Name: Chandana Ramesh Galgali
Batch: P6-1 Roll No.: 16010422234
Experiment / ~~assignment~~ /~~tutorial~~ No. 01
Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Basic concepts in python

AIM: 1) Program to find the distance between two points
2) Program to perform string operations

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

Resource Needed: Python IDE

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 that 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 strings 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

Function Name	Description
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
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

Function Name	Description
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
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

Function Name	Description
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
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

Function Name	Description
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
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) Create four variables x1, y1, x2 and y2. Assign each of them a value from user input using the input() function. Calculate distance [result] between two points having coordinates as (x1,y1) and (x2,y2) by using operators in python and basic built in math functions. Finally, use print() to display "The distance between the two points is [result]" in the output.

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 the table

Implementation details:

(1)

```
#Assuming (x1,y1) and (x2,y2) to be the coordinates lying in the
same plane.
#Taking the coordinates of the first point as input from the user
print("Enter the coordinates of the first point(x1,y1)")
x1=int(input("Enter x1= "))
y1=int(input("Enter y1= "))
print("Enter the coordinates of the second point(x2,y2)")
x2=int(input("Enter x2= "))
y2=int(input("Enter y2= "))
#Formula to calculate the distance between the 2 points
d=((x2-x1)**2+(y2-y1)**2)**(1/2)
print("The distance between the two points is (in units): ",d
```

(2)

```
#(a)Creating a variable 'a' and assigning the string to it
a="Python programming"
#(b)Using slice string operator to access the character "i" from
the variable by its index number
print("The 15th character in the string is: ",a[15])
#(c)Using string function len() to find the length of the string
print("The length of the string is: ",len(a))
#(d)Using range slice string operator to print the slice "Python"
from the variable
print(a[0:6])
#(e)Using range slice string operator to print the slice "program"
from the variable
print(a[7:14])
```

```
#(f)Using a combination of range slice string operator and
concatenation string operator to get the slice "thing" from the
variable
print(a[2:4]+a[15:18])
#(g)Using string function upper() to convert all the characters in
the string into uppercase
print("Converting the string into uppercase: ",a.upper())
#(h)Creating another variable 'b' and assigning another string to
it
b=" is interesting."
#Concatenating both the strings
print(a+b)
```

Output(s):

(1)

```
Enter the coordinates of the first point(x1,y1)
Enter x1= 3
Enter y1= 4
Enter the coordinates of the second point(x2,y2)
Enter x2= 4
Enter y2= 5
The distance between the two points is (in units):  1.4142135623730951
```

(2)

```
The 15th character in the string is:  i
The length of the string is:  18
Python
program
thing
Converting the string into uppercase:  PYTHON PROGRAMMING
Python programming is interesting.
```

Conclusion:

After the successful execution of the assignment, we learnt about basic variables, mathematical functions, strings and almost all string functions in Python.

Post Lab Descriptive Questions:-

1. **How will you perform the following operations?**
   a. **Obtain integer quotient and remainder while dividing 28 with 6.**

Ans- We can obtain an integer quotient and remainder while dividing 28 with 6 using the floor division operator (28 // 6)

b. **Obtain remainder on dividing 3.45 with 1.22**
Ans- We can obtain the remainder on dividing 3.45 with 1.22 using the modulus operator (3.45 % 1.22)

c. **Obtain 4 from 3.5567.**
Ans- We can obtain 4 from 3.5567 by using the built-in round() function (round(3.5567))

d. **Print decimal equivalent of binary '1100001110'.**
Ans- We can use the built-in int() function to convert the binary string to its decimal equivalent, and then use the built-in print() function to output the result. In the int() function the second argument would be 2 to specify that the input string is in base 2 (ie its binary). The function would then return the decimal equivalent of the binary string which would then be printed as the output.

e. **Obtain real part, imaginary part and conjugate of 4 + 2j.**
Ans- To obtain real part, imaginary part and conjugate of 4 + 2j, we can use the built-in functions 'real', 'imag' and 'conjugate' from the 'cmath' module. The 'cmath' module needs to be imported before using these built-in functions.

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

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

Date: _____ **Signature of faculty in-charge**