Unit 3 Question Bank

1.What will be the output of the following programs?

```
def fun(x, y) :     if x == 0 :         return y     else :
      return fun(x - 1, x * y) print(fun(4,
2))
```

2.Write a recursive function to obtain average of all numbers present in a given list.

3. A list contains some negative and some positive numbers. Write a recursive function that sanitizes the list by replacing all negative numbers with 0.

4.What will be the output of the following program

```
def fun(num) :
        if num > 100 :
                return num - 10
        return fun(fun(num + 11))
print(fun(75))
```

5. Write a recursive program to check whether the given no is a power of two 6.

6) Write a recursive Python function to calculate the factorial of a number.

7. Write a python program to implement a recursive function to calculate the nth Fibonacci number.
8. Write a recursive Python function to find the greatest common divisor (GCD) of two numbers.
9. Write a python program to implement a recursive function to reverse a string in Python.
10. Write a recursive Python function to calculate the sum of digits in a given number.
11. Write a python program to implement a recursive function to find the power of a number (x^n) in Python.

12.State whether the following statements are True or False:

a. If a recursive function uses three variables **a**, **b** and **c**, then the same set of variables are used during each recursive call.
b. Multiple copies of the recursive function are created in memory.
c. A recursive function must contain at least 1 **return** statement.
d. Every iteration done using a **while** or **for** loop can be replaced with recursion.
e. Infinite recursion can occur if the base case is not properly defined.
f. A recursive function is easy to write, understand and maintain as compared to a one that uses a loop.

13.Write a program that uses a generator to create a set of unique words from a line input through the keyboard.

14.Write a program that uses a generator to find out maximum marks obtained by a student and his name from tuples of multiple students.

15. Implement a recursive Python function to check if a given word is a palindrome.

16. Write a program that uses a generator that generates characters from a string in reverse order.
17. What is the difference between the following statements?

sum([x**2 for x in range(20)]) sum(x**2
for x in range(20))

18. Suppose there are two lists, each holding 5 strings. Write a closure program to generate a list that consists of strings that are concatenated by picking corresponding elements from the two lists.
19. Explain what a closure is in Python and provide an example of a closure.
20. How does a closure differ from a regular function in Python?
21. Explain the concept of a free variable in the context of closures.
22. How can closures be used to implement data hiding in Python? Provide an example.
23. Discuss a scenario where using closures in Python can be beneficial.
24. Explain the term "lexical scoping" in the context of closures.
25. Give an example of a closure that takes multiple arguments.
26. How does the lifespan of a variable in a closure compare to the lifespan of a variable in a regular function?

27. State whether the following statements are True or False:

a. A decorator adds some features to an existing function.

b. Once a decorator has been created, it can be applied to only one function within the program.

c. It is mandatory that the function being decorated should not receive any arguments.

d. It is mandatory that the function being decorated should not return any value.

28. Explain what a callback function is in Python and provide an example.
29. Illustrate an example of callback in the context of handling user input in a graphical user interface (GUI) application.
30. Explain the concept of a decorator in Python and give an example of a simple decorator function.
31. Discuss the concept of nested decorators in Python and provide an example.
32. Explain the concept of a generator in Python and provide an example of a simple generator function.
33. Discuss the advantages of using generators over lists in terms of memory efficiency. 34. Give an example of using a generator expression to generate a sequence of square numbers.
35. Question: Explain the difference between a generator function and a regular function in
36. Discuss the role of the next function in iterating over generator objects in Python.
37. Question: Give an example of using a generator to implement the Fibonacci sequence in Python.

38. Write a program to add and subtract two numbers using wxPython GUI framework.
39. What is closure in python? List any 3 criteria for closure.
40. Explain different ways to import modules in python.
41. Write a generator program which can iterate over the numbers, which are divisible by 5, between a given rage 0 and n.
42. Create a module called Util.py. Add functions for the following into this file.
    a) convert temperature in Centigrade to Fahrenheit
    b) convert temperature in Fahrenheit to Centigrade
43. Explain 4 ways of importing a file.
44. Find the output when a.py is executed as python a.py.

```
File:abc.py pint("this is
with in abc.py")
print('abc',__name__)

File:a.py
print("this is with in a.py")
a=10 def f1():    print("this
is function f1")
import abc print
('a',__name__)
```

45. State True/False

1 True or False? Modular design allows programs to be broken down into manageable size parts, in which each module (part) is a function with clearly specified functionality.

2 True or False? A module may be just the design of specific functionality, without any implementation.

3 True or False? A module, in terms of program design, may consist of a single function.

4 True of False? Modules are useful for the design, development, testing and maintenance of computer programs.

5 True or False? The specification of a module is referred to as the module's interface.

6 True or False? A client should always be knowledgeable of the implementation of a module using, and not rely soley on the module's specification (interface). 7 True or False? A docstring is a specific feature of Python for providing the specification of program elements (e.g., functions and modules).

8 True or False? The __doc__ extension in Python is used in the creation of docstrings.

9 True or False? A module in Python is a file containing definitions and statements, which must contain a module declaration statement as the first line.

10 True or False? Modules in Python have their own namespace.

11 True or False? A name clash results when the same identifier is used in more than one function of a given program.

12 True or False? A fully qualified identifier is an identifier with the name of the module that it belongs to prepended to it.

13 True or False? The main module of a Python program is the one that is directly executed when a program is run.

46.For the following module, #
module driving_conversions
def milesPerHr(km_speed):
"'Returns miles per hour for kilometers per hour given in speed. '" def milesPerGal(KilometersPerLiter):
"'Returns miles per gallon for provided kilometers per liter. '"
(a) Provide a main module that makes use of this module to prompt the user for the conversion desired, and displays the converted result, using the import modulename form of import.
(b) Modify (a) above using the from modulename import identifier, identifier form of import.
(c) Modify (a) above using the from modulename import * form of import. (d) Modify (a) above using the from modulename import identifier as new_identifier form of import. (e) Describe the namespaces for the main modules of (a), (b), (c) and (d) above.

47. Fill in the Blanks
1._____ extension in Python is used in the creation of docstrings.
2._____ is used to create special directories (folders) for Python modules to be placed.
3._____of a given module cannot be accessed by any importing module.
48.How does a module source code file become a module object?
49. How is changing sys.path different from setting PYTHONPATH to modify the module search path.
50. What does it mean when a module's __name__ variable is the string "__main__"?
51. What is significant about variables at the top level of a module whose names begin with a single underscore?
52. What is the purpose of an __init__.py file in a module package directory?
53. How can you avoid repeating the full package path every time you reference a package's content?
54. Which directories require __init__.py files?
55. When must you use import instead of from with packages?
56. How do you debug a program in Python? Is it possible to step through the Python code?
57. List down some of the PDB commands for debugging Python programs?What is the command to debug a Python program?
58. What is a package?
59.Write a Python script to display the current date and time.
60. What is the special file that each package in Python must contain?

61. Write a Python GUI program using wxPython to create three buttons. Change the background colour of the panel when different buttons are clicked.
62. Explain the steps to create widgets in wxPython. Write a program to display a StaticText label when a button is clicked.
63. Write a Python GUI program to implement a simple calculator using wxPython.
64. Explain in detail about sizer layouts in wxPython (BoxSizer, GridSizer).
65. Write a Python wxPython program with window size 200×300 that displays a button at a fixed position. On clicking the button, display a message dialog saying “Button clicked”.
66. Explain wxPython widgets: MenuBar and Menu.
67. Explain wxPython widgets: Button and RadioButton.
68. Explain layout management in wxPython using sizers and absolute positioning.
69. Write a wxPython GUI program to create two buttons where clicking one hides the other and clicking again shows it.
70. Write a wxPython GUI program to handle button click events.