CSC 120 Lab 09

NOTE: Points ***(+ x)*** indicated for each problem.

1. ***(+5)*** Given the following python code, what is the output (in no particular order)

set1 = set([**"A"**, **"B"**, **"C"**, **"D"**])  
set1.add(**"D"**) **if "C" in** set1:  set1.remove(**"A"**)  
set1.add(**"A"**)  
print (set1)

* 1. {'A', 'B', 'C', 'D'}
  2. {'A', 'B', 'D', 'D', ‘A’}
  3. { 'B',’C’, 'D'}
  4. Error generated

1. ***(+5)*** Given the following python code, what is the output

set1 = set([**"A"**, **"B"**, **"C"**, **"D"**])  
set2 = set([**"A"**, **"U"**, **"V"**])  
set3 = set([**"M"**, **"N"**, **"O"**, **"P"**])  
set4 = set([**"A"**, **"B"**])  
set5 = (set1 & set4) & (set2 ^ set3)  
print(set5)

* 1. {'A', 'B', 'C', 'D'}
  2. **{'A'}**
  3. {'A', 'B'}
  4. None of the above or an error generated

1. ***(+5)*** Given the following python code, how many elements (values are there in set5?

set1 = set([**"A"**, **"B"**, **"C"**, **"D"**])  
set2 = set([**"A"**, **"U"**, **"V"**])  
set3 = set([**"M"**, **"N"**, **"O"**, **"P"**, **"Q"**])  
set4 = set([**"A"**, **"B"**])  
set5 = set1 ^ (set2 - set3)

* 1. 4
  2. 3
  3. 5
  4. None of the above or error generated

1. ***(+5)*** Given the following python code, what is the output

set1 = set([**"A"**, **"B"**, **"C"**, **"D"**])

set4 = set([**"A"**, **"B"**])

set3 = set1 - set4

print(set3)

* 1. {'A', 'B', 'C', 'D'}
  2. {'A', 'B'}
  3. {'C', 'D’'}
  4. Error generated or none of the above

1. ***(+5)*** Given the following python code, what is the output

set1 = set([**"A"**, **"B"**, **"C"**, **"D"**])  
set2 = set([**"A"**, **"U"**, **"V"**])  
set3 = set([**"M"**, **"N"**, **"O"**, **"P"**, **"Q"**])  
set4 = set([**"A"**, **"B"**])

set5 = (set1 & set2) | set4 print(set5)

* 1. {'A', 'B', 'C', 'D'}
  2. {'A', 'B'}
  3. {'C', 'D''}
  4. Error generated or none of the above \*Output is {‘B’, ‘A’}

1. ***(+5)*** Given the following python code, what is the output

set1 = set([**"A"**, **"B"**, **"C"**, **"D"**])  
set2 = set([**"A"**, **"U"**, **"V"**])  
set3 = set([**"M"**, **"N"**, **"O"**, **"P"**, **"Q"**])  
set4 = set([**"A"**, **"B"**])

issubset = set1 <= set4 *# Subset test*print(issubset)  
issuperset = set1 >= set4 *# Superset test*print(issuperset)

* 1. False True
  2. True False
  3. False, False
  4. True True
  5. Error generated or none of the above

1. ***(+5)*** Given the following python code, what is the output

d = {**"A"**: 10, **"B"**:20, **"C"**:30, **"D"**:40 }  
print(d.keys())

* 1. [10, 20, 30, 40]
  2. [‘A’, ‘B’, ‘C’, ‘D’ ]
  3. Error generated or none of the above

1. ***(+5)*** Given the following python code, what is the output

a = (1,[**'a'**, **'b'**])  
b = (**'a'**,**'b'**,1)  
c = a + b  
print(c)

* 1. ( 1 ['a', 'b'], 'a', 'b')
  2. (1, ['a', 'b'], 'a', 'b', 1)
  3. (1, ['a', 'b'] )
  4. Error generated or none of the above

1. ***(+5)*** Given the following sets A and B. Using the Cartesian product A x B create a set that is a function with **three tuples** Note |A x B| == 6 Hint: See the database Theory slides

A = { a, b, c } B = { 1, 2}

--------------AxB = {(a, 1), (b, 1), (c, 1)}------------------------AxB = 3\*1 = 3--------------------------------------------

10. ***(+5)*** Given the following sets A and B. What is the value of the set D = A ∪ B ∩ C

given that

A = { a, b, c } B = {b, c, d } C = {a, b d }

D = ----------{‘b’, ‘a’, ‘d’}--------------------------------

1. Top of Form
2. Bottom of Form

(+40) Write a python program that:

1. (+5)Ask the user for a string andcreates the following dictionary: The values are the letters in the string, with the corresponding key being the place in the string. For example if the user entered the string “CSC120” then create the dictionary d

d = {**'0'**:**'C'**, 1:**'S'**, 2:**'C'**, 3:**'1'**, 4:**'2'**, 5:**'0'**}

1. (+5) Display the length of the dictionary d using a len function In this case the answer is 6
2. (+5) Ask the user for a key(integer). If the key is in the dictionary display the corresponding value. As an example if the user entered 2 display ‘C’ if the user entered a key that was not in the dictionary display “Error key not found”
3. (+5) Ask the user for a value (one of the characters in our example). If the value is in the dictionary display “Value found” otherwise display “Value not found”. As an example if the user entered ‘C’ display “Value found”
4. (+5) Ask the user for a character and add it to the dictionary. As an example, if the user enters ‘X’ then the key should be the next key integer (in this case 6 but in general it should be related to the length of the dictionary ) For this example

d = {**'0'**:**'C'**, 1:**'S'**, 2:**'C'**, 3:**'1'**, 4:**'2'**, 5:**'0', 6:‘X’**}

**see the code to add the name ‘Hopper’ to the dictionary in the Lab09Template code below**

**Display the dictionary**

1. (+15) Ask the user for a string of digits, separated by a comma. Print out the corresponding values for those digits.

For example if the user enters the string “0, 1, 4” then print out “CS2” At issue is **how to handle digits that exceed the range of key values**  for example, in our case, any digits in the range 0…5 should generate a corresponding value but any digit(s) > 5 OR < 0 has to be handled appropriately ***-- I leave that to your decision ( generate an error message or ignore the values) See the last part of the Lab09 template code for an example, on how to handle the string 11, 0, 2 (11 is not a key)***

Tutorial on sets and dictionary can be found at : (*left side in the Table of Contents 5.4 and 5.5)* [**https://docs.python.org/2/tutorial/datastructures.html**](https://docs.python.org/2/tutorial/datastructures.html)

Tutorial on strings can be found at

[***https://en.wikibooks.org/wiki/Python\_Programming/Strings***](https://en.wikibooks.org/wiki/Python_Programming/Strings)

**Try your code on the string “CSC 120ComputingFundamentals”**

See the Lab09Template.py in our Source code tab Duplicated here

Please save your code as LastnameLab09.py

*#lab09Template.py*names = {  
 10:**"Euclid"**, 20:**"Archimedes"**, 30:**"Newton"**, 40:**"Descartes"**,  
 50:**"Fermat"**, 60:**"Turing"**, 70:**"Euler"**, 80:**"Einstein"**,  
 90:**"Boole"**, 100:**"Fibonacci"**, 110:**"Nash"** }  
print(**"1 "**,names.keys()) *# print keys*print(**"2 "**, names.values() )  
*# insert Hopper with a key == 120*names[120] = **"Hopper"**print(**"3 names == "**, names)  
print(**' the length of names is '**, len(names) )  
  
*#use in operator to search for keys as well as non-existence keys*print(**'searching...'**)  
key = 110  
**if** key **in** names.keys():  
 print(**"4 key == "**, key, **' value == '**,names[key])  
**else**:  
 print(**"error key == ', key"**)  
  
s = **"Euler"** *# search for Euler***if** s **in** names.values():  
 print(**" 5 value "**, s,**" found "**)  
**else**:  
 print(**"NO "**, s)  
  
print(**"====ALL KEYS===="**)  
**for** key **in** names:  
 print(key, names[key])  
print(**"==== ALL VALUES===="**)  
**for** values **in** names.values():  
 print(values)  
  
print(**"====string manipulation===="**)  
*# isolating individuals integers using the split() and int() functions*s = (input(**"Enter several integers separated by a comma "**)) *# NOTE cannot convert 110,20,3 into an integer directly*print(**' 6 s =='** , s)  
s = s.split(**','**) *# now split using , as a separator, into individual words*print(**'7 using the split function '**,s, **'length == '**, len(s) ) *# still a string but separated into words  
  
# now convert position 0 in the string to an integers*print(**' 8 int (s[0]) == '**, int(s[0]) ) *# now print it***if** int(s[0] ) **in** names.keys(): *# check if the keys values* print(**"9 key found == "**, key, **' value == '**,names[key])  
**else**:  
 print(**'ERROR key == '**, int(s[0]), **'NOT in key range '**)  
  
*# now the second .... we may need a loop for this !!!!!!*x = int(s[1]) *# now convert position 1 to a string*print(**'10 x == '**, x) *# now print it  
# we need to check if int(s[1]) is with th range of key values***if** x **in** names.keys():  
 print(**"11 key found == "**, x , **' value == '**,names[x])  
**else**:  
 print(**'error key == '**, int(s[1]), **'NOT in key range '**)  
  
**for** i **in** s:  
 print( i) *# displays all the keys check to see in dictionary*

**OUTPUT:**

1 dict\_keys([10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110])

2 dict\_values(['Euclid', 'Archimedes', 'Newton', 'Descartes', 'Fermat', 'Turing', 'Euler', 'Einstein', 'Boole', 'Fibonacci', 'Nash'])

3 names == {10: 'Euclid', 20: 'Archimedes', 30: 'Newton', 40: 'Descartes', 50: 'Fermat', 60: 'Turing', 70: 'Euler', 80: 'Einstein', 90: 'Boole', 100: 'Fibonacci', 110: 'Nash', 120: 'Hopper'}

the length of names is 12

searching...

4 key == 110 value == Nash

5 value Euler found

====ALL KEYS====

10 Euclid

20 Archimedes

30 Newton

40 Descartes

50 Fermat

60 Turing

70 Euler

80 Einstein

90 Boole

100 Fibonacci

110 Nash

120 Hopper

==== ALL VALUES====

Euclid

Archimedes

Newton

Descartes

Fermat

Turing

Euler

Einstein

Boole

Fibonacci

Nash

Hopper

====string manipulation====

Enter several integers separated by a comma 11, 120, 40 USER INPUT

6 s == 11, 120, 40

7 using the split function ['11', ' 120', ' 40'] length == 3

8 int (s[0]) == 11

ERROR key == 11 NOT in key range

10 x == 120

11 key found == 120 value == Hopper

11

120

40

Process finished with exit code 0