



* Practical- 01 *

Aim:

Python program to calculate area of triangle, rectangle circle.

Theory: Every value in python has datatype, since everything is an object in python programming, datatypes are actually classes and variables are instance of these classes.

Operators: Operators are special symbol in python that carry out arithmetic or logical computation. The value that operator operates on is called the operand.

e.g. Add (+), Subtract (-), multiply (*), Divide (/), modulus (%).

Type casting:- The type casting is a method to convert the variable datatype into a certain datatype in order to the operations required to be performed by user.

math module :- Math module provide functions to deal with both basic operations such as addition (+), subtraction (-), multiplication (*), division (/), modulus (%) and other advance operations like trigonometric, logical, logarithmic, exponential, functions etc.



Program: ① Area of circle.

import math

rad = input

rad = int(input("Enter radius of circle"))

Area = math.pi * pow(rad, 2)

print ("Area of circle is", Area)

O/P : Enter radius of circle 5

Area of circle is 78.5

② Area of triangle

base = int(input("Enter base of triangle:"))

height = int(input("Enter height of triangle"))

Area = (1/2) * (base * height)

print ("Area of triangle: ", Area)

O/P : Enter base of triangle: 2

Enter height of triangle 3

Area of triangle is : 3

③ Area of Rectangle:

len = int(input("Enter length of rectangle"))

width = int(input("Enter width of rectangle"))

Area = len * width

print ("Area of Rectangle is: ", Area)

O/P : Enter length of rectangle 3

Enter width of rectangle 4

Area of Rectangle is : 12

~~conclusion~~ Hence we have successfully calculated the area
of circle, triangle and Rectangle.

~~But~~
S



* Practical - 02 *

Aim: Write a python program to calculate / find union of two list.

Theory: List: python list are one of the most versatile data type that allow us to work with multiple elements at once.

Syntax: `a = ['Elements', Numbers]`

Create a python list: In python a list is created by placing elements inside square brackets `[]`, separated by commas. A list can have any number of items and they may be of different types.

Accessing a list: ① we can use the `index operator []` to access an item in a list. In python, indices start at 0. So a list having five elements will have an index from 0 to 4.

E.g. `My-list = ['P', 'r', 'o', 'b', 'e']`
`print(my-list[0]) # P`
`print(my-list[2]) # o`.

② using slice method (:) :

Syntax: `List[initial : End : Index Jump]`

E.g. `print(My-list[0:2]) # ['P', 'r']`.

③ Negative index: We can access the elements from last to negative index. E.g `print(my-list[-1]) # 'e'`.



We can Add or change list elements by using.

① append(): Used for appending, dealing/ adding element to list.

e.g. `My_list.append(element)`

e.g. `My_list.append('t')`.

`print(my_list) # ['p', 'g', 'o', 'b', 'e', 't']`

② insert(): Used to insert element at specific position

Syntax: ~~List.~~ List.insert(position, element)

e.g. `my_list(2,'c') # ['p', 'g', 'c', 'o', 'b', 'e', 't']`

③ extend(): Adding content of List2 to the list]

Syntax: List1.extend(List2)

e.g. `List2 = [1, 2, 3]`

`mylist.extend(List2)`

`print(my_list) # ['p', 'g', 'e', 'o', 'b', 'e', 't', 1, 2, 3]`

④ count(): calculates total occurrence of the given element in the list.

Syntax: List.count(element)

e.g. `List.count('P') # 1`

* Deleting list elements.

⑤ pop(): Index is not necessary parameter if not mentioned it will take the last index.

Syntax: ~~List.pop([index])~~

e.g. `mylist.pop() # 3`



⑥ `del()`: Element to be deleted is mentioned using name and index.

Syntax: `del list[index]`

e.g. `del mylist[-1]`

⑦ `remove()`: Element to be deleted is mentioned using list-name & element.

e.g. Syntax: `list.remove(element)`

e.g. `mylist.remove(1)`

⑧ `clear()`: The `clear()` method only empties the given list.

Syntax: `List.clear()`

e.g. `my-list.clear() # []`

* To know the type of list the data it contains, we use `type()` method.

⑨ `type()`: It takes one parameter whose type is to be known.

Syntax: `type(my-list)`.

⑩ `Input()` method / function: The `input()` function takes input from the user and returns it.

Syntax: `input([prompt])`

e.g. `x = input("Enter a number")`.

Note: prompt (optional) - a string that is written to standard output without trailing new line.



* Union function:-

The union() method returns a set that contains all items from the original set, and all items from the specified set(s).

You can specify as many sets you want, separated by commas.

It doesn't have to be set, it can be any iterable object

Syntax : set.union(set1, set2...)

example : x = {"a", "b", "c"}

y = {"f", "d", "a"}

z = {"c", "d", "e"}

result = x.union(y, z)

print(result)

O/P : {"a", "b", "c", "d", "e", "f"}

Program: with Repetation.

```
def Union(list1, list2)
    return list1+list2
```

list1 = [23, 15, 2, 14, 10]

list2 = [30, 30, 14, 16, 16]

print(Union(list1, list2))

O/P : [23, 15, 2, 14, 10, 30, 30, 14, 16, 16]



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

Without Repetation.

```
def Union(list1, list2)
    return list(set(list1) | set(list2))
```

list1 = [23, 15, 2, 14, 14, 16, 20, 52]

list2 = [2, 48, 15, 12, 26, 32, 47, 54]

print(Union(list1, list2))

O/P : [32, 2, 12, 14, 15, 16, 48, 47, 20, 52, 54, 23, 26]

Conclusion: Hence we have successfully implemented Union of two lists.

~~Red
07/03/22~~

*** Practical - 03 ***

Aim : Write a python program to find the intersection of two lists.

Theory: List : python lists are one of the most versatile data type that allow us to work with multiple elements at once.

Syntax : `a = ['Elements', Numbers]`

Creating a python list : In a python list is created by placing elements inside square brackets `[]`. Separated by commas. A list can have any number of items and they may be of different types.

Accessing a list : we can access a list by using different methods.

i) by using index operator `[]` : We can use the index operator `[]` to access an item in a list. In python, indices start at 0. So a list having five elements will have an index from 0 to 4.

e.g. `my_list = ['P', 'R', 'O', 'B', 'E']`
`print(my_list[0])` # P
`print(my_list[2])` # O

ii) Using slicing operator `:`

Syntax : `list[Initial:End:IndexJump]`
e.g. `print(my_list[0:2])` # ['P', 'R']



③ By using Negative Index (-i): We can access the list elements from last by using negative indices.

e.g. print(my-list [-1]) # 'e'.

We can add or change list elements by using.

① append(): It is used for appending, i.e., adding element to list.

Syntax: list.append(element)

e.g. : my-list.append('t')

print(my-list) # ['p', 'r', 'o', 'b', 'e', 't']

② insert(): Used to insert element at specific position:

Syntax: list.insert(position, element).

e.g. my-list.insert(2, 'e') # ['p', 'r', 'e', 'o', 'b', 'e', 't']

③ extend(): Adding content of list2 to the list1.

Syntax: list1.extend(list2)

e.g. list2 = [1, 2, 3]

my-list.extend(list2)

print(my-list) # ['p', 'r', 'o', 'b', 'e', 't', 1, 2, 3]

④ count(): calculates the total occurrence of the given element in a list.

Syntax: list.count(element)

e.g. list.count('P') # 1.



* Deleting List Elements.

⑤ `pop()`: Index is not necessary parameter if not mentioned it will take the last index.

Syntax: `list.pop([index])`,

e.g. `my-list.pop()` # 3

⑥ `del()`: Element to be deleted is mentioned using name and index.

Syntax: `del list[index]`

e.g. `del my-list[-1]`.

⑦ `remove()`: Element to be deleted is mentioned using list name & element.

Syntax: `list.remove(element)`

e.g. `mylist.remove(1)`

⑧ `clear()`: The `clear()` method only empties the given list.

Syntax: `list.clear()`

e.g. `my-list.clear()` # []

⑨ `type()`: It takes one parameter whose type is to be known.

Syntax: `type(my-list)`.

For loop :-

A for loop is used for iterating over a sequence. With for loop we can execute a set of statements, once for each item in a list type, set etc

e.g. if x in range(10):
print(x).

function in python :-

A function is A block of code which only runs when it is called.

You can pass data, known as parameters, into a function. It can ~~result~~ return data or a result.

e.g. def my_function():
print("Hello from a function")

Intersection () :-

The intersection method returns a set that contains the similarity between two or more sets. -

Syntax: set.intersection(set1, set2, ...etc)

e.g. $x = \{'a', 'b', 'c'\}$
 $y = \{'c', 'd', 'e'\}$

result = x .intersection(y)

print(result)

O/P : $\{'c'\}$



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

Program : # intersection

```
def intersection(list1, list2)
```

```
# return [value for value in list1 if value in list2]
```

```
list1 = [4, 9, 1, 17, 11, 26, 28, 54, 69]
```

```
list2 = [9, 9, 74, 21, 45, 11, 63, 28, 26]
```

```
print(intersection(list1, list2))
```

Output : [9, 11, 26, 28]

Conclusion : Hence we have successfully implemented python program for finding intersection of two lists.

~~6/10/22~~



* Practical - 04 *

Aim :

Write a python program to remove 'i' th occurrence of given word / element in a list where word repeats

Theory :

By taking another list. make new list say newlist
 iterate the elements in a list and check if the word to be removed matches the elements and the occurrence number, otherwise, append the element to new list .

Algorithm :

- ① start
- ② take / create a list with same repeating elements
- ③ initialize count variable with 0.
- ④ take input and typecast into int. i.e. position of word.
- ⑤ take input word to remove from list.
- ⑥ iterate through for loop with index i in list.
 if word == i
 increase count by 1.
 if count \neq position
 newlist.append(i) # append words to newlist with position
 else:
 newlist.append # append word to newlist
- ⑦ if count == 0. print("item not found")
- ⑧ else print(newlist)
- ⑨ end.

Program:

```
# Remove item element occurrence from a list.  
list1 = ['a', 'b', 'c', 'd', 'a', 'a', 'f']  
pos = int(input("Enter position/occurrence:"))  
word = input("Enter word/element want to remove")  
count = 0  
list2 = []  
  
for i in list1:  
    if (i == word):  
        count = count + 1  
    if count != pos:  
        list2.append(i)  
    else:  
        list2.append(i)  
  
if count == 0:  
    print("Item not found")  
else:  
    print("Updated list is:", list2)
```

Output: Enter position/occurrence : 2

Enter word/element want to remove : a

Updated list is: ['a', 'b', 'c', 'd', 'a', 'f']

Conclusion: Hence we have successfully implemented python program which removes item occurrence of an element from a list.



Practical No: 05

Aim:- write a python program to count the occurrences of given word in the given string sentence.

Theory:

- split() method:

The split() method splits a string into a list we can specify the operator, default separator is any whitespace.

Syntax:

String.split(separator, maxsplit)

separator: optional specifies the separator to use when splitting the string. by default any whitespace is a separator.

maxsplit: optional specifies how many splits to do. default value is -1. which is all occurrences.

e.g. text = "Hi, my name is Subodh. I am learning python".

```
x = text.text.split(",")  
print(x).
```

O/P: ['Hi', 'my name is subodh', 'I am learning python']

Algorithm:

1. Start.
2. Take string as input.
3. Split the string by space.
4. take variable count = 0
5. take word input.
6. Now run loop and check if our string is equal to given word.
7. If condition is true then we increment count
8. print count.
9. End.

Program:

```
s = input(" Enter the sentence : ")  
lI = s.split(" ")  
word = input(" Enter the word to count : ")  
count = 0
```

```
for i in lI:  
    if i == word:  
        count += 1.
```

```
print(count).
```

O/P :-

① Enter the sentence: I am Subodh Gaikwad.
I am learning python.
Enter the word to count : python.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

② Enter the sentence : I am Subodh Gaikwad
i am learning python.

Enter the word to count : am

2

Conclusion: we have successfully implemented the program to calculate the count of word in a string.



Practical No: 06

Aim: Write a python program to create if a substring is present in given string or not.

Theory:

- o Strings

String in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello".

Assigning a string to a variable is done with the variable name followed by an equal sign and the string.

a = "hello".

We can assign a multiline string to a variable by using three quotes:

e.g. s = """ I am Subodh Gaikwad, I am CSE student """

String Methods :

1) capitalization():

The capitalize() method returns a string when the first character is uppercase, and the rest is lower case.

Syntax: String.capitalize()

2) casefold():-

The `casefold()` method returns a string where all the characters are lower case.

Syntax: `String.casefold()`.

3) index():-

The `index()` method finds first occurrence of specified value.

Syntax: `String.index(value, start, end)`

4) split():-

The `split()` method splits a string into a list e.g. `String.split(separator, maxsplit)`

5) strip():-

The `strip()` method removes any leading and trailing characters.

Syntax: `String.strip(characters)`

6) find():

The `find()` method finds the first occurrence of the specified value.

The `find()` method returns -1 if the value is not found.

Syntax: `String.find(value, start, end)`

7) replace():

The `replace()` method replaces a specified phrase with another specified phrase.



Syntax : string.replace (oldvalue, newvalue, count)

8) rindex()

The rindex() method finds the last occurrence of the specified value.

Syntax : string.rindex (value, start, end)

Algorithm :

1. Start
2. take the string as input.
3. Take the sub-string as input.
4. find the substring in string.

string.find (substring)

5. print the result whether it found or not
6. End.

Program :

```
s = input("Enter a string: ")
ss = input("Enter substring to find: ")
i = s.find(ss)
if i >= 0:
    print("Sub-string found at index", i)
else:
    print("Sub-string not found!")
```

O/P :

- ① Enter a string: I am learning python.
Enter a substring to find: python.
Sub-string found at index 14.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

- (2) Enter a string: I am a CSE student.
Enter a substring to find: Subodh
Sub-string not found.

Conclusion: we have successfully implemented
the program to check if substring is
present in given string.



Practical No : 07

Aim : Program to map two list into dictionary .

Theory :

Dictionary :

Dictionaries are used to store data values in key value pairs.

A dictionary is a collection which is ordered, changeable and not allow duplicates.

e.g. dict1 = { "brand": "ford",
"model": "mustang",
"year": 1964,
}

print(dict)

O/P: {'brand': 'Ford', 'model': 'mustang', 'year':
1964}

zip() function :-

The zip() function returns a zip object, which is an iterator of tuples where the first item in each passed iterator is paired together, and the second item in each passed iterator are paired together etc.

If the passed iterators have different length the iterator with the least items decided the length of the new iterator .



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

Syntax :- zip (iterator1, iterator2, iterator3, ...)

example :

a = ("Name", "class", "Branch")

b = ("Vaibhav", "SE", "CSE")

x = zip (a,b)

print (dict (x))

O/P:

{ "Name": "Vaibhav", "class": "SE", "Branch": "CSE" }

Algorithm :

1. start.

2. make two lists.

3. Initialise the lists.

4. Zip two lists using zip() function and convert it to dictionary using dict().

5. print the dictionary.

6. End.

Program:-

using zip() function:

list1 = ["Name", "class", "Branch"]

list2 = ["Vaibhav", "SE", "CSE"]

dict1 = dict (zip (list1, list2))

print (dict1).



O/P:

```
{'Name': 'Vaibhav', 'class': 'SE', 'Branch':  
'CSE'}
```

using list comprehension:

```
list1 = ['Name', 'class', 'Branch']  
list2 = ['Vaibhav', 'SE', 'CSE']
```

```
dict1 = {list1[i]: list2[i] for i in range(len(list2))}  
print(dict1)
```

O/P:

```
{'Name': 'Vaibhav', 'class': 'SE', 'Branch': 'CSE'}
```

Conclusion:- Hence we have successfully implemented the program to map two lists into dictionary

Practical No : 08

Aim :- Program to count frequency of words appearing in a string using a dictionary.

Theory :-

Dictionary :-

Dictionaries are used to store data in key values pairs.

e.g. dict1 = { key1:value1,
key2: value2;
:
}

split() function :-

The split() function splits a string into a list.

string.split(separator, maxsplit).

key() function :-

The keys() method returns a view object. The view object contains the key of the dictionary as a list.

The view object will reflect any change done to the dictionary.

Syntax : dictionary.keys().



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

Algorithm:

1. start
2. Take the input string.
3. split the string and store it in list.
4. store the count of each word from list in a dictionary.
5. print count of each word from dictionary.
6. End.

Program:

```
s = input("Enter the string")  
list1 = s.split("")  
count = {}
```

```
for i in list1:  
    if i in count.keys():  
        count[i] += 1.  
    else:  
        count[i] = 1.
```

```
print(count).
```

O/P:

```
Enter the string: i am subodh i am student  
in csmss i am in second year  
{} i : 3,  
've Subodh': 1,
```



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

31

'studying' : 1,

'in' : 2,

'csmss' : 1,

'am' : 3,

'second' : 1,

'year' : 1 }

Conclusion :- Hence we have successfully implemented the program to count the frequency of which in string using dictionary.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date:

Practical No.09

Aim: Write a python program to compute the diameter, circumference, and volume of a sphere using class.

Theory:

class:-

A class is like an object structure, or a blueprint for creating objects.

To create a class, use the keyword class

e.g.

class MyClass:

x = 5.

Object creation :-

We can use the class named MyClass to create objects.

e.g.

PI = MyClass().

--init--() function:

All classes have a function called --init-- which is always executed when the class is being initiated use the --init--() function to assign values to object properties, or other operations that are necessary to do when the object is being created.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

The self parameter :-

The self parameter is a reference to the current instance of the class, and it is used to access variables that belong to the class.

Delete Objects :-

We can delete objects by using the del keyword.

e.g. del objectName.

Algorithm:

1. START
2. create a class named as Sphere
3. define the `--init()` method to take radius
4. Define the methods to calculate perimeter, volume and diameter.
5. create a object of class Sphere and pass the radius as argument.
6. Call various methods of class for calculating perimeter, and volume, diameter.
7. END.

Program:-

```
import math.
```

```
class Sphere:
```

```
    def __init__(self, radius):
```

```
        self.radius = radius.
```



```
def perimeter(self):
```

```
    return (2 * math.pi * self.radius)
```

```
def diameter(self):
```

```
    return (self.radius * 2)
```

```
def volume(self):
```

```
    return ((4 * math.pi * self.radius ** 3) / 3)
```

```
s1 = Sphere(7)
```

```
print ('the circumference of sphere is',  
      round(s1.perimeter(), 2))
```

```
print ('the diameter of sphere is', round(s1.diameter(), 2))
```

```
print ('The volume of sphere is, round(s1.volume(), 2))
```

O/P:

The circumference of sphere is 43.98

The diameter of sphere is 14

The volume of sphere is 1436.75.

Conclusion:-

We have successfully implemented the program to calculate the circumference, diameter and volume of Sphere using a class.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad

Date:

Practical No :10

Aim:- Write a python program to read file and capitalize the first letter of every word in the file.

Theory :-

opening a file :-

The key function for working with files in python is the open() function.

The open() function takes two parameters filename and mode.

mode :-

"r" - Read. Open a file for reading.

"a" - Append. Open a file for appending.

"w" - write. opens a file for writing.

"x" - create. creates the specified file.

Syntax :-

```
f = open ("filename.txt"; "mode")
```

close() method :-

The close() method closes an open file.

Syntax :-

```
file.close()
```



read() method :-

The read() method returns the specified number of bytes from the file. Default is -1 which means the whole file.

Syntax:-

file.read()

readline() method :-

The readline() method returns one line from the file.

We can also specify how many bytes from the line to return , by using the size parameter.

Syntax :-

file.readline(size).

Algorithm :-

1. Start

2. Create a text file named as demo.

3. Store some sentence in demo.txt.

4. Open demo.txt file in read mode.

5. Print the content of file with first letter of word capitalize.

6. End.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

Program :-

demo.txt is

Hi, we are students of CSE

currently, we are studying in B.tech
Second year

Academic year 2021-22.

Python program:

```
f = open("demo.txt", "r")
```

```
for line in f:
```

```
    print(line.title())
```

O/P:

Hi, we are the students of CSE

currently, we are studying in B.tech Second
Year

Academic Year 2021-22.

Conclusion: We have successfully implemented
the python program to read file and capitali-
the first letter of every word.