p>
<p>3 (a)</p>	<p>Define dictionaries and give example. Dictionary Dictionaries are used to store data values in key: value pairs. A dictionary is a collection which is ordered*, changeable and do not allow duplicates. Dictionaries are written with curly brackets, and have keys and values: Example: thisdict = { "brand": "Ford",</p>	<p>[03]</p>	<p>CO4</p>	<p>L1</p>

	<pre>"model": "Mustang", "year": 1964 } print(thisdict) Output: {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}</pre>			
(b)	<p>How can you use list comprehension in Python to create a new list?</p> <p>List Comprehension List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.</p> <p>Example: Based on a list of fruits, you want a new list, containing only the fruits with the letter "a" in the name. Without list comprehension you will have to write a for statement with a conditional test inside:</p> <p>Example: fruits = ["apple", "banana", "cherry", "kiwi", "mango"] newlist = []</p> <pre>for x in fruits: if "a" in x: newlist.append(x) print(newlist) Output: ['apple', 'banana', 'mango']</pre>	[07]	CO4	L2
(c)	<p>Analyze the role of various built-in modules like datetime, time, and math in Python.</p> <p>Math Module in Python The math module offers mathematical functions used for advanced arithmetic operations. This includes trigonometric functions, logarithmic functions, and constants like pi and e. This module is used to perform complex calculations using Python program.</p> <p>Example In the below example, we have used math module the find the square root of a number using math.sqrt() method and the value of PI using math.pi method and then print the result using print() function of Python.</p> <pre>import math sqrt_val = math.sqrt(64) pi_const = math.pi print(sqrt_val) print(pi_const)</pre> <p>Output</p> <pre>8.0 3.141592653589793</pre> <p>datetime module in Python The “datetime” module allows for manipulation and reading of date and time values. Some of the basic method of “datetime” module are “datetime.date”, “datetime.time”, “datetime.datetime”, and “datetime.timedelta”.</p> <p>Example In the below example, we have print the today’ date and current time by using datetime.date.today() method and datetime.datetime.now().time() method of “datetime” module in Python.</p>	[10]	CO5	L4

	<pre>import datetime date_today = datetime.date.today() time_now = datetime.datetime.now().time() print(date_today) print(time_now)</pre> <p>Output</p> <pre>2023-10-19 07:28:16.279090</pre> <p>Importing time module The time module comes with Python's standard utility module, so there is no need to install it externally. We can simply import it using the import statement.</p> <pre>import time</pre>			
Part B - Compulsory (01*10=10 marks) – CASE STUDY				
4	<p>Develop a Python program that extensively uses lists, tuples, and dictionaries.</p> <p>Python List Python Lists are just like <u>dynamic-sized arrays</u>, declared in other languages (vector in C++ and ArrayList in Java). Lists need not be homogeneous always which makes it the most powerful tool in Python.</p> <p>Applications of Python List</p> <ul style="list-style-type: none"> • Used in JSON format • Useful for Array operations • Used in Databases <p>Python Tuple A <u>Tuple</u> is a collection of Python objects separated by commas. In some ways, a tuple is similar to a list in terms of indexing, nested objects, and repetition but a tuple is immutable, unlike lists that are mutable.</p> <p>Applications of Python Tuple</p> <ul style="list-style-type: none"> • Used to insert records in the database through SQL query at a time.Ex: (1.'sravan', 34).(2.'geek', 35) • Used in parentheses checker <p>Python Dictionary <u>Dictionary</u> in Python is an ordered (since Py 3.7) [unordered (Py 3.6 & prior)] collection of data values, used to store data values like a map, which, unlike other Data Types that hold only a single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized.</p> <p>Applications of Python Dictionary</p> <ul style="list-style-type: none"> • Used to create a data frame with lists • Used in JSON 	[10]	CO4	L3

Course Outcomes (COs)		PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4
CO1:	Understand the concepts of python programming.									
CO2:	Structure a simple python programs for solving problems.									
CO3:	Apply the knowledge of decompose a python program into functions.				1a,1b,1c		2c			
CO4:	Analyse and represent compound data using python lists, tuples, and dictionaries.		2a, 2b		3a,4					3b
CO5:	Read and write data form/to files in python program.		3c							
