LAB # 14

STRINGS

OBJECTIVE

Working on the strings formatting.

THEORY

A string is a sequence of characters which are surrounded by either single quotation marks, or double quotation marks. A str object is immutable; that is, its content cannot be changed once the string is created.

Creating Strings:

Create strings by using the constructor, as follows:

```
s1 = str()                  # Create an empty string object
s2 = str("Welcome")         # Create a string object for Welcome
```

Create strings by simple syntax of a string value

```
s1 = " "                    # Same as s1 = str()
s2 = "Welcome"              # Same as s2 = str("Welcome")
print("It's me!")           # quotes inside a string.
a = """Hello Class, Welcome to the Lab of Strings.
        We have already learnt about integers in last lab but today
         We will learn about strings! """
```

Example:

```
s = input("Enter a string: ")
if len(s) % 2 == 0:
   print(s, "contains an even number of characters")
else:
   print(s, "contains an odd number of characters")
```

Output:

```
>>> %Run task1.py
Enter a string: Strings
Strings contains an odd number of characters
>>>
```

LAB # 14

Access Characters & String Slicing:

A character in the string can be accessed through the index operator using the syntax: **s[index],** the indexes are 0 based, that range from 0 to len(s)-1.

Example:

```
s = "Welcome"
print(s[-2])
print(s[0 : 4])
print(s[4 : ])
```

Output:

```
>>> %Run task2.py
m
Welc
Ome
```

String Iteration

Strings are iterable; you can loop through characters one by one.

Example:

```
s = "Python"
for char in s:
    print(char)
```

Output:

```
>>> %Run Lab_14.py
P
y
t
h
o
n
```

String Immutability: Deleting & Updating a String

- ✓ Strings are **immutable,** which means that they cannot be changed after they are created.
- ✓ If we need to manipulate strings then we can use methods like **concatenation, slicing** or **formatting** to create new strings based on original.

LAB # 14

- ✓ In Python, it is not possible to delete individual characters from a string since strings are immutable. However, we can delete an entire string variable using the **del** keyword.
- ✓ As strings are immutable, "updates" create new strings using slicing or methods such as **replace()**.

Example:

```
# Immutable

s = "helloHelpers"
s = "H" + s[1:]   # create new string
print(s)

# Deleting

s = "Lab"
del s
print(s)

# Updating a String

s = "hello helpers"
s1 = "H" + s[1:]                          # update first character
s2 = s.replace("helpers", "Students")     # replace word
print(s1)
print(s2)
```

Output:

```
>>> %Run Lab_14.py

HelloHelpers

NameError

Hello helpers
hello Students
```

Concatenating and Repeating Strings

We can concatenate strings using + operator and repeat them using * operator.

Example:

```
s1 = "Hey"
s2 = "Everyone"
```

```
print(s1 + " " + s2)
```

Output:

```
>>> %Run task2.py
Hey Everyone
```

Comparing Strings:

Compare the strings by using the comparison operators (==, !=, > , >= , < , and <=). Python compares strings by comparing their corresponding characters, and it does this by evaluating the characters' numeric codes. The ASCII Character Set, to find the numeric codes for characters.

Example:

```
s1 = input("Enter the first string: ")
s2 = input("Enter the second string: ")
if s2 < s1:
   s1, s2 = s2, s1
print("The two strings are in this order:", s1, s2)
```

Output:

```
>>> %Run task3.py
Enter the first string: aza
Enter the second string: abz
The two strings are in this order: abz aza
```

Converting Strings

The ***str*** class contains these methods for converting letter cases in strings and for replacing one string with another.

capitalize(): str	Returns a copy of this string with only the first character capitalized
lower(): str	Returns a copy of this string with all letters converted to lowercase.
upper(): str	Returns a copy of this string with all letters converted to uppercase.
title(): str	Returns a copy of this string with the first letter capitalized in each word.
swapcase(): str	Returns a copy of this string in which lowercase letters are converted to uppercase and uppercase to lowercase.
replace(old, new): str	Returns a new string that replaces all the occurrences of the old string with a new string.

Stripping Whitespace Characters from a String:

LAB # 14

Use the following methods to strip whitespace characters from the front, end, or both the front and end of a string. Recall that the characters ' ', \t, \f, \r, and \n are called the whitespace characters. Following methods are : **lstrip(), rstrip(), strip().**

EXERCISE

A. Point out the errors, if any, in the following Python programs.

1. Code:

```
a = "PYTHON"
a[0] = "x"
#Apply Exception for mention error
```

Output:

2. Code:

```
a = STRING
i = 0
while i < len(b):
    c = a[i]
    print(c)
    i+=i + 1
```

Output:

3. Code:

```
Def 1my_function(x):
return x[::-1]

mytxt = 1my_function("I wonder how this text looks like backwards")
print(mytxt)
```

Output:

LAB # 14

B. What would be the output of the following programs:

1. Code:

```
s= "Welcome"
for i in range(0, len(s), 2):
    print(s[i], end = '')
```

Output:

2. Code:

```
s = input("Enter a string: ")
if "Python" in s:
    print("Python", "is in", s)
else:
    print("Python", "is not in", s)
```

Output:

3. Code:

```
str='cold'
list_enumerate=list(enumerate(str))
print("list enumerate:", list_enumerate)
print("list length:", len(str))
s1 = "Welcome to Python"
s2 = s1.replace("o","abc")
print(s2)
a = "Python" + "String"
b = "<" + a*3 + ">"
print(b)
```

Output:

LAB # 14

C. Write Python programs for the following:

1. Write a program that Store a person's name, and include some whitespace characters at the beginning and end of the name. Make sure you use each character combination, "\t" and "\n", at least once. Print the name once, so the whitespace around the name is displayed. Then print the name using each of the three stripping functions, lstrip(),rstrip(), and strip().

2. Write a program that asks the user for their favourite color. Create the following output (assuming blue is the chosen color) (hint: use '+' and '*')

```
blueblueblueblueblueblueblueblueblueblue
blue                                blue
blueblueblueblueblueblueblueblueblueblue
```