MCQ

1. What will be the output of the following code snippet?

```
def func(a, b):
   return b if a == 0 else func(b % a, a)
print(func(30, 75))
```

a) **10**
b) **20**
c) **15**
d) **0**

Answer is:- c) 15

Explanation
1. First, call the function with func(30,75)
2. If ‘a’ is not equal to 0 then go to else part with arguments(75%30,30), which is equal to func(15,30). So, here we got a=15 then , repeat the procedure until ‘a’ is equal to 0.
3. By using the value a=15 in funcion again with func(30 % 15,15), which is equal to func(0, 15)
4. Since ‘a’ is equal to 0, it returns the value of b.
Therefore, the output b is 15.

```
2 numbers = (4, 7, 19, 2, 89, 45, 72, 22)
sorted_numbers = sorted(numbers)
even = lambda a: a % 2 == 0
even_numbers = filter(even, sorted_numbers)
print(type(even_numbers))
```

a) **Int**
b) **Filter**
c) **List**
d) **Tuple**

Answer is :- b) Filter

Explanaton
1. numbers = (4, 7, 19, 2, 89, 45, 72, 22) → It defines a tuple.
2. sorted_numbers = sorted(numbers) → It creates a new list for sorted numbers.
3. even = lambda a: a % 2 == 0 →It defines a lambda function, the variable even returns True for even numbers and False for odd numbers.
4. even_numbers = filter(even, sorted_numbers) →It uses the filter function to filter even numbers from sorted_numbers.
5. print(type(even_numbers)) →Prints the type of the object resulting from the filter operation.
Therefore, the output is <class 'filter'>

3) As what datatype are the *args stored, when passed into

a) Tuple

b) List

c) Dictionary

d) none

Answer is:- a) Tuple

Explanation
Any additional positional arguments are gathered into a tuple when a function declaration uses *args.

```
4) set1 = {14, 3, 55}
   set2 = {82, 49, 62}
   set3={99,22,17}
print(len(set1 + set2 + set3))
```

a) 105
b) 270
c) 0
d) Error

Answer is:- d) Error

Explanation
Because sets are not supported by the + operator, it will result in a TypeError. Therefore, to combine the sets, use the | operator or the union technique.

5) What keyword is used in Python to raise exceptions?

a) raise

b) try

c) goto

d) except

Answer is:- a) raise

Explanaton

To raise an exception directly in Python, use the raise keyword.

6) Which of the following modules need to be imported to handle date time computations in Python?

a) timedate

b) date

c) datetime

d) time

Answer is:- c) datetime

Explanation

We must import the datetime module if we want to perform time and date computations in Python.

7) What will be the output of the following code snippet?

print(43 + (7 + 5)**(1 + 1))**

a) 248
b) 169
c) 208
d) 233

Answer is:- c) 208

Explanation

64+144=208

8) Which of the following functions converts date to corresponding time in Python?

a) strptime

b) strftime

c) both a) and b)

d) None

Answer is:- b) strftime

Explanation

strftime() function in Python converts a date to its corresponding time in Python. If we want to convert a date to a string representing the time, would typically use strftime.

9) The python tuple is______in nature.

a) mutable

b)immutable

c) unchangeable

d) none

Answer is:- a) immutable

Explanation
The Python tuple is an immutable type of data, which means that its elements can not be modified after they are created.

10) The______is a built-in function that returns a range object that consists series of integer numbers, whichwe can iterate using a for loop.

A. range()
B. set()
C. dictionary{}
D. None of the mentioned above

Answer is :- A. range()

Explanation
The range() function in Python is a builtin function that returns a range object representing a sequence of numbers. In a for loop, it is commonly used for iterating over a sequence of numbers.

Question 11

Amongst which of the following is a function which does not have any name?

A. Del function
B. Show function
C. Lambda function
D. None of the mentioned above

Answer: C. Lambda function

Explanation
An anonymous function in Python that can have a number of input parameters, but only one expression, is a lambda function. A lambda keyword is used to define this function, but it doesn't have a name like normal functions.

Question 12

The module Pickle is used to____.

A. Serializing Python object structure
B. De-serializing Python object structure
C. Both A and B
D. None of the mentioned above

Answer is:- C. Both A and B

Explanation
For the serialisation and deserialisation of Python object structures, the pickle module is used in Python. Serializing and deserialising are processes for converting Python objects to byte streams, as opposed to reconstructing the original object from a stream of bytes.

Question 13

Amongst which of the following is / are the method of convert Python objects for writing data in a binary file?

A. set() method
B. dump() method
C. load() method
D. None of the mentioned above

Answer is:- B. dump() method

Explanation
Dump() is a Python pickle module, and it's useful for converting Python objects to binary format so they can be written into a binary file. In particular, when saving data structures to a file in a binary format, it is commonly used for serialization.

14) Amongst which of the following is / are the method used to unpickling data from a binary file?

A. load()
B. set() method
C. dump() method
D. None of the mentioned above

Answer is:- A. load()

Explanation
load() method is used to unpickle data from binary file that has been compressed.

15.

A text file contains only textual information consisting of____.

A. Alphabets
B. Numbers
C. Special symbols
D. All of the mentioned above

Answer is:- D. All of the mentioned above

Explanation
A combination of the alphabet, numbers and special symbols can be included in text file. It's a type of file that doesn't contain formatting or binary data, and it contains just text information.

16.

Which Python code could replace the ellipsis (...) below to get the following output? (Select all that apply.)

```
captains = {

    "Enterprise": "Picard",

    "Voyager": "Janeway",

    "Defiant": "Sisko",

 }
```

```
Enterprise Picard,

   Voyager Janeway

   Defiant Sisko
```

a) for ship, captain in captains.items():

```
    print(ship, captain)
```

b) for ship in captains:

```
    print(ship, captains[ship])
```

c) for ship in captains:

```
    print(ship, captains)
```

d) both a and b

Answer is :- d) both a and b

Explanation
Both options will produce the specified output-

```
Enterprise Picard,
Voyager Janeway
 Defiant Sisko
```

17)

Which of the following lines of code will create an empty dictionary named captains?

a) captains = {dict}
b) type(captains)
c) captains.dict()
d) captains = {}

Answer is:- d) captains = {}

Explanation
The code will create a blank dictionary with the name of Captain. The {} syntax is used to denote an empty dictionary in Python

18) Now you have your empty dictionary named captains. It’s time to add some data!

Specifically, you want to add the key-value pairs "Enterprise": "Picard", "Voyager": "Janeway", and "Defiant": "Sisko".

Which of the following code snippets will successfully add these key-value pairs to the existing captains dictionary?

a) captains{"Enterprise" = "Picard"}

```
captains{"Voyager" = "Janeway"}

captains{"Defiant" = "Sisko"}
```

b) captains["Enterprise"] = "Picard"

```
captains["Voyager"] = "Janeway"

captains["Defiant"] = "Sisko"
```

c) captains = {

```
    "Enterprise": "Picard",

    "Voyager": "Janeway",

    "Defiant": "Sisko",

}
```

d) None of the above

Answer is :- b)

```
captains["Enterprise"] = "Picard"

captains["Voyager"] = "Janeway"

captains["Defiant"] = "Sisko"
```

Explanation

Both option b and c are correct to achieve the desired result. So, option c is also a valid way to create a dictionary with specified key-value pairs.

19) You’re really building out the Federation Starfleet now! Here’s what you have:

```
captains = {

    "Enterprise": "Picard",

    "Voyager": "Janeway",

    "Defiant": "Sisko",

    "Discovery": "unknown",

}
```

Now, say you want to display the ship and captain names contained in the dictionary, but you also want to provide some additional context. How could you do it?

a)

```
for item in captains.items():

    print(f"The [ship] is captained by [captain].")
```

b)

```
for ship, captain in captains.items():

    print(f"The {ship} is captained by {captain}.")
```

c)

```
for captain, ship in captains.items():

    print(f"The {ship} is captained by {captain}.")
```

d) All are correct

Answer is:- b)

```
for ship, captain in captains.items():
        print(f"The {ship} is captained by {captain}.")
```

Explanation

Output:

```
The Enterprise is captained by Picard.
The Voyager is captained by Janeway.
The Defiant is captained by Sisko.
The Discovery is captained by unknown.
```

20)

You’ve created a dictionary, added data, checked for the existence of keys, and iterated over it with a `for` loop. Now you’re ready to delete a key from this dictionary:

```
captains = {

    "Enterprise": "Picard",

    "Voyager": "Janeway",

    "Defiant": "Sisko",

    "Discovery": "unknown",
           }
```

What statement will remove the entry for the key `"Discovery"`?

a) del captains
b) captains.remove()
c) del captains["Discovery"]
d) captains["Discovery"].pop()

Answer is :- c) del captains["Discovery"]

Explanation
This statement will remove the entry for the key "Discovery" from the captains dictionary.