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 : c) 15

The given code snippet is an implementation of the Euclidean algorithm to find the greatest common divisor (GCD) of two numbers. Let's analyze it step by step.

The function func(a, b) is defined to return b if a is 0, otherwise it calls itself recursively with the arguments (b % a, a).

The call is func(30, 75), so the execution proceeds as follows:

1. func(30, 75)
   - a is 30, b is 75
   - Since a is not 0, it calls func(75 % 30, 30)
2. func(15, 30)
   - a is 15, b is 30
   - Since a is not 0, it calls func(30 % 15, 15)
3. func(0, 15)
   - a is 0, b is 15
   - Since a is 0, it returns b, which is 15

Therefore, the output of the code snippet is 15.

So, the correct answer is: c) 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: b) Filter

The filter function in Python returns an iterator, not a list, tuple, or integer. The type of the object returned by filter is filter.

Let's break down the code:

1. numbers = (4, 7, 19, 2, 89, 45, 72, 22): This is a tuple of numbers.
2. sorted_numbers = sorted(numbers): This sorts the numbers and returns a list [2, 4, 7, 19, 22, 45, 72, 89].
3. even = lambda a: a % 2 == 0: This defines a lambda function that returns True if a number is even.
4. even_numbers = filter(even, sorted_numbers): This applies the even lambda function to each element of sorted_numbers and returns an iterator (of type filter) containing only the even numbers.
5. print(type(even_numbers)): This prints the type of even_numbers.

Therefore, the type of even_numbers is filter.

The correct answer is: b) Filter

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

a) Tuple

b) List

c) Dictionary

d) none

Answer: a)Tuple

In Python, when we use `*args` in a function definition, it allows the function to accept an arbitrary number of positional arguments. These arguments are stored as a tuple.

Therefore, the correct answer is: a) Tuple

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:d) Error

In Python, the + operator cannot be used to concatenate sets. Attempting to do so will result in a TypeError. Sets in Python do not support concatenation using the + operator.

Therefore, the correct answer is: d) Error

5) What keyword

is used in Python to

raise exceptions?

a) raise

b) try

c) goto

d) except

Answer: a) raise

In Python, the keyword used to raise exceptions is raise.

Therefore, the correct answer is: a) raise

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: c) datetime

To handle date and time computations in Python, you typically use the `datetime` module. Therefore, the correct answer is: c) datetime

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: c) 208

Let's break down the above expression step by step:

1. `4**3` calculates 4 raised to the power of 3, which is 4×4×4=64.
2. Inside the parentheses, `(7 + 5)` calculates 7+5=12.
3. The expression `(1 + 1)` calculates 1+1=2.
4. Now, the expression `(7 + 5)**(1 + 1)` becomes `12**2`, which is 12×12=144.
5. Adding these two results together, 64+144=208.

Therefore, the output of the code snippet is: c) 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: b)strftime

In Python:

- `strptime` is used to parse a string representing a date and/or time and convert it to a `datetime` object.
- `strftime` is used to format a `datetime` object into a string representing a date and/or time.

Since the question is about converting a date to a corresponding time (i.e., formatting a date/time object into a string representation), the correct function to use is `strftime`.

Therefore, the correct answer is: b) strftime

9) The python tuple is _____ in nature.

a) mutable

b)immutable

c)unchangeable

d) none

Answer: b)immutable

In Python, tuples are immutable, meaning once a tuple is created, its elements cannot be changed, added, or removed.

Therefore, the correct answer is: b) immutable

10) The ___ is a built-in function that returns a range object that consists series of integer numbers, which we can iterate using a for loop.

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

Answer: A. range()

The range() function is a built-in function in Python that returns a range object consisting of a series of integer numbers. This range object can be iterated using a for loop.

Therefore, the correct answer is: A. range()

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

In Python, a lambda function is an anonymous function, meaning it does not have a name. It is defined using the lambda keyword.

Therefore, the correct answer is: C. Lambda function

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: C. Both A and B

The pickle module in Python is used for both serializing (converting a Python object into a byte stream) and de-serializing (converting a byte stream back into a Python object) Python object structures.

Therefore, the correct answer is: C. Both A and B

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: B. dump() method

To convert Python objects for writing data in a binary file, the dump() method is used. This method is part of the pickle module and is used to serialize an object and write it to a binary file.

Therefore, the correct answer is: B. dump() method

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: A.load()

To unpickle data from a binary file, the load() method is used. This method is part of the pickle module and is used to deserialize a byte stream from a binary file back into a Python object.

Therefore, the correct answer is: A. load()

15. **A text file contains only textual information consisting of ___.**

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

Answer: D. All of the mentioned above

A text file can contain alphabets, numbers, and special symbols. It includes any character that can be represented in text form.

Therefore, the correct answer is: D. All of the mentioned above

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: d) both a and b

To produce the desired output, both option a) and option b) will work. Let's analyze each option:

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

This iterates over the items in the captains dictionary, unpacking each key-value pair into ship and captain, and then prints them.

b)for ship in captains:

print(ship, captains[ship])

This iterates over the keys in the captains dictionary and uses the key to access and print the corresponding value.

Both of these options will produce the desired output:

Enterprise Picard
Voyager Janeway
Defiant Sisko

Therefore, the correct answer is: d) both a and b

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: d) captians = {}

To create an empty dictionary in Python, we can use the following line of code

```
captains = {}
```

where {} denotes empty dictionary

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:

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

captains["Voyager"] = "Janeway"

captains["Defiant"] = "Sisko"

To add key-value pairs to an existing dictionary named captains, you should use the following syntax:

```
captains["Enterprise"] = "Picard"
captains["Voyager"] = "Janeway"
captains["Defiant"] = "Sisko"
```

This code correctly adds the specified key-value pairs to the existing captains dictionary.

Option c) would create a new dictionary rather than adding to the existing one.

Therefore, the correct answer is:

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

captains["Voyager"] = "Janeway"

captains["Defiant"] = "Sisko"

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

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

To display the ship and captain names contained in the dictionary with additional context, the correct syntax would be to iterate over the key-value pairs using a loop and format the output string accordingly.

Option b) correctly uses for ship, captain in captains.items(): to unpack the key-value pairs and formats them with f-strings.

Here's the correct code snippet:

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

This will correctly print each ship and its captain.

Therefore, the correct answer is: b) for ship, captain in captains.items(): print(f"The {ship} is captained by {captain}.")

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: c) del captains[“Discovery”]

To remove the entry for the key "Discovery" from the captains dictionary, you should use the del statement with the specific key.

The correct statement is:

```
del captains["Discovery"]
```

Therefore, the correct answer is: c) del captains["Discovery"]