	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESMENT YEAR: 2021-2022	CLASS: SE-5
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2021-22/	Starting date: 6/12/2021
	Roll No:22108	Submission date:11/12/2021
Title:	**String Operations on Python**	
Problem statement	Implement String operations on python	
Prerequisites:	Basics of Python Programming	
	Decision making and loop controls	
	Choice based program	
	String	
Objectives:	Learn to create and display a String	
	Implement various operation on string to understand its effect on data.	
Theory:		

<u>Python</u> –

- Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.
- It is used for:
  - web development (server-side),
  - software development, -   mathematics,
  - system scripting.
- Python has simpler syntax than C or CPP but it is still in development and doesn’t have some functions that C or CPP has.
- Usually, files containing python code is given domain as ‘.py’.

<u>Strings</u> –

- Strings in python are surrounded by either single quotation marks, or double quotation marks.
- Strings are a built-in data structure in python and they are immutable.
- You can handle textual data in Python using the *str* object.
- Single (‘’), double (“”) and triple (”””” ””””) quotes can be used to wrap around a string.
- To declare a string variable –

Empty String – string = “”

User Input – string =

str(input(“Enter a String: “)) or

string1 = “Hello World”

<u>Functions</u> –

- In Python, a function is a group of related statements that performs a specific task.
- Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable.
- Furthermore, it avoids repetition and makes the code reusable.
- Syntax – def function(): ←This line is for defining function with name and arguments if required

  print(“Hello”) ← main code in the function, which will be executed when called.

  function() ←Function call, here we can add arguments in the parenthesis if mentioned in the definition of function above.

- Return can also be used if returning a value/data and in the main program we need to create a variable, and assign it the function call with argument if required. Example –

  def add(a, b):
  return a+b

  c = add(10, 10) ← Here we can have a variable (with preassigned value) or direct value as argument. print(c)
  Output –
  20

ERROR and REMEDY	**-**

Code	import sys
	
	def end():
	    print("\n\tThis is the end of Execution!")

```
    sys.exit()


def concatenation(string,
string1):     string2 =
string+string1     return string2


def
stringlength(string):
count = 0     for letter
in string:         count +=
1
    return count


def substring(string, position, size):
    count = 0
    length  =  stringlength(string)         newstring  =  ""          if  position  >  length:
print("\nThe position of substring entered by user is greater than the size of the
string, which is not possible as position should be less than the size of string!")
end()     elif size > (length-position):         print(
            f"\nThis is not possible as size of substring exceeds the no. of
remaining letters in the string after {position} position in string is {length-
position} which is less than {size}. Try again with proper size or position of
substring!!")         end()

    for letter in string:         count += 1
count_sub = 0         if (count < position and
count_sub < size):             count_sub += 1
            newstring += letter

    print(
        f"\nThe substring in the main string starting from {position} position and
of size {size} is : {newstring}.")     end()


def palindrome(string):
    stringlen = stringlength(string)
for i in range(0, stringlen/2):         if
string[i] != string[stringlen-i-1]:
return False
```

```
        return True


def reverse(string):     stringlen =
stringlength(string)     revstring
= ""
    str_l = list(string)
     i = 0     for i in range(0,
stringlen):         revstring +=
str_l[stringlen-i-1]

    print("The Reversed string is: "+revstring)



print("\n\t\tString Opertions") string
= str(input("Enter a string: "))
print("The string entered: ", string)

print("Here we have 4 string Oerations that can be applied to string entered
above.\n 1) Concatenation\n2) Calculating Length of string\n3) Substring\n4)
Membership Test (Searching if input phrase is in the string)\n5) Enumerate the
string\n6) Check for Palindrome\n7) Reverse a string\n8) Exit.\n") choice =
int(input("Enter your Choice: "))

if (choice == 1):  # Concatenate strings
string1 = str(input(
        "Enter the second string for concatinating to the first already entered
before:
"))
    string2 = concatenation(string, string1)
print(f"the concatenated string: '{string2}'")     end()

elif (choice == 2):  # Calculate Length of string
count = stringlength(string)
    print(f"The Length of string is {count}")
end()

elif (choice == 3):  # Substrings
position = int(
        input("Enter the position of start of substring in the main string: "))
size = int(input("Enter the size of substring: "))

    substring(string, position, size)

elif (choice == 4):  # Membership Test
print("\n")     string2 = str(
```

	input("Enter a phrase or letters to check if they are in the main string: "))

```
    if string2 in string:
        print("'"+string2+"'" was found in main string.")
else:
        print("'"+string2+"'" was not found in main string.")
end()

elif (choice == 5):  # Enumerate
print("\n")
    string_list = enumerate(string)
    print(list(string_list))
end()

elif (choice == 6):  #
Palindrome     print("\n")
answer = palindrome(string)
if (answer):
        print("The string has Palindrome!")
else:
        print("The string does not have Palindrome!")
end()

elif (choice == 7):  # Reverse
    reverse(string)
end()

elif (choice == 8):  # Exit
    end()

else:
    print("Invalid Entry, Enter choice number between 1 to 8 respective to the
operation list above!")
```

CONCLUSION:

In this experiment, we implemented a python code for string operations, where we applied concepts of function, string, iteration through string and typecasting. Here, operations are mainly Substring, Palindrome, Reverse of String, Membership check, Enumerate the string and Concatenation.

REFERENCES:

Seymour Lipschutz, Data Structure with C, Schaum’s Outlines, Tata McGrawHill

E Balgurusamy - Programming in ANSI C, Tata McGraw-Hill (Third Edition)

Yashavant Kanetkar- Let Us C, BPB Publication, 8th Edition.

Continuous Assessment for DS AY 2021-22			
RPP (5)	**SPO (5)**	**Total (10)**	**Signature:**
			Assessed By: Mr. V. B. Vaijapurkar
Start date	**Submission date**		**Date:**
6/12/2021	**11/12/2021**		**Roll. No.22108**
***Regularity, Punctuality, performance** ***Submission, Presentation, orals**			