Q1. Using Python script as a calculator Create the variables n, r, p and assign them values 10, 5, and 100 respectively. Then evaluate the following expression in the Python console. A = p (1 + r/ 100)n a. 100 b. 162.89 c. 189 d. None of the above

```
In [1]: n = 10
        r = 5
        p = 100

        A = p * (1 + r / 100) ** n
        print(A)
```

```
162.88946267774432
```

Q2.In a given string format operation, how will you print the given string. A = 10 B = 20 Str = "There are {} students in the class, with {} who play at least one sport." a. print(string.format(a,b)) b. print(string+a+b) c. print(string.format(b,a)) d. None of the above

```
In [2]: A = 10
        B = 20
        string = "There are {} students in the class, with {} who play at least one sport."
        print(string.format(A, B))
```

```
There are 10 students in the class, with 20 who play at least one sport.
```

Q3. In a given sample string, How do you print a double quoted string in between a regular string using the escape character? Sample output = It goes without saying, "Time is Money", and none can deny it. a. print("It goes without saying, \"Time is Money\", and none can deny it.") b. print("It goes without saying, \Time is Money\, and none can deny it.") c. print("It goes without saying" + "Time is Money" + "and none can deny it.") d. None of the above.

ans

The correct way to print the given sample output with a double quoted string in between a regular string using the escape character is

```
In [3]: print("It goes without saying, \"Time is Money\", and none can deny it.")
```

```
It goes without saying, "Time is Money", and none can deny it.
```

Q4. What will be the output of the following code? x = lambda a,b: a//b x(10,3) a. 3.3333333333 b. 3 c. 30 d. 1000

ans

x = lambda a, b: a // b print(x(10, 3))

```
In [5]: x = lambda a, b: a // b
        print(x(10, 3))
```

```
3
```

Q5. What will be the output of the following code A = 10 B = 12 print("Smaller") if A == B else print("Greater") if A < B else print("True") a. True b. Smaller c. Greater d. None of the

ans

B = 12 print("Smaller") if A == B else print("Greater") if A < B else print("True")

```
In [7]: B = 12
        print("Smaller") if A == B else print("Greater") if A < B else print("True")
```

```
Greater
```

Q6. What will be the output of the following code? a. [2 7 3 5 4 6] b. TypeError c. NameError: name 'numpy' is not defined d. None of the above

ANS import numpy as np arr = [2, 7, 3, 5, 4, 6] np.sort(arr)

```
In [8]: import numpy as np
        arr = [2, 7, 3, 5, 4, 6]
        np.sort(arr)
```

```
Out[8]: array([2, 3, 4, 5, 6, 7])
```

Q7. Create a string called 'string' with the value as 'Machine Learning'. Which code(s) is/are appropriate to slice the substring "Learn"? a. string[slice(13,8,1)] b. string[slice(1,8,1)] c. string[8:14] d. string[slice(8,13,1)]

ANS

string = 'Machine Learning' print(string[8:13])

So, the correct option is: c. string[8:14]

```
In [9]: string = "Machine Learning"
        print(string[8:13])
```

```
Learn
```

Q8. Create a sequence of numbers from 10 to 25 and increment by 4. What is the index of the value 18? a. 3 b. 2 c. 0 d. 1

ANS :

The index of the value 18 in the sequence [10, 14, 18, 22] is 2.

sequence = list(range(10, 26, 4)) print(sequence.index(18))

So, the correct option is: b. 2

```
In [10]: sequence = list(range(10, 26, 4))
         print(sequence.index(18))
```

```
2
```

Q9. Which of the following is true with respect to the below codes? a. num1 = num2 b. num1 ≠ num2 c. num1 < num2 d. num1 > num2

ANS

num1 = 10 num2 = 20

if num1 == num2: print("a. num1 = num2") else: print("b. num1 ≠ num2")

if num1 < num2: print("c. num1 < num2")

```
In [11]: num1 = 10
         num2 = 20

         if num1 == num2:
            print("a. num1 = num2")
         else:
            print("b. num1 ≠ num2")

         if num1 < num2:
            print("c. num1 < num2")
```

```
b. num1 ≠ num2
c. num1 < num2
```

Q10.A Python NameError exception is raised when: - a. Trying to access a variable which has not been defined b. Trying to access a key in a dictionary that does not exist c. Accessing a column with misspelled column name d. Accessing the function from a module that has not been imported

ans

The Python NameError exception is raised when:

a. Trying to access a variable which has not been defined

Here's an example code that raises a NameError:

print(my_variable)

This code will raise a NameError because my_variable has not been defined.

Trying to access a variable which has not been defined

```
In [12]: print(my_variable)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[12], line 1
----> 1 print(my_variable)

NameError: name 'my_variable' is not defined
```

Q11.What type of exception will be raised for the code given below? a. NameError b. KeyError c. ValueError d. AttributeError

ANS

The type of exception raised for the given code will be:

b. KeyError

Here's an example code that raises a KeyError:

my_dict = {'a': 1, 'b': 2} print(my_dict["c"])

```
In [14]: my_dict = {'a': 1, 'b': 2}
         print(my_dict['c'])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[14], line 2
      1 my_dict = {'a': 1, 'b': 2}
----> 2 print(my_dict["c"])

KeyError: 'c'
```

Q12.A FileNotFoundError exception is raised by operating system errors when: - a. Trying to create a file or directory which already exists b. A file or directory is requested but does not exist in the working directory c. Trying to run an operation without the adequate access rights d. A directory operation, os.listdir() is requested on something which is not a directory

ANS

A FileNotFoundError exception is raised by operating system errors when:

b. A file or directory is requested but does not exist in the working directory

Q13.Consider a variable Z. The value of Z is "ID-5632". Data type of Z is: - a. Complex b. Character c. Integer d. Boolean

ANS The data type of Z is:

b. Character

Here's a code example:

Z = "ID-5632" print(type(Z)) # Output: <class 'str'>

```
In [15]: Z = "ID-5632"
         print(type(Z))  # Output: <class 'str'>
```

```
<class 'str'>
```

Q14.Which of the following variable(s) are character data type? a. K= '4' b. J= "Welcome" c. L= "7" d. All of the above

ANS

d. All of the above

K = '4' J = "Welcome" L = "7" print(type(K)) # Output: <class 'str'> print(type(J)) # Output: <class 'str'> print(type(L)) # Output: <class 'str'>

```
In [20]: K = "4"
         J = "Welcome"
         L = "7"
         print(type(K))  # Output: <class 'str'>
         print(type(J))  # Output: <class 'str'>
         print(type(L))  # Output: <class 'str'>
```

```
<class 'str'>
<class 'str'>
<class 'str'>
```

```
In [ ]: K = "4"
        J = "Welcome"
        L = "7"
        print(type(K))  # Output: <class 'str'>
        print(type(J))  # Output: <class 'str'>
        print(type(L))  # Output: <class 'str'>
```

Q15.Choose the symbol/s that does not have the ability to convert any values to string? a. { } b. " " c. [] d. #

ANS

d. #

Q16.Create a dictionary 'Country' that maps the following countries to their capitals respectively: Country India China Japan Qatar France State Delhi Beijing Tokyo Doha Marseilles Find 2 commands to replace "Marseilles" with "Paris" is:

ANS

Here's a code example to create the dictionary and replace "Marseilles" with "Paris":

Creating the dictionary

Country = { "India": "Delhi", "China": "Beijing", "Japan": "Tokyo", "Qatar": "Doha", "France": "Marseilles" }

Replacing 'Marseilles' with 'Paris'

Country["France"] = "Paris" Country.update({"France": "Paris"})

```
In [19]: Country = {
             "India": "Delhi",
             "China": "Beijing",
             "Japan": "Tokyo",
             "Qatar": "Doha",
             "France": "Marseilles"
         }

         Country["France"] = "Paris"
         Country.update({"France": "Paris"})
```

Q17. Create the tuples given below tuple_1 = (1,5,6,7,8) tuple_2 = (8,9,4) Identify which of the following code does not work on a tuple. a. sum(tuple_1) b. len(tuple_2) c. tuple_2 + tuple_1 d. tuple_1[3] = 45

ANS

The code that does not work on a tuple is:

d. tuple_1[3] = 45

tuple_1 = (1, 5, 6, 7, 8) tuple_2 = (8, 9, 4)

Check if sum works

```
In [23]: tuple_1 = (1, 5, 6, 7, 8)
         tuple_2 = (8, 9, 4)

         # Check if sum works
         try:
          sum_result = sum(tuple_1)
          print(f"sum(tuple_1) = {sum_result}")
         except TypeError as e:
          print(f"sum(tuple_1) throws error: {e}")

         # Check if len works
         tuple_length = len(tuple_2)
         print(f"len(tuple_2) = {tuple_length}")

         # Check if concatenation works
         combined_tuple = tuple_1 + tuple_2
         print(f"tuple_1 + tuple_2 = {combined_tuple}")

         # Try to modify element (will raise error)
         try:
          tuple_1[3] = 45
          print(f"tuple_1[3] changed to 45")
         except TypeError as e:
          print(f"Modifying tuple element throws error: {e}")
```

Check if sum works

try: sum_result = sum(tuple_1) print(f"sum(tuple_1) = {sum_result}") except TypeError as e: print(f"sum(tuple_1) throws error: {e}")

Check if len works

tuple_length = len(tuple_2) print(f"len(tuple_2) = {tuple_length}")

Check if concatenation works

combined_tuple = tuple_1 + tuple_2 print(f"tuple_1 + tuple_2 = {combined_tuple}")

Try to modify element (will raise error)

try: tuple_1[3] = 45 print(f"tuple_1[3] changed to 45") except TypeError as e: print(f"Modifying tuple element throws error: {e}")

a. sum(tuple_1): Calculates the sum of the elements in tuple_1. b. len(tuple_2): Returns the length (number of elements) of tuple_2. c. tuple_2 + tuple_1: Concatenates the two tuples, creating a new tuple containing all elements from both.

```
sum(tuple_1) = 27
len(tuple_2) = 3
tuple_1 + tuple_2 = (1, 5, 6, 7, 8, 8, 9, 4)
Modifying tuple element throws error: 'tuple' object does not support item assignment
```

Q18. How many elements in the following data structure?

s={1,2,3,4,4,4,5,6}

ANS

The data structure s has 6 elements.

s is a set, as indicated by the curly braces {}. Sets are unordered collections of unique elements.

```
In [24]: def find_pythagorean_triplets(N):
             """
             Finds all Pythagorean triplets of triangles whose sides are no greater than N.

             Args:
                 N: A natural number.

             Returns:
                 A list of tuples representing Pythagorean triplets (a, b, c), where a, b, and c are the sides of the triangle.
             """

             triplets = []
             for a in range(1, N + 1):
                 for b in range(a, N + 1):
                     c = (a**2 + b**2)**0.5
                     if c.is_integer() and c <= N:
                         triplets.append((a, b, int(c)))
             return triplets

         # Example usage
         N = 20  # Change this value to find triplets for different N
         triplets = find_pythagorean_triplets(N)

         print(f"Pythagorean triplets for N = {N}:")
         for triplet in triplets:
             print(f"\t- {triplet}")
```

```
Pythagorean triplets for N = 20:
        - (3, 4, 5)
        - (5, 12, 13)
        - (6, 8, 10)
        - (8, 15, 17)
        - (9, 12, 15)
        - (12, 16, 20)
```

```
In [ ]:
```