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

Explanation = This function will provide output in first iteration as 75%30 = 15
Second iteration 30%15 as 0 so output 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

Explanation = Its a filter class as it uses filter class and it returns filter.

3) As what datatype are the *args stored, when passed into
a) Tuple
b) List
c) Dictionary
d) none
Explanation = Args is stored as tuple in python

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

Explanation

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

c) goto
d) except
Explanation : Raise keyword is used to create exception in Python

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

Explanation
The `datetime` module provides classes for manipulating dates and times in both simple and complex ways.

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

Explanation 4∗∗34**34∗∗3: 4∗∗3=4×4×4=644**3 = 4 \times 4 \times 4 = 644∗∗3=4×4×4=64

1. (7+5)∗∗(1+1)(7 + 5)**(1 + 1)(7+5)∗∗(1+1):
    - **First, compute the inner addition: 7+5=127 + 5 = 127+5=12**
    - **Then, compute the exponent: 1+1=21 + 1 = 21+1=2**
    - Finally, compute 12∗∗212**212∗∗2: 12×12=14412 \times 12 = 14412×12=144
2. **Add the two results together: 64+144=20864 + 144 = 20864+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

Explanation **Both `strptime` and `strftime` are involved in date and time conversions in Python.**

9) The python tuple is _____ in nature.
a) mutable
b)immutable
c)unchangeable
d) none

Explanation A **Python tuple is immutable, meaning that once it is created, its elements cannot be changed.**

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

A. range()

B. set()

C. dictionary{}

D. None of the mentioned above

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

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

Explanation **A lambda function in Python is an anonymous function, meaning it does not have a name. It is defined using the `lambda` keyword.**

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

Explanation

The `pickle` module in Python is used for serializing and de-serializing Python object structures.

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

Explanation **The `dump()` method is used to serialize Python objects and write them to a binary file.**

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

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

A. Alphabets

B. Numbers

C. Special symbols

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

Explanation **Option a: `for ship, captain in captains.items(): print(ship, captain)` This iterates through the dictionary items, unpacking each key-value pair into `ship` and `captain`, and then prints them.**

Option b: `for ship in captains: print(ship, captains[ship])` This iterates through the dictionary keys, and for each key (`ship`), it accesses the corresponding value (`captains[ship]`) and prints them.

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 = {}

Explanation d: **This line of code correctly creates an empty dictionary named `captains`**

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

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].")

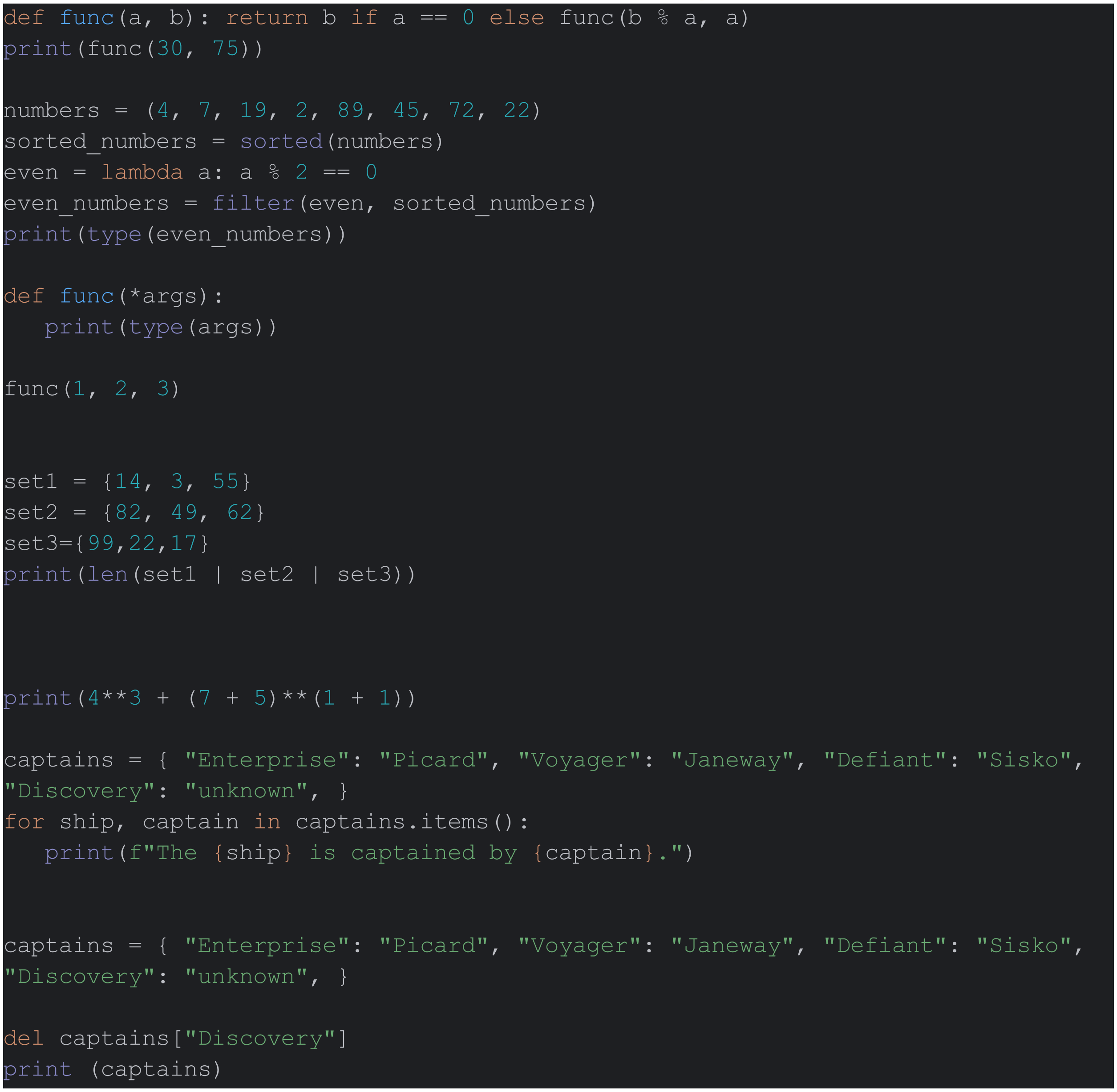
b) for ship, captain in captains.items(): print(f"The {ship} is captained by {captain}.")

c) for captain, ship in captains.items(): print(f"The {ship} is captained by {captain}.") d) All are correct

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()

All executed code in python is here for the above answers 👍



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

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))

def func(*args):
   print(type(args))

func(1, 2, 3)


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



print(4**3 + (7 + 5)**(1 + 1))

captains = { "Enterprise": "Picard", "Voyager": "Janeway", "Defiant": "Sisko",
"Discovery": "unknown", }
for ship, captain in captains.items():
   print(f"The {ship} is captained by {captain}.")


captains = { "Enterprise": "Picard", "Voyager": "Janeway", "Defiant": "Sisko",
"Discovery": "unknown", }

del captains["Discovery"]
print (captains)
```

```
C:\Users\parag\PycharmProjects\pythonProject\.venv\Scripts\python.exe
"C:\Users\parag\PycharmProjects\pythonProject\Internship 1\GCDgreatestcoomon divisor.py"
15
```

```
<class 'filter'>
<class 'tuple'>
9
208
The Enterprise is captained by Picard.
The Voyager is captained by Janeway.
The Defiant is captained by Sisko.
The Discovery is captained by unknown.
{'Enterprise': 'Picard', 'Voyager': 'Janeway', 'Defiant': 'Sisko'}

Process finished with exit code 0
```