B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

May 2023 Semester End Main Examinations

Programme: B.E.   **Semester: I**
Branch: Common to all Branches   **Duration: 3 hrs.**
Course Code: 22CS1ESPYP   **Max Marks: 100**
Course: Introduction to Python Programming   **Date: 08.05.2023**

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

UNIT - I

1   a)   Justify whether Python is procedural or object-oriented programming language.   **6**

b)   Consider the code given below. Analyze, identify the problem & correct the code.   **6**

```
1. print("Python is a scripting language")
2. print("Python can be used as a procedural or object-oriented
programming language")
3. print("PYTHON is interpreted language")
4. Errors = 10
5. Sum= 0
6. Logic = 3
7. while syntax!=0:
8.     Sum+= 1
9.     Syntax -= logic
10. print("Python is a dynamic typed language,)
11. print("Python is the most preferred language for gaming
applications")
```

c)   A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: "The quick brown fox jumps over the lazy dog". Write a Python program to check a sentence to see if it is a pangram or not.   **8**

UNIT – II

2   a)   Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. The numbers obtained should be printed in a comma-separated sequence on a single line.   **6**

b)   Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers.
Suppose the following input is supplied to the program:
1,2,3,4,5,6,7,8,9
Then, the output should be:
1,3,5,7,9   **8**

	c)	Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples. Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)] Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]	**6**

UNIT - III

3	a)	Illustrate with an example function call and passing arguments to a function in Python.	**6**
	b)	Demonstrate with an example the advantages of sequence packing and unpacking of tuples.	**6**
	c)	Write a Python program for the following scenario: Given an array of names of candidates in an election. A candidate name in the array represents a vote cast to the candidate. Print the name of candidates received Max vote. If there is tie, print a lexicographically smaller name.	**8**

OR

4	a)	Given a list of elements, perform grouping of similar elements, as different key-value list in dictionary. Sample Input : test_list = [4, 6, 6, 4, 2, 2, 4, 8, 5, 8] Output : {4: [4, 4, 4], 6: [6, 6], 2: [2, 2], 8: [8, 8], 5: [5]}	**6**
	b)	Analyse the following code and write the output with suitable justification: def Merge(dict1, dict2): return(dict2.update(dict1)) dict1 = {'a': 10, 'b': 8} dict2 = {'d': 6, 'c': 4} print(Merge(dict1, dict2)) print(dict2)	**6**
	c)	Write a Python program to find the maximum and minimum value in a tuple	**8**

UNIT - IV

5	a)	Demonstrate how inheritance supports reuse with a suitable example.	**6**
	b)	Demonstrate how exceptions are handled in Python with a suitable example.	**8**
	c)	Write a Python class to get all possible unique subsets from a set of distinct integers.	**6**

OR

6	a)	Demonstrate with an example how multiple except blocks can be written for a single try block to handle multiple exceptions.	**8**
	b)	Create a Vampire class that has one class attribute and two attributes. The class attribute is a tuple holding the following hunger levels: "stuffed","full","peckish","hungry", and "starving". The two attributes need to hold the name and the hunger. The hunger is an integer that will be used as an index into the hunger levels tuple. i. Define a constructor (__init__) that has one parameter for the name. ii. Create a method (__str__) to be able to print the object. Use the hunger	**12**

levels and hunger attributes.

iii. Create a suckBlood method that will decrease the hunger by 1.

iv. Create a main function. In this function, create a list of human objects. You can hard code the data. Let the user enter his/her name and create a Vampire Object.

v. Create a menu function to allow the user to print the humans and suck blood.

vi. Create a function to print the humans. To do this, use the enumerate function to print an index number or each human and then his/her info.

vii. Create a function to suck blood. This will ask the user to enter an index number of a human. It will then call the suckBlood method on the human and the vampire   (provided the human had some blood left).

viii. Please add error checking(exceptional handling) and comments

UNIT - V

7   a)   Suppose we have created a file with 500 lines of data and the the file object   **5**
reference is "f".
Illustrate what each of these following operations does:
   (i)   F.seek(0)
   (ii)   F.seek(100,1)
   (iii)   F.seek(-10,2)
   (iv)   F.seek(0,2)
   (v)   F.tell()

   b)   Demonstrate serialisation operation of files with a suitable example.   **5**

   c)   Analyze the program, find the error and write the correct program to get the   **5**
output given below:
#PROGRAM
text_file = open("c.txt", "a+")
text_file.write("Line 1\n")
text_file.write("This is line 2\n")
text_file.write("That makes this line 3\n')
lines = ["L1\n","L2";"L3\n"]
text_file.write(lines)
a=text_file.read()
print(a)
text_file.close()
OUTPUT:
Line 1
This is line 2
That makes this line 3
L1
L2
L3

   d)   Write a program that will calculate the average word length of a text stored in   **5**
a file (i.e the sum of all the lengths of the word tokens in the text, divided by
the number of word tokens).

******