WORKSHEET 2 PYTHON

Q1 to Q7 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? A) list B) struct C) tuple C) set

Ans: B) struct

2. Which of the following is an invalid variable name in python?

A) _init_ B) no_1 C) 1_no D) _1

Ans: D) _1

3. Which one of the following is a keyword in python? A) in B) _init_ C) on D) foo

Ans: in

4. In which of the following manner are the operators of the same precedence executed in python? A) Left to Right B) BODMAS C) Right to Left D) None of these.

Ans: A) Left to Right

5. 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

Ans: C) iv – iii – ii – I

6. (28//6)**3/3%3 = ? A) 7.1111… B) 0 C) 0.3333… D) 1

Ans: C) 0.3333…

7. a = input("Enter an integer"). What will be the data type of a? A) int B) str C) float D) double

Ans: B) str

Q8 and Q10 have multiple correct answers. Choose all the correct options to answer your question.

8. Which of the following statements are correct? A) Division and multiplication have same precedence in python B) Python's operators' precedence is based on PEDMAS C) Python's operators' precedence is based on VBODMAS D) In case of operators' having the same precedence, the one on the left side is executed first.

Ans: B) Python's operators' precedence is based on PEDMAS

D) In case of operators' having the same precedence, the one on the left side is executed first.

9. 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

Ans: A) abc = 1,000,000

D) a_b_c = 1,000,000

10. Which of the following is not equal to x16 in python? A) x**4**4 B) x**16 C) x^16 D) (x**4)**4

Ans: A) x**4**4

Q11 to Q13 are subjective questions, answer them briefly

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

Ans:

List	**Tuple**	**Set**	**Dictionary**
List is a non-homogeneous data structure that stores elements in single rows and multiple rows and columns.	A tuple is a non-homogeneous data structure that can store a single row and multiple rows and columns.	The set data structure is also non-homogeneous but only stores one row.	A dictionary is another type of non-homogeneous data structure that stores key-value pairs.
The list is represented by [].	The tuple is represented by ().	The set is represented by {}.	The dictionary is represented by {}.
Lists are mutable, which means they can be changed.	The tuple is immutable, which means it cannot be changed.	The set is mutable, which means that we can change it. However, no elements are duplicated.	The dictionary is mutable. However, keys are not duplicated.
The list is ordered.	The tuple has been ordered.	The set is unordered.	The dictionary has been ordered (Python 3.7 and above)
Lists can be nested among themselves.	Tuple can make use of nested among all.	Sets can be nested among themselves.	All dictionaries can use nested among themselves.

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

Ans: Strings are not mutable in Python. Strings are a immutable data types which means that its value cannot be updated.

a small code to replace ‘+’ with space in string “I+Love+Python”in python :

```
s="I+Love+Python"

s.replace("+"," ")
```

13. What does the function ord() do in python? Explain with an example. Also, write down the function for getting the data type of a variable in python.

Ans: ord() function returns the Unicode code from a given character. This function accepts a string of unit length as an argument and returns the Unicode equivalence of the passed argument. In other words, given a string of length 1, the ord() function returns an integer representing the Unicode code point of the character when an argument is a Unicode object, or the value of the byte when the argument is an 8-bit string.

Python ord() syntax:

Syntax: *ord(ch)*

Python ord() parameters:

ch – A unicode character

Python ord() example

For example, ord('a') returns the integer 97, ord('€') (Euro sign) returns 8364. This is the inverse of chr() for 8-bit strings and of unichr() for Unicode objects. If a Unicode argument is given and Python is built with UCS2 Unicode, then the character's code point must be in the range [0..65535] inclusive

Note: If the string length is more than one, and a TypeError will be raised. The syntax can be ord("a") or ord('a'), both will give same results.

If the string length is more than one, and a TypeError will be raised. The syntax can be ord("a") or ord('a'), both will give same results.

Example 1: Demonstration **of Python ord() function**

- Python

```
# inbuilt function return an
# integer representing the Unicode code
value = ord("A")

# writing in ' ' gives the same result
value1 = ord('A')

# prints the unicode value
print (value, value1)
```

Output:

```
65 65
```

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

14. Write a python program to solve a quadratic equation of the form ax^2+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.

Ans: $ax^2 + bx + c$
where,
a, b, and c are coefficient and real numbers and also a ≠ 0.
If a is equal to 0 that equation is not valid quadratic equation.

Using the below quadratic formula we can find the root of the quadratic equation.

Ans:

```
#import libraries

import math

def equationroots( a, b, c):

   #calcualte discrimilar

   dis = b * b - 4 * a * c

   sqrt_val = math.sqrt(abs(dis))


   #check condition for discriminant

   if dis>0:

      print("\nreal and different roots:")

      print((-b + sqrt_val)/( 2 * a))

      print((-b - sqrt_val)/( 2 * a))

   elif dis == 0:

      print("\nreal and same roots")

      print(-b/(2*a))

   #when discriminant is less than 0

   else:

      print("\nComplex roots")

      print( - b / ( 2 * a),"+i",sqrt_val)

      print(- b/ ( 2 * a),"-i",sqrt_val)
print("Quadratic equation: a*x*2 + 2*a*b*x + c")


a = int(input("Enter value of a:"))

b = int(input("Enter value of b:"))

c = int(input("Enter value of c:"))

if a==0:

    print("Incorrect equation")

else:

    equationroots( a, b, c)
```

OUTPUT:

```
Quadratic equation: a*x*2 + 2*a*b*x + c
Enter value of a:2
Enter value of b:10
Enter value of c:-24

real and different roots:
1.7720018726587652
-6.772001872658765
```

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

Ans:

```
num = int(input("Enter value of n:"))

print("Sum of first",num,"numbers=",int(num*(num+1)/2))
```

Output:-

```
Enter value of n:3
Sum of first 3 numbers= 6
```