Batch: C5_1	**Roll No.: 19**
Experiment / assignment / tutorial No: 1	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Write a program to demonstrate use of basic data types in Python.

AIM: 1) Program to calculate salary of an employee
2) Program to perform string operations.

Expected outcome of Experiment:

CO1: Formulate problem statement and develop the logic (algorithm/flowchart) for its solution.
CO2: Understand the concepts of data structures in python.

Use of input output function, arithmetic operators in python and different operations on string.

Resource Needed: Python IDE

Books/ Journals/ Websites referred:

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,India
3. https://www.geeksforgeeks.org/python-strings/?ref=lbp

Theory:

How the input function works in Python:

- When input() function executes program flow will be stopped until the user has given an input.
- The text or message displayed on the output screen to ask a user to enter input value is optional i.e. the prompt, will be printed on the screen is optional.

- Whatever you enter as input, the input function converts it into a string. If you enter an integer value still input() function convert it into a string. You need to explicitly convert it into an integer in your code using typecasting.

  Example:
  Name=input("Enter your name")
  print('Hello, ' + Name)

  Output:-
  Enter your name Mahesh
  Hello, Mahesh

Python Arithmetic Operators:
Assume variable **a** holds 10 and variable **b** holds 20, then

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	a + b = 30
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -10
* Multiplication	Multiplies values on either side of the operator	a * b = 200
/ Division	Divides left hand operand by right hand operand	b / a = 2
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 0
** Exponent	Performs exponential (power) calculation on operators	a**b =10 to the power 20

//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) −	9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0

Strings:
We can create string simply by enclosing characters in quotes. Python treats single quotes the same as double quotes. Creating strings is as simple as assigning a value to a variable.

Example:-
var1= “Hello World”
var2=”Python Programming”

String Special Operators:

Assume string variable **a** holds 'Hello' and variable **b** holds 'Python', then

Operator	Description	Example
+	Concatenation - Adds values on either side of the operator	a + b will give HelloPython
*	Repetition - Creates new strings, concatenating multiple copies of the same string	a*2 will give - HelloHello
[]	Slice - Gives the character from the given index	a[1] will give e

[:]	Range Slice - Gives the characters from the given range	a[1:4] will give ell
in	Membership - Returns true if a character exists in the given string	H in a will give 1
not in	Membership - Returns true if a character does not exist in the given string	M not in a will give 1

String Methods:

Function Name	Description
capitalize()	Converts the first character of the string to a capital (uppercase) letter
casefold()	Implements caseless string matching
center()	Pad the string with the specified character.
count()	Returns the number of occurrences of a substring in the string.
encode()	Encodes strings with the specified encoded scheme
endswith()	Returns “True” if a string ends with the given suffix

Function Name	Description
expandtabs()	Specifies the amount of space to be substituted with the "\t" symbol in the string
find()	Returns the lowest index of the substring if it is found
format()	Formats the string for printing it to console
format_map()	Formats specified values in a string using a dictionary
index()	Returns the position of the first occurrence of a substring in a string
isalnum()	Checks whether all the characters in a given string is alphanumeric or not
isalpha()	Returns "True" if all characters in the string are alphabets
isdecimal()	Returns true if all characters in a string are decimal
isdigit()	Returns "True" if all characters in the string are digits
isidentifier()	Check whether a string is a valid identifier or not
islower()	Checks if all characters in the string are lowercase

Function Name	Description
isnumeric()	Returns “True” if all characters in the string are numeric characters
isprintable()	Returns “True” if all characters in the string are printable or the string is empty
isspace()	Returns “True” if all characters in the string are whitespace characters
istitle()	Returns “True” if the string is a title cased string
isupper()	Checks if all characters in the string are uppercase
join()	Returns a concatenated String
ljust()	Left aligns the string according to the width specified
lower()	Converts all uppercase characters in a string into lowercase
lstrip()	Returns the string with leading characters removed
maketrans()	Returns a translation table
partition()	Splits the string at the first occurrence of the separator

Function Name	Description
replace()	Replaces all occurrences of a substring with another substring
rfind()	Returns the highest index of the substring
rindex()	Returns the highest index of the substring inside the string
rjust()	Right aligns the string according to the width specified
rpartition()	Split the given string into three parts
rsplit()	Split the string from the right by the specified separator
rstrip()	Removes trailing characters
splitlines()	Split the lines at line boundaries
startswith()	Returns “True” if a string starts with the given prefix
strip()	Returns the string with both leading and trailing characters
swapcase()	Converts all uppercase characters to lowercase and vice versa
title()	Convert string to title case

Function Name	Description
translate()	Modify string according to given translation mappings
upper()	Converts all lowercase characters in a string into uppercase
zfill()	Returns a copy of the string with '0' characters padded to the left side of the string

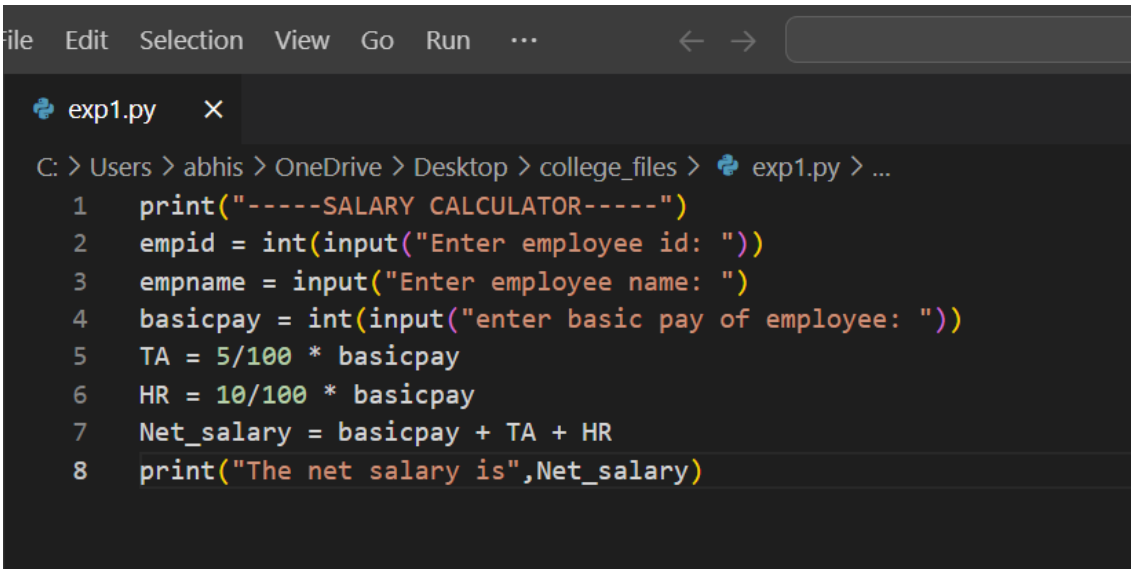
Problem Definition:

1) Write a python program to calculate salary of an employee given his basic pay (to be entered by user), HRA = 10 percent of basic pay, TA = 5 percent of basic pay. Define HRA and TA as constants and use them to calculate the salary of the employee.

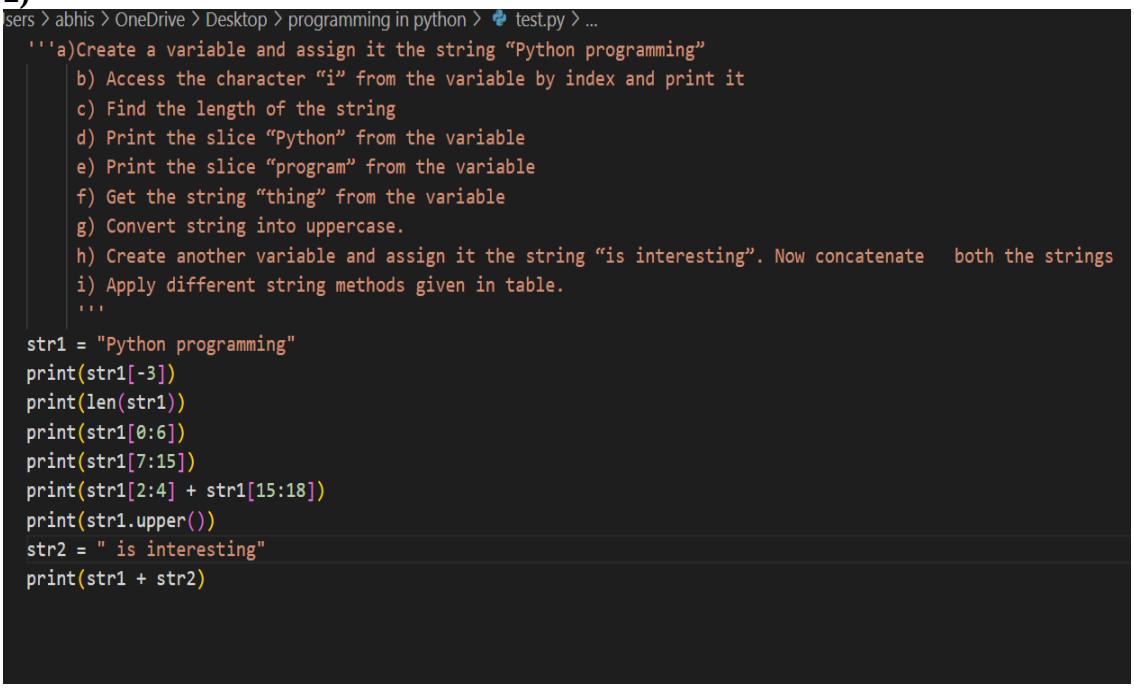
2) a) Create a variable and assign it the string "Python programming"
b) Access the character "i" from the variable by index and print it
c) Find the length of the string
d) Print the slice "Python" from the variable
e) Print the slice "program" from the variable
f) Get the string "thing" from the variable
g) Convert string into uppercase.
h) Create another variable and assign it the string "is interesting". Now concatenate both the strings
i) Apply different string methods given in table.

Implementation details:

1)


File Edit Selection View Go Run ···
exp1.py
C: > Users > abhis > OneDrive > Desktop > college_files > exp1.py > ...
print("-----SALARY CALCULATOR-----")
empid = int(input("Enter employee id: "))
empname = input("Enter employee name: ")
basicpay = int(input("enter basic pay of employee: "))
TA = 5/100 * basicpay
HR = 10/100 * basicpay
Net_salary = basicpay + TA + HR
print("The net salary is",Net_salary)

2)


sers > abhis > OneDrive > Desktop > programming in python > test.py > ...
'''a)Create a variable and assign it the string "Python programming"
b) Access the character "i" from the variable by index and print it
c) Find the length of the string
d) Print the slice "Python" from the variable
e) Print the slice "program" from the variable
f) Get the string "thing" from the variable
g) Convert string into uppercase.
h) Create another variable and assign it the string "is interesting". Now concatenate both the strings
i) Apply different string methods given in table.
'''
str1 = "Python programming"
print(str1[-3])
print(len(str1))
print(str1[0:6])
print(str1[7:15])
print(str1[2:4] + str1[15:18])
print(str1.upper())
str2 = " is interesting"
print(str1 + str2)

```
1   x = "today is a beautiful day"
2   print(x.capitalize())
3   print(x.isprintable())
4   print(x.lower())
5   print(x.upper())
6   print(x.startswith("beautiful"))
7   print(x.title())
8   print(x.rindex("u"))
9   print(x.rpartition("beautiful"))
10  print(x.strip("today"))
11  print(x.splitlines())
12  print(x.swapcase())
13  print(x.replace("is","was"))
14  print(x.join("##"))
15
16
```

Output(s):

1)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

hon3.10.exe c:/Users/abhis/OneDrive/Desktop/college_files/exp1.py
-----SALARY CALCULATOR-----
Enter employee id: 123
Enter employee name: sai
enter basic pay of employee: 1000000
The net salary is 1150000.0
PS C:\Users\abhis>
```

2)

```
i
18
Python
programm
thing
PYTHON PROGRAMMING
Python programming is interesting
PS C:\Users\abhis>
```

```
Today is a beautiful day
True
today is a beautiful day
TODAY IS A BEAUTIFUL DAY
False
Today Is A Beautiful Day
18
('today is a ', 'beautiful', ' day')
 is a beautiful
['today is a beautiful day']
TODAY IS A BEAUTIFUL DAY
today was a beautiful day
#today is a beautiful day#
PS C:\Users\abhis>
```

Conclusion:

I was able to formulate problem statement and develop the logic for its solution. I learned different built-in functions for strings and special string operator which will be useful.

Post Lab Descriptive Questions:-

1. **Which data type will you use to represent the following data values?**
   a. **Number of days in a year**
      Ans. int
   b. **The circumference of a circle**
      Ans. int
   c. **Distance between moon and earth**
      Ans. int
   d. **Whether you will go for a trip?**
      Ans. boolean
   e. **Name of your favourite celebrity**
      Ans. String