1. Write a Python program to Extract Unique values dictionary values?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def unique\_dict(my\_dict):

# assuming our dictionary is named as "my\_dict"

initial\_run = {i[1]:i[0] for i in my\_dict.items()}

unique = {i[1]:i[0] for i in initial\_run.items()}

print(unique)

try:

example ={"a": 1, "b":1, "c":2, "d": 3, "e":4, "f" :1, "g":1 }

unique\_dict(example)

lg.info("""Function unique\_dict() has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python program to find the sum of all items in a dictionary?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def sum\_of\_dict(dic):

#sum variable

sumof=0

#iterate through values

for i in dic.values():

if type(i) == int or type(i) == float:

sumof += i

print(sumof)

try:

dic={ 'a':300, 'b':145, 'c':55, 'd': "hool", 'e': '98','f': 87 }

sum\_of\_dict(dic)

lg.info("""Function has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python program to Merging two Dictionaries?

Ans

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def add\_dicts(dict\_1, dict\_2):

new\_dict = list(dict\_1.items())+list(dict\_2.items())

result = {i[0]:i[1] for i in new\_dict}

print(result)

try:

dic1={ 'a':300, 'b':145, 'c':55, 'd': "hool", 'e': '98','f': 87 }

dic2 = {"charlie": "boy", "bill": 87655}

add\_dicts(dic1,dic2)

lg.info("""Function has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python program to convert key-values list to flat dictionary?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def list\_to\_dict(lst):

final = {lst[i]: lst[i + 1] for i in range(0, len(lst), 2)}

print( final)

try:

my\_list = ["a", 56, "b", 897,"c", "johhn"]

list\_to\_dict(my\_list)

lg.info("""Class list\_to\_dict() has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python program to insertion at the beginning in OrderedDict?

Ans.

from collections import OrderedDict

my\_dict = OrderedDict()

my\_dict["a"] = 1

my\_dict["b"] = 2

my\_dict["c"] = 3

my\_dict["d"] = 4

my\_dict["e"] = 5

my\_dict["f"] = 6

new\_keys = []

new\_values = []

my\_dict["p"] = 10

my\_dict.move\_to\_end("p", last=False)

print(my\_dict)

1. Write a Python program to check order of character in string using OrderedDict()?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

from collections import OrderedDict

def check\_order(my\_input, my\_pattern):

my\_dict = OrderedDict.fromkeys(my\_input)

pattern\_length = 0

for key,value in my\_dict.items():

if (key == my\_pattern[pattern\_length]):

pattern\_length = pattern\_length + 1

if (pattern\_length == (len(my\_pattern))):

return 'The order of pattern is correct'

return ('The order of pattern is incorrect')

try:

from collections import OrderedDict

check\_order("abcdefghi", "cea")

check\_order("abcdefghi", "cei")

lg.info("""Class special\_search() has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python program to sort Python Dictionaries by Key or Value?

Ans.

main\_dict = {"A": "c1", "B": "a3", "D": "k2", "C": "b4", "E": "e5"}

inverted\_dictionary = {i[1]: i[0] for i in main\_dict.items()}

my\_keys = [i for i in main\_dict]

my\_keys.sort()

sort\_by\_keys = {i:main\_dict[i] for i in my\_keys}

my\_values = [i for i in inverted\_dictionary]

my\_values.sort()

sort\_by\_values = {i:inverted\_dictionary[i] for i in my\_values}

print(sort\_by\_keys)

print(sort\_by\_values)