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 initial call is func(30,75)

since a is not equal to 0,it goes into the recursive call with arguments(75%30,30), which is equivalent to func(15,30).

Again, since a is not equal to 0, it goes into another recursive call with arguments (30 % 15, 15), which is equivalent to func(0, 15)

Now, since a is equal to 0, it returns the value of b, which is 15. Therefore, the output of the code snippet for the input (30, 75) 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

numbers = (4, 7, 19, 2, 89, 45, 72, 22) − Defines a tuple of numbers.

sorted_numbers = sorted(numbers) − Creates a new list containing the sorted numbers.

even = lambda a: a % 2 == 0 − Defines a lambda function even that returns True for even numbers and False for odd numbers.

even_numbers = filter(even, sorted_numbers) − Uses the filter function to filter out the even numbers from sorted_numbers.

print(type(even_numbers)) − Prints the type of the object resulting from the filter operation. Therefore, the output of the code will be:

```
<class 'filter'>
```

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

a) Tuple

b) List

c) Dictionary

d) none

Answer: When *args is used in a function definition, it collects any additional positional arguments into 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

This will result in a TypeError because the + operator is not supported for sets. To fix this, you can use the union method or the | operator to combine the sets

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

a) raise

b) try

c) goto

d) except

Answer: a) raise

The raise keyword in Python is used to explicitly raise an exception.

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 need to import the datetime module.

7) What will be the output of the following code snippet? print(4**3 + (7 + 5)**(1 + 1))

a) 248

b) 169

c) 208

d) 233

Answer: c) 208

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: b) strftime

The strftime function in Python is used to convert a date object into a string representing the date, controlled by format codes

9) The python tuple is in nature.

a) mutable b)immutable c)unchangeable

d) none

Answer: a) immutable

A Python tuple is an immutable data type, meaning its elements cannot be changed after the tuple is created

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 in Python is a built−in function that returns a range object representing a sequence of numbers. It is commonly used for iterating over a sequence of numbers in a for loop

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

A lambda function in Python is an anonymous function that can have any number of input parameters but can only have one expression. It is defined using the lambda keyword and doesn't have a name like a regular function

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 serializing and deserializing Python object structures. Serializing refers to the process of converting a Python object into a byte stream, and deserializing is the reverse process of reconstructing the original object from a byte stream

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

The dump() method is part of the pickle module in Python and is used to convert Python objects into a binary format and write them to a file. This is commonly used for serialization, especially when saving data structures to a file in a binary format.

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 a combination of alphabets, numbers, and special symbols. It is a type of file that stores plain text information without any formatting or binary data.

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

Both of these 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: d) captains = {}

This line of code creates an empty dictionary named captains. 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: 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: b) for ship, captain in captains.items(): print(f"The {ship} is captained by {captain}.") 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:c) del captains["Discovery"]

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