

Lab 1 – Report

Meet Kansara – 220929270 Roll no. 54

Aim: To familiarize with fundamental Python programming concepts

Code Execution and analysis:

1. Input Handling:

```
a = input("Enter number 1: ")  
  
b = input("Enter number 2: ")  
  
print("Number a is: ", a)  
print("Number b is: ", b)  
  
print("sum of a+b =", a+b)
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/1_input.py  
Enter number 1: 5  
Enter number 2: 7  
Number a is: 5  
Number b is: 7  
sum of a+b = 57
```

2. Strings:

```
name = "Kashyap"  
  
print(name[0:3])  
  
print(name[-4: -1])  
print(name[1:4])  
  
print(name[:4]) # is same as print(name[0:4])  
print(name[1:]) # is same as print(name[1:5])  
print(name[1:5])
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/2_negative_slicing.py  
Kas  
hya  
ash  
Kash  
ashyap  
ashy
```

3. Lists:

```

friends = ["Apple", "Orange", 5, 345.06, False, "Aakash", "Rohan"]
print(friends)
friends.append("Harry")
print(friends)

l1 = [1, 34, 62, 2, 6, 11]
# l1.sort()
# l1.reverse()
# l1.insert(2, 333333) # Insert 333333 such that its index in the list is 3
value = l1.pop(3)
print(value)
print(l1)

```

```

meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/3_list_methods.py
['Apple', 'Orange', 5, 345.06, False, 'Aakash', 'Rohan']
['Apple', 'Orange', 5, 345.06, False, 'Aakash', 'Rohan', 'Harry']
2
[1, 34, 62, 6, 11]

```

4. Tuples:

```

a = (1, 45, 342, 3424, False, 45, "Rohan", "Shivam")
print(a)

no = a.count(45)
print(no)

i = a.index(3424)
print(i)

print(len(a))

```

```

meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/4_tuple_methods.py
(1, 45, 342, 3424, False, 45, 'Rohan', 'Shivam')
2
3
8

```

5. Dictionary:

```

marks = {
    "Kashyap" : 100,
    "Vivek" : 85,
    "Sumit" : 65
}

print(marks, type(marks))

print(marks["Kashyap"])

```

```

meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/5_dic.py
{'Kashyap': 100, 'Vivek': 85, 'Sumit': 65} <class 'dict'>
100

```

6. Sets:

```
s1 = {1, 45, 6, 78}
s2 = {7, 8, 1, 78}

print(s1.union(s2))
print(s1.intersection(s2))
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/6_set_union.py
{1, 6, 7, 8, 45, 78}
{1, 78}
```

7. Conditional statements:

```
a = int(input("Enter your age: "))
# If elif else ladder
if(a>=18):
    print("You are above the age of consent")
    print("Good for you")

elif(a<0):
    print("You are entering an invalid negative age")

elif(a==0):
    print("You are entering 0 which is not a valid age")

else:
    print("You are below the age of consent")

print("End of Program")
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/7_if_elif_else_ladder.py
Enter your age: 21
You are above the age of consent
Good for you
End of Program
```

8. Break and continue statements:

```
for i in range(5):
    if(i == 3):
        break # Exit the loop right now
    print(i)

for i in range(5):
    if(i == 3):
        continue # Skip this iteration
    print(i)
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/8_break_continue.py
0
1
2
0
1
2
4
```

9. Function definitions:

```
def my_func(num):  
    return num*2  
  
seq=[2,3,4,5,6,7]  
  
map(my_func,seq) # predefined function  
  
a= list(map(my_func,seq))  
  
print(a)
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/9_Def_function.py  
[4, 6, 8, 10, 12, 14]
```

10. Classes and objects:

```
import math  
  
class Number:  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
    def distance(self, p):  
        dis = (math.sqrt((self.x - p.x)**2 + (self.y-p.y)**2))  
        return(dis)  
  
point1 = Number (2, 3)  
point2 = Number (4,5)  
# point3 = Number (6, 7)  
  
distance1=point1.distance(point2)  
# distance2=point1.distance(point3)  
  
print(distance1)  
# print(distance2)
```

```
meet@meet:~/Desktop/Lab/codes$ /bin/python3 /home/meet/Desktop/Lab/codes/10_Classes.py  
2.8284271247461903
```

Conclusion: A comprehensive introduction to essential Python programming concepts crucial for ROS applications was successfully achieved.