Aatmaj 16010121110

| |
|---|
| **Batch: B2 Roll No.: 16010121110** |
| **Experiment / assignment / tutorial No.** |
| **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 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<br>9.0//2.0 = 4.0,<br>-11//3 = -4,<br>-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 | M not in a will |

| | | |
|---|---|---|
| | in the given string | 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 |

| Function Name | Description |
|---|---|
| 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 |
| 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 |

| Function Name | Description |
|---|---|
| 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 |
| replace() | Replaces all occurrences of a substring with another substring |
| rfind() | Returns the highest index of the substring |

| Function Name | Description |
|---|---|
| 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 |
| translate() | Modify string according to given translation mappings |
| upper() | Converts all lowercase characters in a string into uppercase |

| Function Name | Description |
|---|---|
| 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:**

**Program 1**

```
#Aatmaj 16010121110
#programe to calculate the volume and diagonal of prism.
import math
```

```
while True: #accept valid inputs
   try:
      length=int(input("Please enter the length ")) #get length
      breadth=int(input("Please enter breadth ")) #get breadth

      volume=length*breadth*breadth #calculate volume
      diagonal_length=math.sqrt(length**2+2*breadth**2) #calculate length of
diagonal
      #negative numbers will be considered as positive numbers and minus sign will be
treated as a mistake.
      #print outputs
      print(volume)
      print(diagonal_length)
      break
   except ValueError: #error if strings are printed
      print("Please enter a valid number. retry.....")

'''
Test cases
Input-
Length=10
breadth=20

Output-
volume 4000
diagonal length 30

Input-
Length=12
breadth=14

Output-
volume 2352
diagonal length 23.151
'''
```

**Program 2**

```
#Aatmaj 16010121110
#string functions in Python.

string="Python programing" #Create a variable and assign it the string "Python programming"
print(string[-3]) #Access the "i" from the variable by index and print it
print(len(string)) #Find the length of the string
print(string[0:5]) # Print the slice "Python" from the variable
print(string[-10:]) #Print the slice “program” from the variable
thing=string[2:4]+string[-3:] #Get the string “thing” from the variable
print(thing)
print(string.upper()) #Convert string into uppercase
string2="is intersting" #Create another variable and assign it the string “ is interesting” now concatenate   both the strings
print(string+" "+string2)

#some string functions
print(string.capitalize())  #Converts the first character of the string to a capital (uppercase) letter
print(string.count('i'))  #Returns the number of occurrences of a substring in the string.
print(string.isalnum())  #Checks whether all the characters in a given string is alphanumeric or not
print(string.isdigit())  #Returns “True” if all characters in the string are digits
```

**Output(s):**

**Program 1-**

```
Please enter the length avb
Please enter a valid number. retry.....
Please enter the length 12
Please enter breadth edf
Please enter a valid number. retry.....
Please enter the length 12
Please enter breadth 3er
Please enter a valid number. retry.....
Please enter the length 23
Please enter breadth 43
42527
65.01538279515087
```

**Program 2-**

```
i
17
Pytho
programing
thing
PYTHON PROGRAMING
Python programing is intersting
Python programing
1
False
False
```

**Conclusion:**

**We have understood how strings work in python. Also we have understood the basic arithmetic and working of python language. We used string functions for uppercase, capitalization, concatenation, etc.**

**Post Lab Descriptive Questions :-**

1. **What is the difference in C language and Python?**
   1) **Python s interpreted unlike C.**

2) **Python is highly portable- Python is a device independent language. Like Java, It can be run on any device (Linux, windows, mac or any other) with the same code.**
3) **Python is slow unlike C which is fast.**
4) **Python is easier to learn than C with a simpler syntax**

2. **Explain different data types in python.**

**There are three main data types in python**

1) **Int integer data type consists of float, negative and positive numbers.**
2) **Str string data type consists of a list of characters. A character is simply a list of characters with length zero.**
3) **Complex complex numbers are present in python as a separate data type**

**Data structures like tuple, dictionary, lists are present.**

**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
4. https://dev.to/aatmaj/learning-python-basic-course-day-1-introduction-and-installation-ee8

**Date: 4-11-22** **Signature of faculty in-charge**