MMM001 – Programming in Python

Questions:

1. What is Python? List some of the language properties and what makes it widely adopted.
2. Regarding the code structure:
    a. How are code blocks defined?
    b. What is PEP8?
3. What is the difference between Lists and Tuples?
4. Which data types are (built-in) supported in Python?
5. What is the result of:
    a. print([(c,n) for c in 'abc' for n in [1,2,3]])
    b. print((c,n) for c in 'abc' for n in [1,2,3])
    - What is a *generator*? How do we print the content of a generator?
6. *Slice* (in Python!) is the action of selecting one or more items from a sequence (list, tuple, string, etc.) using the according indexes. Consider a size-N list of random numbers. Using slicing notation, what are the statements for:
    a. Print the first 5 elements of the list
    b. Print the last 5 elements of the list
    c. Print every other element of the list, starting from the items at the middle
    - How do I copy a list?
7. The Python interpreter provides a **help** function to access the internal/online documentation of objects and functions, modules, etc.
    a. What is the name of this function?
    b. How do we use it?
    c. Give an example of its use to access dictionaries documentation.
    d. From where such function takes the content (i.e., the documentation) from?
8. Consider the code below. Write the values of A1, A2, …, An.

```
A0 = dict(zip((5,4,3,2,1) ,('a','b','c','d','e')))
A1 = list(range(10))
A2 = sorted([i for i in A1 if i in A0])
A3 = sorted([A0[s] for s in A0])
A4 = [i for i in A1 if i in A3]
A5 = {i:i*i for i in A1}
A6 = [[i,i*i] for i in A1]
```

9. What is the output of the following code:

```
def f(x,l=[]):
   for i in range(x):
         l.append(i*i)
   print(l)
f(2)
f(3,[3,2,1])
f(3)
```

10. Explain mutable and immutable objects.
11. What is '`lambda`' used for in Python? Give an example of its use.
12. What are *docstrings*? Where are they used, and how can we access them from the interpreter?
13. What are **`@classmethod`**, **`@staticmethod`**, **`@property`**? What are they used for?
14. How can you convert:
    a. A number to a string
    b. A string to an integer
    c. A string to a float
    d. A tuple to a list
    e. A list to a set
15. What are *local* and *global* variables?
    a. Can we use (read) a global variable inside a function?
    b. Can we modify (write) the value of a global from inside a function?
    c. Can we read a *local* variable, defined in a function, from an outer scope?
    d. Can we read a *local* variable, defined in an outer scope, from inside a inner function?
16. How can we generate *random* numbers?
    a. Write the code for generating 10 random numbers between [0,1]
    b. Write the code for generating a list of 10 sorted random numbers
17. Consider a list of strings; Write the code to:
    a. Count the *total* number of capital letters in the list
    b. Print any (string) element of the list that is a number
    c. Print "Empty string" and the *position* in the list for every *empty* string
18. Regarding lists, how to:
    a. Sort a list in ascending order
    b. Sort a list in descending order
    c. Reverse a list
    d. Create another list with only the items from *even* index/positions (0,2,4,…)
    e. Create another list with only the *even* (valued) items
19. How do "try-except" statements work? Why are they used for?
20. In lists, what is the difference between '`append`' and '`extend`' methods?

21. In loops, what are '`break`' and '`continue`' used for? Give an example (and explain it).
22. Write a Python program to check whether a given string is a palindrome or not, without using an iterative method. Note: A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., *madam* or *nurses run*.
23. Write a Python program to calculate the sum of a list of numbers.
24. How do you read a *random line* from a file?
25. Write a Python program to count the total number of lines in a text file.
26. How do you check the *type* of an object?
27. What are *membership operators* in Python? Give an example.
28. What are *identity operators* in Python? Give an example.
29. Explain *binary operators* in Python? What are they used for?
30. Implement (i.e., write) the Birthday Paradox:

    The Birthday Paradox (or Problem) concerns the probability
    of two randomly selected people having anniversary at the
    same date. We want to simulate different sizes of groups that by simulating birthdays and reading
    out those probabilities