Python Assignment 1

June 25, 2023

[33]:

```
#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.
#   =   (1 +  / 100)n
# a. 100
# b. 162.89
# c. 189
# d. None of the above

n = 10
r = 5
p = 100

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

print('d. None of the above')
```

```
d. None of the above
```

[32]:

```
# 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

A = 10
B = 20

# print('There are {} students in the class, with {} who play at least one␣
 ↪sport'.format(B,A))
print('c. print(string.format(b,a))')
```

```
c. print(string.format(b,a))
```

[30]:

```python
# 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.

# print("It goes without saying, \"Time is Money\", and none can deny it.")
print('a. print("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.")
```

[34]:

```python
# 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

# x = lambda a,b : a//b
# print(x(10,3))
print('b. 3')
```

```
b. 3
```

[28]:

```python
# 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 above

# print("Smaller") if A == B else print("Greater") if A < B else print("True")
print('c. Greater')
```

```
c. Greater
```

[27]:

```python
# 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
print('d. None of the above')
```

```
d. None of the above
```

[37]:
```
# 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)]

# string = 'Machine Learning'
# sti = string[slice(8,13,1)]
print('d. string[slice(8,13,1)]')
```

```
d. string[slice(8,13,1)]
```

[2]:
```
# 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

print('b. 2')
```

```
b. 2
```

[3]:
```
# 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

print('a. num1 = num2')
```

```
a. num1 = num2
```

[4]:
```
# 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

print('a. Trying to access a variable which has not been defined')
```

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

[5]:

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

print('c. ValueError')
```

```
c. ValueError
```

[6]:

```
# 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

print('b. A file or directory is requested but does not exist in the working␣
 ↪directory')
```

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

[7]:

```
# 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

z = "ID-5632"
print(type(z))
```

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

[9]:

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

print('a. K= "4"\nc. L= "?"')
```

```
a. K= "4"
c. L= "?"
```

[10]:

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

print('b. " "\nd. #')
```

```
b. " "
d. #
```

[24]:

```
# 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:

country = {'India':'Delhi','China':'Beijing','Japan':'Tokyo','Quatar':'Doha','France':'Marseilles'}
country['France'] = 'Paris'
country.popitem()
country['France']='Paris'
for i, j in country.items():
    print(i,j)
```

```
India Delhi
China Beijing
Japan Tokyo
Quatar Doha
France Paris
```

[30]:

```
# 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

print('d. tuple_1[3] = 45')
```

```
d. tuple_1[3] = 45
```

[33]:

```
# Q18. How many elements in the following data structure?
print('6')
```

```
6
```

[36]:

```python
# Q19.Write a function which finds all pythagorean triplets of triangles whose sides are no greater than a natural number N.


def pythtrip(N):
    trip =[]
    for i in range(1,N):
        for j in range(1,N):
            for k in range(1,N):
                if i**2 + j**2==k**2:
                    trip.append([i,j,k])
    return trip

give = pythtrip(45)
print(give)
```

```
[[3, 4, 5], [4, 3, 5], [5, 12, 13], [6, 8, 10], [7, 24, 25], [8, 6, 10], [8, 15,
17], [9, 12, 15], [9, 40, 41], [10, 24, 26], [12, 5, 13], [12, 9, 15], [12, 16,
20], [12, 35, 37], [15, 8, 17], [15, 20, 25], [15, 36, 39], [16, 12, 20], [16,
30, 34], [18, 24, 30], [20, 15, 25], [20, 21, 29], [21, 20, 29], [21, 28, 35],
[24, 7, 25], [24, 10, 26], [24, 18, 30], [24, 32, 40], [28, 21, 35], [30, 16,
34], [32, 24, 40], [35, 12, 37], [36, 15, 39], [40, 9, 41]]
```