Dewesh Chopra

MCA/10032/22

ML Lab

Assignment no. 1

Perform following operations using Python:

a) Updating an existing String

b) Using built-in String Methods to Manipulate Strings e.g len(),find(),decode(),isalpha(), isdigit() ,count() etc.

```
# Question 1

str = "Hello, world."

# update
str = "Hello World!"

# methods
len(str)
```

```
12
```

```
str.find("e")
```

```
1
```

```
str.isalpha()
```

```
False
```

```
str.isdigit()
```

```
False
```

```
str.count("l")
```

```
3
```

```
strenc = str.encode('utf-8')
strenc.decode(encoding='utf-8',errors='strict')
```

```
'Hello World!'
```

Perform the following on ‘Lists’ using Python

a) Updating, Slicing, Indexing

b) Using following built-in Methods to manipulate lists: append(), extend(), insert(), pop(), remove(),reverse(), sort().

```
# Question 2

lst = [10, 20, 30, 40, 50]

# update
lst = ["Cat", "Dog", "Bear", "Snake", "Horse", "Tiger"]

# slicing
lst2 = lst[1:4]

lst2
```

```
['Dog', 'Bear', 'Snake']
```

```
# indexing
lst[3]
```

```
'Snake'
```

```
# append
lst.append("Wolf")
lst
```

```
['Cat', 'Dog', 'Bear', 'Snake', 'Horse', 'Tiger', 'Wolf']
```

```
# extend
lst3 = ["Eagle", "Koala"]
lst.extend(lst3)
lst
```

```
['Cat', 'Dog', 'Bear', 'Snake', 'Horse', 'Tiger', 'Wolf', 'Eagle',
'Koala']
```

```
# insert
lst.insert(2, "Panda")
lst
```

```
['Cat',
 'Dog',
 'Panda',
 'Bear',
 'Snake',
 'Horse',
 'Tiger',
 'Wolf',
 'Eagle',
 'Koala']
```

```
# pop and remove
lst.pop()
lst
```

```
['Cat', 'Dog', 'Panda', 'Bear', 'Snake', 'Horse', 'Tiger', 'Wolf',
'Eagle']
```

```
lst.remove("Panda")
lst
```

```
['Cat', 'Dog', 'Bear', 'Snake', 'Horse', 'Tiger', 'Wolf', 'Eagle']
```

```
# reordering
lst.reverse()
```

```
lst
```

```
['Eagle', 'Wolf', 'Tiger', 'Horse', 'Snake', 'Bear', 'Dog', 'Cat']
```

```
lst.sort()
lst
```

```
['Bear', 'Cat', 'Dog', 'Eagle', 'Horse', 'Snake', 'Tiger', 'Wolf']
```

Perform the following on ‘Tuples’ using Python

a) Updating, Indexing, deleting, slicing.

b) Using following built-in Methods to manipulate Tuples: max(), min(), len(), tuple().

```
# Question 3

# update
tup = (2, 4, 6, 8, 10)
tup
```

```
(2, 4, 6, 8, 10)
```

```
tup = ("Eagle", "Crow", "Owl", "Nightingale")
tup
```

```
('Eagle', 'Crow', 'Owl', 'Nightingale')
```

```
# indexing, deleting and slicing
tup[2]
```

```
'Owl'
```

```
tup2 = tup[1:3]
tup2
```

```
('Crow', 'Owl')
```

```
# built-in methods
max(tup)
```

```
'Owl'
```

```
min(tup)
```

```
'Crow'
```

```
len(tup)
```

```
4
```

```
x = [1, 2, 3]
x
```

```
[1, 2, 3]
```

```
x = tuple(x)
```

```
type(x)
```

```
tuple
```

Perform the following on ‘Dictionary’ using Python

a) Updating, deleting

b) Using following built-in Methods to manipulate Dictionary: update(), values(),get(), clear(), copy(), type(), len().

```
# Question 4
```

```
dic = {"india": "new delhi",
       "pakistan": "islamabad",
       "nepal": "new katmandu",
       "bangladesh": "dhaka",
       "sri lanka": "Sri jayawardhanapura"}

dic
```

```
{'india': 'new delhi',
 'pakistan': 'islamabad',
 'nepal': 'new katmandu',
 'bangladesh': 'dhaka',
 'sri lanka': 'Sri jayawardhanapura'}
```

```
# update
dic.update({"nepal": "katmandu"})
dic
```

```
{'india': 'new delhi',
 'pakistan': 'islamabad',
 'nepal': 'katmandu',
 'bangladesh': 'dhaka',
 'sri lanka': 'Sri jayawardhanapura'}
```

```
# deleting
del dic["pakistan"]
dic
```

```
{'india': 'new delhi',
 'nepal': 'katmandu',
 'bangladesh': 'dhaka',
 'sri lanka': 'Sri jayawardhanapura'}
```

```
# built-in methods
dic.update({"sri lanka": "sri jayawardhanapura kotte"})
dic
```

```
{'india': 'new delhi',
 'nepal': 'katmandu',
 'bangladesh': 'dhaka',
 'sri lanka': 'sri jayawardhanapura kotte'}
```

```
dic.values()
```

```
dict_values(['new delhi', 'katmandu', 'dhaka', 'sri jayawardhanapura
kotte'])
```

```
dic.keys()
```

```
dict_keys(['india', 'nepal', 'bangladesh', 'sri lanka'])
```

```
dic.get("india")
```

```
'new delhi'

# clearing
dic2 = {"x": "y"}
dic2

{'x': 'y'}

dic2.clear()

dic2

{}

dic2 = dic.copy()
dic2

{'india': 'new delhi',
 'nepal': 'katmandu',
 'bangladesh': 'dhaka',
 'sri lanka': 'sri jayawardhanapura kotte'}

type(dic)

dict

len(dic)

4
```

Create Separate Sets for Indian Cricket Players playing in T20, ODI and Test Match for current West Indies tour. Also perform the union(), intersection(), difference() operations on the above sets.

```
# Question 5

# Create Separate Sets for Indian Cricket Players playing in T20, ODI
and Test Match for current West Indies tour.
# Also perform the union(), intersection(), difference() operations on
the above sets.

t20 = ("Sharma", "Kohli", "Rahane", "Dhawan", "Chahal", "Bumrah",
"Ashwin", "Pandya", "Jadeja", "Rahul")
odi = ("Sharma", "Kohli", "Rahane", "Dhawan", "Pant", "Shami",
```

```
"Ashwin", "Pandya", "Jadeja", "Iyer")
test = ("Sharma", "Kohli", "Nehra", "Dhawan", "Dhoni", "Bumrah",
"Ashwin", "Pandya", "Jadeja", "Rahul")

t20 = set(t20)
odi = set(odi)
test = set(test)

set_union = t20 | odi
set_union
```

```
{'Ashwin',
 'Bumrah',
 'Chahal',
 'Dhawan',
 'Iyer',
 'Jadeja',
 'Kohli',
 'Pandya',
 'Pant',
 'Rahane',
 'Rahul',
 'Shami',
 'Sharma'}
```

```
set_int = t20 & odi
set_int
```

```
{'Ashwin', 'Dhawan', 'Jadeja', 'Kohli', 'Pandya', 'Rahane', 'Sharma'}
```

```
set_diff = t20 - odi
set_diff
```

```
{'Bumrah', 'Chahal', 'Rahul'}
```

```
set_symm_diff = t20 ^ odi
set_symm_diff
```

```
{'Bumrah', 'Chahal', 'Iyer', 'Pant', 'Rahul', 'Shami'}
```

Find the Largest of the 4 Strings using Conditional Statements(if-elif-else) using Python.

```
# Question 6
```

```
s1 = "Long"
s2 = "Large"
s3 = "Larger"
s4 = "Largest"

if len(s1) >= len(s2) and len(s1) >= len(s3) and len(s1) >= len(s3):
    print(s1, " is largest")
elif len(s2) >= len(s1) and len(s2) >= len(s3) and len(s2) >= len(s4):
    print(s2, " is largest")
elif len(s3) >= len(s1) and len(s3) >= len(s2) and len(s3) >= len(s4):
    print(s3, " is largest")
else:
    print(s4, "is largest")
```

```
Largest is largest
```

Write a function to generate even numbers between 1 to 30. Create a list squaring these numbers and display the list as well. Create another list by filtering the squared list using ‘anonymous’ function to get those numbers which are even numbers(Hint: Use filter() method and ‘lambda’ keyword)

```
# Question 7

def get_even_nums():
    even_nums = [x for x in range(1, 31) if x % 2 == 0]
    print(even_nums)
    even_nums_sq = [x ** 2 for x in even_nums]
    print(even_nums_sq)
    # filter function with lambda
    filtered_lst = list(filter(lambda x: x % 2 == 0, even_nums_sq))
    print(filtered_lst)

get_even_nums()
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
[4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576, 676, 784, 900]
[4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576, 676, 784, 900]
```