Python for Data Science

Assignment – 1

Name : Priyanshu Deepak Bari

SRN : PES2UG22EC102

Section : 7-B

MCQs :

1. What is the correct order of precedence for the following operators from highest to lowest?

   A) Parentheses, Exponent, Multiplication, Division, Addition, Subtraction
   B) Exponent, Parentheses, Multiplication, Division, Addition, Subtraction
   C) Parentheses, Exponent, Addition, Subtraction, Multiplication, Division
   D) Exponent, Parentheses, Addition, Subtraction, Multiplication, Division

   Ans. Option A

2. Which of the following data types **does NOT** support the clear() method to remove all items from the object?

   A) List
   B) Dictionary
   C) Set
   D) Tuple

   Ans. Option D

Descriptive Questions :

1. State the 5 basic data types in Python. Provide a brief description and an example for each. 5 marks

   Ans.

Basic data types	Description	Values
Boolean	represents two values of logic and associated with conditional statements	True and False
Integer	positive and negative whole numbers	set of all integers, Z
Complex	contains real and imaginary part (a+ib)	set of complex numbers
Float	real numbers	floating point numbers
String	all strings or characters enclosed between single or double quotes	sequence of characters

   Examples :

   Boolean : True, False

   Integer : 5

   Complex : 4+9j

   Float : 5.6

   String : “Hello”

2. i. Explain the key characteristics of NumPy arrays, differentiating them from standard Python lists. 4 marks
   ii. Create a 2-D NumPy array from nested Python lists, then change the shape of this array using the appropriate NumPy attribute. 2 marks
   iii. Explain .itemsize attribute from NumPy. 1 mark

Ans. NumPy arrays store only one data type (homogeneous), lists can hold mixed data types (heterogeneous).

Once created, a NumPy array's size cannot change, lists can grow/shrink dynamically.

NumPy arrays are much faster because they use contiguous memory blocks and are implemented in C, Lists are slower.

Creating a 2-D NumPy array from a nested Python list and then reshaping:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
arr.reshape(3,2)
```

.itemsize attribute : returns the length of each element of the array in bytes.