Batch: E1 **Roll No.: 16010321005**

Experiment / assignment / tutorial No.1

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 volume of a rectangular prism and diagonal length
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 display 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, input function convert 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
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

Function Name	Description
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
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

Function Name	Description
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
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 representing length, width, height and unit. Assign each of them a value as user input using the input() function. Calculate volume and diagonal length of rectangular prism by using operators in python and basic built in math functions.
Finally, use print() to display "The volume of the rectangular prism is [calculated volume] cubic [unit]." "Diagonal length of the rectangular cube is [diagonal length] [unit]" in the output.

2) a) Create a variable and assign it the string "Python programming"
b) Access the "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.)

```
import math

length=float(input("Enter Length: "))
height=float(input("Enter Height: "))
width=float(input("Enter Width: "))

unit=input("enter unit: ")

if length<0 or height<0 or width<0:
  print("Invalid Input")
elif length == 0 or height == 0 or width == 0:
  print("Invalid dimensions try again")
else:
  volume=length*height*width
  d_length=math.sqrt(pow(length,2)+pow(height,2)+pow(width,2))

print(f'The Volume and diagonal length are {volume,unit} and {d_length,unit}')
```

2.)

```
str1 = "Python programming" #a)
print("a: ", str1)
print("b: ",str1[-3]) #b)
print("c: ",len(str1)) #c)
print("d: ",str1[:6]) #d)
print("e: ",str1[7:14]) #e)
print("f: ",str1[2:4]+string1[-3:]) #f)
print("g: ",str1.upper()) #g)

str2 = " is interesting"

print("f: ",str1+str2) #h)
print("\n")
print("<------i)------->\n")
print(str1.capitalize())
print(str1.count('p'))
print(str1.isalpha())
print(str1.isdigit())
print(str1.isupper())
```

Output(s):

1.)

```
Enter Length: 10
Enter Height: 20
Enter Width: 30
enter unit: metres
The Volume and diagonal length are (6000.0, 'metres') and (37.416573867739416, 'metres')
```

2.)

```
a:  Python programming
b:  i
c:  18
d:  Python
e:  program
f:  thing
g:  PYTHON PROGRAMMING
f:  Python programming is interesting


<------i)------->

Python programming
1
False
False
False
```

Conclusion:

The following problem statements were solved using the knowledge the python programming like various built in math function, operators and string methods.

Post Lab Descriptive Questions :-

1. **What is the difference in C language and Python?**

Ans-

Features of C	Features of Python
C is a general-purpose procedural computer programming language	Python is an interpreted, high level, general-purpose programming language.
Compiled programs execute faster as compared to interpreted programs.	Interpreted programs execute slower as compared to compiled programs.
Program syntax is harder than Python.	It is easier to write a code in Python as the number of lines is less comparatively.
C is generally used for hardware	Python is a general-purpose

related applications.	programming language.
Pointers are available in C	No pointer functionality is available in Python.

2. **Explain different data types in python.**
Ans-Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes. Following are the standard or built-in data type of Python:
    - Numeric-Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.
    - Sequence Type-Sequences allows to store multiple values in an organized and efficient fashion.
    - Boolean-Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are false (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.
    - Set-Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.
    - Dictionary-Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

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: 15/04/2022 **Signature of faculty in-charge**