**WORKSHEET 2**

**PYTHON**

# Q1 to Q8 have only one correct answer. Choose the correct option to answer your question.

1. Which of the following is not a core datatype in python?
   1. list B) struct

C) tuple C) set

Answer: (B)

1. Which of the following is an invalid variable name in python?
   1. \_init\_ B) no\_1

C) 1\_no D) \_1

Answer: (D)

1. Which one of the following is a keyword in python?
   1. in B) \_init\_

C) on D) foo

Answer: (A)

1. In which of the following manner are the operators of the same precedence executed in python?
   1. Left to Right B) BODMAS

C) Right to Left D) None of these

Answer: (B)

1. Arrange the following in decreasing order of the precedence when they appear in an expression in python?

i) Multiplication ii) Division iii) Exponential iv) Parentheses

A) iii – iv – ii – i B) iii – iv – i – ii

C) iv – iii – ii – i D) iii – ii – i – iv

Answer: (B)

6. (28//6)\*\*3/3%3 = ?

A) 7.1111… B) 0

C) 0.3333… D) 1

Answer: (C)

1. Which of the following is not equal to x16 ?

A) x\*\*4\*\*4 B) x\*\*16

C) x^16 D) (x\*\*4)\*\*4

Answer: (D)

1. a = input(“Enter an integer”). What will be the data type of a?
   1. int B) str

C) float D) double

Answer: (A)

# Q9 and Q10 have multiple correct answers. Choose all the correct options to answer your question.

1. Which of the following statements are correct?
   1. Division and multiplication have same precedence in python
   2. Python’s operators’ precedence is based on PEDMAS
   3. Python’s operators’ precedence is based on VBODMAS
   4. In case of operators’ having same precedence, the one on the left side is executed first.

Answer: (A),(B),(D)

1. Which of the following is(are) valid statement(s) in python?

A) abc = 1,000,000 B) a b c = 1000 2000 3000

C) a,b,c = 1000, 2000, 3000 D) a\_b\_c = 1,000,000

Answer: (A),(B)

# Q11 to Q13 are subjective questions, answer them briefly

1. Differentiate between a list, tuple, set and dictionary.

Answer: **List** and **tuple** is an ordered collection of items. **Dictionary** is unordered collection. **List** and **dictionary** objects are mutable i.e. it is possible to add new item or delete and item from it. **Tuple** is an immutable object.

## Lists

Lists is one of the most versatile collection object types available in Python. (dictionaries and tuples being the other two, which in a way, more like variations of lists).

* A list is a mutable, ordered sequence of items. As such, it can be indexed, sliced, and changed. Each element can be accessed using its position in the list.Python lists do the work of most of the collection data structures found in other languages and since they are built-in, you don’t have to worry about manually creating them.
* Lists can be used for any type of object, from numbers and strings to more lists.
* They are accessed just like strings (e.g. slicing and concatenation) so they are simple to use and they’re variable length, i.e. they grow and shrink automatically as they’re used. List variables are declared by using brackets [ ] following the variable name.

In List,

we can take portions (including the lower but not the upper limit). List slicing is the method of splitting a subset of a list, and the indices of the list objects are also used for this. e.g. if my List name is ***products***

From the first element up to a given position: products[:3]  
From a given position until the last element: products[2:]  
Between two given positions in the list: products[2:4]  
all the members of the list: products[:]

Python lists are upper-bound exclusive, and this means that the last index during list slicing is usually ignored. That is why, for a list [3, 22, 30, 5.3, 20]

# Tuple

Tuples are used to hold together multiple objects. Think of them as similar to lists, but without the extensive functionality that the list class gives you. One major feature of tuples is that they are immutable like strings i.e. you cannot modify tuples. However, you can take portions of existing tuples to make new tuples. list are declared in square brackets and can be changed while **tuple is declared in parentheses**

**Tuple indexing and splitting**

The indexing and slicing in tuple are similar to lists. The indexing in the tuple starts from 0 and goes to length(tuple) — 1.

The items in the tuple can be accessed by using the slice operator. Python also allows us to use the colon operator to access multiple items in the tuple.

Unlike lists, the tuple items can not be deleted by using the **del** keyword becasuse tuples being immutable. To delete an entire tuple, we can use the **del** keyword with the tuple name.

1. Are strings mutable in python? Suppose you have a string “I+Love+Python”, write a small code to replace ‘+’ with space in python.

Answer: An immutable object is an object that is not changeable and its state cannot be modified after it is created.

In Python, a string is immutable. You cannot overwrite the values of immutable objects.

It’s not modifying the string object; it’s creating a new string object.

**Immutability may be used to ensure that an object remains constant throughout your program. The values of mutable objects can be changed at any time and place, whether you expect it or not.**

**Program:**

git\_repo= 'I ' + ' Love ' + ' Python'

print('Git repository: ', git\_repo)

Output:

Git repository: I Love Python

1. What does the function **ord()** do in python? Explain with example. Also, write down the function for getting the datatype of a variable in python.

Answer: The **ord() function** returns the number representing the unicode code of a specified character.

The **ord()** function in **Python** accepts a string of length 1 as an argument and returns the unicode code point representation of the passed argument. For example **ord**('B') returns 66 which is a unicode code point value of character 'B'.

Example:

# unicode code point of integer

print("The ASCII value of 9 is",ord('9'))

# unicode code point of alphabet

print("The ASCII value of B is",ord('B'))

# unicode code point of special character

print("The ASCII value of ! is",ord('!'))

Type()

a = 12

print(type(a))

# Q14 and Q15 are programming questions. Answer them in Jupyter Notebook.

1. Write a python program to solve a quadratic equation of the form 𝑎𝑥2 + 𝑏𝑥 + 𝑐 = 0. Where a, b and c are to be taken by user input. Handle the erroneous input, such as ‘a’ should not be equal to 0.

Program:

# Solve the quadratic equation ax\*\*2 + bx + c = 0

# import complex math module

import cmath

a = 1

b = 5

c = 6

# calculate the discriminant

d = (b\*\*2) - (4\*a\*c)

# find two solutions

sol1 = (-b-cmath.sqrt(d))/(2\*a)

sol2 = (-b+cmath.sqrt(d))/(2\*a)

print('The solution are {0} and {1}'.format(sol1,sol2))

Output: The solution are (-3+0j) and (-2+0j)

15.Write a python program to find the sum of first ‘n’ natural numbers without using any loop. Ask user to input the value of ‘n’.

Program:

n=16

n\*(n+1)/2

print("enter the value")

enter the value

12