Parth Agarwal 18BCAN094
BCA B2
04/06/2020 Programming in Python (BCA134A)

Parth

Date
Page

END SEM EXAM
Section - B

Q4. i) Lists, strings and tuples are ordered sequences of objects. Unlike strings that contain only characters, list and tuples can contain only type of objects. Lists are like arrays and are mutables so they can be extended or reduced at will. Strings are immutables.
Lists are enclosed in brackets:

```
l = [1, 2, 3, 4]
```

Dictionaries are built with curly brackets:

```
d = {"a" : 1, "b" : 2}
```

⊙ Dict :- A dictionary is a sequence of items. Each item is a pair made of a key & a value. Dictionaries are not sorted.

ex:
```
d = {'first' : 'string value', 'second' : [1,2]}
d.keys()
['first', 'second']
d.values()
['string value', [1,2]]
```

Scanned with Cam

18BCAN094 parth.

functions:

- items: returns a list of items of the form (key, value).

ex:-
```
d = {'first': 'string value', 'second': [1, 2]}
d.items()
[('a', 'string value'), ('b', [1, 2])]
```

- has_key: to check for existence of a specific key.

ex:- d.has_key('first')
true.

⊙ strings:- single & double quotes are special characters. string can be stored in '', "" and ''' '''.

slicing:
```
text[0]
text[-1]
text[0:]
```
string are immutable that's why text[0] = 'a' is incorrect.

indexing:

0	1	2	3	4
p	a	r	t	h
-5	-4	-3	-2	-1

Scanned with Cam

18BCAN094 Party

Functions:

- isdigit() - is the string made of digits only?
  ex: "44".isdigit()
  - True
- isalpha() - is the string made of alphabetic characters only?
  ex: "44".isalpha()
  - False
- isupper() - upper case only
  ex: "Aa".isupper()
  - False
- islower() - lower case only
  ex: "aa".islower()
  - True.

⊙ Lists :- Lists are part of standard language.

Functions:

- index - searches for an element in a list.
  ex: my_list = ['a', 'b', 'c', 'b', 'a']

Scanned with Cam

18BCAN094 Fazly

my_list.index('b')
— 1

- insert - removes element but also insert element wherever you want in list.

ex: my_list.insert(2, 'a')
my_list
— ['b', 'c', 'a', 'b']

- pop - remove last element
ex:- my_list.pop()
'a'
my_list
— ['b', 'c', 'b']

iii.) In python, an anonymous function is a function that is defined without a name. while normal functions are defined using the def keyword, anonymous functions are defined using lambda keyword.

Syntax: lambda arguments: expression

Ex: double = lambda x: x * 2
print(double(5))

output = 10

Scanned with Cam

18BCAND94 Parth

Date
Page

Example use with filter():

The filter() function takes in a function and a list as arguments.
The function is called with all the items in the list and a new list is returned which contains items for which the function evaluates to true.

```
ex:- my_list = [1, 5, 4, 6, 8, 11, 13, 12]

my_list2 = list(filter(lambda
              x : (x%2 == 0),
              my_list))

print(my_list2)
```

output = [4, 6, 8, 12]

Example use with map():

The map() function in Python takes in a function and a list.
The function is called with all the items in the list and a new list is returned which contains items returned by that function for each item.

Scanned with CamScanner

18BCAN094 Party

Ex:- my_list = [1, 5, 4, 6, 8, 11, 3, 12]

new_list = list(map(lambda x : x*2, my_list))

print(new_list)

output - [2, 10, 8, 12, 16, 22, 6, 24]

Example use with reduce():

The reduce() function continually applies the function func() to the sequence seq. It returns a single value.

Ex:- reduce(lambda x, y : x+y, [47, 11, 42, 13])

output - 113

ii) l = ["A", "sunny", "day"]
resultant_list = []
for i in l:
    resultant_list.append((i.swapcase() + i.swapcase()))
print(resultant_list)

output = ["aa", "SUNNYSUNNY", "DAYDAY"]

Scanned with Cam

18BCAN094 Partha.

iv.) Reading :

a) You can read a file in Python by calling that file in a "read mode" "(r).

1. open the file in read mode
   f = open ("data.txt", "r")
2. We use mode function in the code to check that the file is in open mode.
   if f.mode == 'r':
3. Use f.read to read file data & store it in variable content
   contents = f.read()
4. print contents.

Writing :

You can write a file in Python by calling .txt file in a "write mode" "(w).

Reading & writing in files:

To do this you can call .txt file in "plus mode" (+).

b) File Handler - Python too supports file handling and allows users to handles files i.e. to read and write files, along

Scanned with Cam

18BCAN094 Rasty

with many other file handling options to operate on files. Each line of code includes a sequence of characters and they form text file.

Tell function - this method returns current position of file object. This method takes no parameters and returns an integer value. Initially file pointer points to beginning of the file. So, the initial value of tell() is zero.

Seek function - this method set the file's current position at the offset. The whence argument is optional and defaults to 0, which means absolute file positioning, other values are 1 which means seek relative to the current position and 2 means seek relative to the file's end.

v.) Default Arguments :- can have default provide values in python. We can provide default value to an argument by using the assignment operator (=).

Ex:

18BCA0094 Partly

```
def greet(name, msg="Good morning!"):
    print("Hello", name + ', ' + msg)

greet("Kate")
greet("Bruce", "How you doing?")
```

In this Function, the parameter name does not have a default value and is required during a call.

Named Arguments :- When we call a function with some values, these values get assigned to the arguments according to their position.

Ex:- greet(name = "Bruce", msg = "How you doing?")

greet(msg = "How you doing?", name = "Bruce")

Section - III

Q3 i) File modes -

r : Open file in read - only mode

18BCA N074 Pavithra.

r+ : Opens a file in reading / writing mode.
w : open file in writing mode
w+ : Opens a file for reading & writing
a : opens file for appending new information.
a+ : opens for both appending and reading

ii) Mutable objects can change their state or contents and immutable objects can't change their state or content.

2. Immutable objects - These are of in-built types like int, float, bool, string, unicode.
ex:- tuple1 = (0, 1, 2, 3)
tuple1[0] = 4
print (tuple1)

Output - Error

Mutable objects - These are of type list, dict, set.
Custom classes are generally mutable.

ex:- color = ["red", "blue", "green"]

Scanned with CamScanner

18BCAN094 Parth.

Date
Page

```
print (color)

color [0] = "pink"
color [-1] = "orange"
print (color)
```

Output = ['red', 'blue', 'green']
['pink', 'blue', 'orange']

3. Use of mutable objects is recommended when there is a need to change the size or content of object.

iii.)

```
def triangle (n):
    for i in range (n):
        for j in range (n-i):
            print (' ', end = '')
        for k in range (2*i+1):
            print ('*', end = '')
        print ()
def tree (n):
    for i in range (n):
        for j in range (n-1):
            print (' ', end = ' ')
        print ('***')
n = int (input ('enter no. of rows : '))
triangle (n)
triangle (n)
```

Scanned with CamScanner

1BTCAN074 Rajath.

tree(n)

IV.) Python define type conversion functions to directly convert one data type to another which is useful in day to day and competitive programming.

1. int(a, base) : converts any data type to integer.
2. float() : convert any data type to float.

Ex:-

```
s = "10010"

c = int(s, 2)
print("After converting to integer base 2 : ", end = " ")
print(c)                    # output = 18.

e = float(s)
print("After converting to float : ", end = " ")
print(e)                    # output = 10010.0
```

3. ord() : converts character to integer.
4. hex() : converts integer to hexadecimal string.
5. oct() : converts integer to octal string.

Scanned with CamScanner

18BCAN094 Party

Ex:-

```
s = '4'

c = ord (s)
print (c)

c = hex (56)
print (c)

c = oct (56)
print (c)
```

Output = 52
0x38
0o70

6. tuple()
7. set()
8. list()
9. dict()
10. str().
11. complex(real, image)

V.) Python documentation string provide a convenient way of associating documentation with Python modules, functions, classes and methods. It's specified in source code that is used, like a comment to document a specific segment

Scanned with Cam

18BCA1N0704 Parth.

Q. Code.

The docstrings are declared using """ triple double quotes """ just below the class, method or function declaration. All functions should have docstring.

Ex:-

```
def my_function():
    """ Demonstrates docstrings and
        does nothing really.
    """
    return None

print "using __doc__:"
print my_function.__doc__

print "using help:"
help(my_function)
```

Section- II

Q2.i) 1. Lists are mutable that is can be extended or changed in contents whereas Tuples are immutable contents can't be changed.

2. Ex:- List = ['Hello', 'Bye!']
   List[0] = 'Hi'

Scanned with CamScanner

18BCAN094 Part B

Date
Page

List
→ output = ['Hi', 'BYE!']

tuple1 = ('Hi', 'BYE')
tuple1[0] = 'Hel'
tuple1
→ output = Error

ii.) Unpacking Tuples:

ex:- >>>(a, b, c) = (1, 2, 3)
>>> a
1
>>> b
2
>>> c
3

iii.) a.) False

b.) 17.4

iv.) input_string = input("Enter list numbers or elements seperated by space: ")

userList = input_string.split()
print("User List is", userList)

Scanned with CamScanner

1 BSCAN0 94 Reddy.

Date
Page

vi) x = " awesome "

```
def myfunc():
    print("Python is:" + x)

myfunc()
```

section - II

Q1 i.) c.) 2.0
ii) b.) x**y
iii.) b.) 1
iv.) a.) left to right manner
v.) c.) 3
vi.) b.) cd
vii.) b.) 'n'
viii.) a.) 2
ix.) c.) 79 characters
x.) d.) None

Scanned with Cam