

In

**Course Code: UGCA-1917 Course**

**Name: Python Laboratory**

**Submitted to :**

**Submitted by :**

1. Compute sum, subtraction, multiplication, division and exponent of given variables input by the user.
2. Compute area of following shapes: circle, rectangle, triangle, square, trapezoid and parallelogram.
3. Compute volume of following 3D shapes: cube, cylinder, cone and sphere.
4. Compute and print roots of quadratic equation  $ax^2+bx+c=0$ , where the values of a, b, and c are input by the user.
5. Print numbers up to N which are not divisible by 3, 6, 9,, e.g., 1, 2, 4, 5, 7,....
6. Write a program to determine whether a triangle is isosceles or not?
7. Print multiplication table of a number input by the user.
8. Compute sum of natural numbers from one to n number.
9. Print Fibonacci series up to n numbers e.g. 0 1 1 2 3 5 8 13.....n
10. Compute factorial of a given number.
11. Count occurrence of a digit 5 in a given integer number input by the user.
12. Print Geometric and Harmonic means of a series input by the user.
13. Evaluate the following expressions: a.  $x-x^2/2!+x^3/3!- x^4/4!+ \dots x^n/n!$
14. b.  $x-x^3/3!+x^5/5!- x^7/7!+ \dots x^n/n!$
15. Print all possible combinations of 4, 5, and 6.
16. Determine prime numbers within a specific range.
17. Count number of persons of age above 60 and below 90.
18. Compute transpose of a matrix.
19. Perform following operations on two matrices.
20. 1) Addition 2) Subtraction 3) Multiplication
21. Count occurrence of vowels.
22. Count total number of vowels in a word.
23. Determine whether a string is palindrome or not.
24. Perform following operations on a list of numbers: 1) Insert an element 2) delete an element 3) sort the list 4) delete entire list
25. Display word after Sorting in alphabetical order.
26. Perform sequential search on a list of given numbers.
27. Perform sequential search on ordered list of given numbers.
28. Maintain practical note book as per their serial numbers in library using Python dictionary.
29. Perform following operations on dictionary 1) Insert 2) delete 3) change
30. Check whether a number is in a given range using functions.
31. Write a Python function that accepts a string and calculates number of upper case letters and lower case letters available in that string.
32. To find the Max of three numbers using functions.
33. Multiply all the numbers in a list using functions.
34. Solve the Fibonacci sequence using recursion.
35. Get the factorial of a non-negative integer using recursion.
36. Write a program to create a module of factorial in Python.
37. Design a Python class named Rectangle, constructed by a length & width, also design a method which will compute the area of a rectangle.
38. Design a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.
39. Design a Python class to reverse a string 'word by word'.
40. Write a Python program to read an entire text file.
41. Design a Python program to read first n lines of a text file.
42. Construct a Python program to write and append text to a file and display the text.

In

# 1) Compute sum, subtraction, multiplication, division and exponent of given variables

```
1 def main():
2     x, y = input("Enter a two value: ").split()
3     print(f"arg 1: ", x)
4     print(f"arg 2: ", y)
5     print(f("""
6     Supported Operations:
7     .... +
8     .... -
9     .... *
10    .... /
11    .... ^
12    /n
13    """))
14
15    op = input("And the operation to be performed:\t")
16
17    def do_operation(op):
18        match op:
19            case (+):
20                result = arg1+arg2
21            case (-):
22                result = arg1-arg2
23            case (*):
24                result = arg1*arg2
25            case (/):
26                result = arg1/arg2
27            case (^):
28                result = arg1**arg2
29            case _:
30                raise TypeError("not a supported type")
31        return (print({x} {op} {y} gives {result}))
32
33
34 if __name__ == "__main__":
35     main()
```

```
Enter a two value: 1 2
arg 1: 1
arg 2: 2
Supported Operations:
.... +
.... -
.... *
.... /
.... ^

And the operation to be performed:    *
1 * 2 gives 2
```

## 2) Compute area of following shapes: circle, rectangle, triangle, square, trapezoid and parallelogram.

```
In [ ]: 1 while(True):
2     print("Menu :\n1. Circle\n2. Ractangle\n3. Triangle\n4. Square\n5. Trapezoi
3     c=int(input("Enter your choice : "))
4     area = 0
5
6     if c==1:
7         r=int(input("Enter the value of radius : "))
8         area = 3.13 * r *r
9     elif c==2:
10        l=int(input("Enter the value of Length : "))
11        b=int(input("Enter the value of Breadth : "))
12        area = b * l
13    elif c==3:
14        b=int(input("Enter the value of base : "))
15        h=int(input("Enter the value of height : "))
16        area = 0.5 * b * h
17    elif c==4:
18        s=int(input("Enter the value of side : "))
19        area = s*s
20    elif c==5:
21        a=int(input("Enter the value of top; "))
22        b=int(input("Enter the value of base : "))
23        h=int(input("Enter the value of height : "))
24        area = (a+b)/2 * h
25    elif c==6:
26        b=int(input("Enter the value of base : "))
27        h=int(input("Enter the value of height : "))
28        area = b * h
29    else:
30        print("Wrong choice.")
31
32    print(area)
33    choice=int(input("Do you want to continue? IF 'YES' press 1 : "))
34    if(choice!=1):
35        break
36
```

Menu :

1. Circle
2. Ractangle
3. Triangle
4. Square
5. Trapezoid
- 6.Parallelogram

Enter your choice : 2

Enter the value of Length : 1

Enter the value of Breadth : 2

2

Do you want to continue? IF 'YES' press 1 : 2

In

### 3) Compute volume of following 3D shapes: cube, cylinder, cone and sphere.

```
1 while(True):
2     print("Menu :\n1. Cube\n2. Cylinder\n3. Cone\n4. Sphere\n")
3     c=int(input("Enter your choice : "))
4     vol = 0
5
6     if c==1:
7         r=int(input("Enter the value of radius : "))
8         vol = r *r *r
9     elif c==2:
10        r=int(input("Enter the value of radius : "))
11        h=int(input("Enter the value of heigth : "))
12        vol = 3.14 * r *r *h
13    elif c==3:
14        r=int(input("Enter the value of radius : "))
15        h=int(input("Enter the value of heigth : "))
16        vol = 0.33 * 3.14 * r*r*h
17    elif c==4:
18        r=int(input("Enter the value of radius : "))
19        vol = (4/3)*3.14 * r *r *r
20    else:
21        print("Wrong choice.")
22
23    print(vol)
24    choice=int(input("Do you want to continue? IF 'YES' press 1 : "))
25    if(choice!=1):
26        break
27
```

Menu :

1. Cube
2. Cylinder
3. Cone
4. Sphere

Enter your choice : 2

Enter the value of radius : 4

Enter the value of heigth : 2

100.48

Do you want to continue? IF 'YES' press 1 : 2

## 4) Print roots of quad. eq. inputted

```
In [ ]: 1 import cmath
2 a = int(input("Input a: "))
3 b = int(input("Input b: "))
4 c = int(input("Input c: "))
5
6 d = (b**2) - (4*a*c)
7
8 solution1 = (-b + cmath.sqrt(d))/(2*a)
9 solution2 = (-b - cmath.sqrt(d))/(2*a)
10
11 print("The solutions to the quadratic " "equation are {} and {}".format(solution1, solution2))
```

Input a: 1

Input b: 2

Input c: 3

The solutions to the quadratic equation are (-1+1.4142135623730951j) and (-1-1.4142135623730951j)

## 5 ) N numbers not divisble by 3,6,9...

```
In [ ]: 1 N = int(input("Enter : "))
2
3 for i in range(1,N+1):
4     if ((i%3)!=0) or ((i%6)!=0) or ((i%9)!=0):
5         print(i,"\n")
```

Enter : 10

3

6

9

## 6 ) Triangle - isosceles or not?

```
In ]: 1 side1 = int(input("side A : "))
2 side2 = int(input("side b : "))
3 side3 = int(input("side c : "))
4
5 if side1 == side2 or side2 == side3 or side1 ==side3:
6     print("This is an isosceles triangle!")
7 else print("Not an Isosceles ")
```

side A : 1

side b : 2

side c : 1 This is an  
isosceles triangle!

In

## 7) Multiplication table using input()

In [ ]:

```
1 num = int(input("Input: "))
2 for i in range(0,num+1):
3     print(num, 'x\t', i, '=\t', num*i)
```

Input: 10

```
10 x      0 =      0
10 x      1 =     10
10 x      2 =     20
10 x      3 =     30
10 x      4 =     40
10 x      5 =     50
10 x      6 =     60
10 x      7 =     70
10 x      8 =     80
10 x      9 =     90
10 x     10 =    100
```

## 8) Sum of N natural numbers

```
1 s=1
2 i=1
3 n=int(input("Input : "))
4 for i in range(0,n+1):
5     s=s+i
6     print ("S : ",s)
7 print ("sum of first 10 natural numbers",s)
```

Input : 10

```
S :  1
S :  2
S :  4
S :  7
S : 11
S : 16
S : 22
S : 29
S : 37
S : 46 S : 56 sum of first 10
natural numbers 56
```

## 9) Fibonacci N

```
In [ ]: 1 n = int(input("Enter n : "))
2 f = 0 #first
3 s = 1 #second
4
5 if(n == 1):
6     print(f)
7 if(n == 2):
8     print(f,s,end=" ")
9
10
11 for i in range(3,n+1):
12     new = f+s
13     print(new, end=" ")
14     f = s
15     s = new
```

Enter n : 7  
0 1 1 2 3 5 8

## 10 ) Factorial

```
1 n = int(input("Enter n : "))
2
3     if( n == 0 or n == 1):
4         print(1)
5
6         fact = 1
7         for i in range(1,n+1):
8             fact *= i
9
10 print("The factorial is : ",fact)
11 %time # time ~1 ns
```

Enter n : 100000  
The factorial is : 28242294079603478742934215780245355184774949260912248505789  
1808654297795090106301787255177141383116361071361173736196295147499618312391802  
272607340909383242200555696886678403803773794449612683801478751119669063860449

## 11 ) Occurence of 5 in an int. via input()

```
In [ ]: 1 a=(input("Enter List Elements:")).split()
2 l=list(map(int,a))
3 print("Original List:",l)
4 d=dict()
5 for i in l:
6     if i in d:
7         d[i]+=1
8     else:
9         d[i]=1
10 print("Printing count of each item:",d)
```

Enter List Elements:3 4 5 555  
Original List: [3, 4, 5, 555]



In

Printing count of each item: {3: 1, 4: 1, 5: 1, 555: 1}

## 12) Geometric and Harmonic Mean

```
1 import statistics
2 multiply = 1
3 values = [8,16,22,12,41]
4
5 AMean = statistics.mean(values)
6
7 for i in values:
8     multiply = (multiply)*(i)
9 GMean = (multiply)**(1/n)
10
11 HMean = (GMean**2)/AMean
12 print ('The Geometric Mean is: ' + str(GMean))
13 print ('The Harmoic Mean is: ' + str(HMean))
```

The Geometric Mean is: 16.916852032061655

The Harmoic Mean is: 14.453529428013562

## 13 ) Evaluate expressions

In [1]:

```
1 def factorial(n):
2     if n <= 1:
3         return 1
4     return n*factorial(n-1)
```

### a. $x - x^2/2! + x^3/3! - x^4/4! + \dots x^n/n!$

In [3]:

```
1 def sum_series(x, n):
2     sum = 0
3     for i in range(1, n+1, 1):
4         term = ((-1)**(i+1))*(x**(2*i-2)/factorial(2*i-2)) #Calculating nth term
5         sum += term
6     return sum
7
8 if __name__ == "__main__":
9     n = int(input("Enter n: "))
10    x = int(input("Enter x: "))
11    sum = sum_series(x, n)
12    print("\nSum of {} terms of series for x={}: {}".format(n, x, sum))
```

Enter n: 7

Enter x: 4

Sum of 7 terms of series for x=4: -0.6507594463150026

### b. $x - x^3/3! + x^5/5! - x^7/7! + \dots x^n/n!$

```
[8]: 1 def sum_series(x, n):
2     sum = 0
3     for i in range(1, n+1, 1):
4         term = ((-1)**(i+1))*(x**(2*i-1))/factorial(2*i-1)) #Calculating nth term
5         sum += term
6     return sum
7
8 if __name__ == "__main__":
9     n = int(input("Enter n: "))
10    x = int(input("Enter x: "))
11    sum = sum_series(x, n)
12    print("\nSum of {} terms of series for x={}: {}".format(n, x, sum))
```

Enter n: 7

Enter x: 4

Sum of 7 terms of series for x=4: -0.7560275115830666

## 14) Combinations 4,5,6

In [20]:

```
1 my_list = list(map(int, input("Enter the list of numbers (sep by space): ").split()))
2 my_list
```

Enter the list of numbers (sep by space): 4 5 6

Out[20]: [4, 5, 6] In [19]:

```
1 res = [(a, b, c) for a in my_list for b in my_list for c in my_list]
2 res
```

```
[(4, 4, 4), (4, 4, 5),
(4, 4, 6), (4, 5, 4),
(4, 5, 5), (4, 5, 6),
(4, 6, 4), (4, 6, 5),
(4, 6, 6), (5, 4, 4),
(5, 4, 5), (5, 4, 6),
(5, 5, 4), (5, 5, 5),
(5, 5, 6), (5, 6, 4),
(5, 6, 5), (5, 6, 6),
(6, 4, 4), (6, 4, 5),
(6, 4, 6), (6, 5, 4),
(6, 5, 5), (6, 5, 6),
(6, 6, 4), (6, 6, 5),
(6, 6, 6)]
```

In

## 15) Prime numbers within range

```
In [ ]: 1 s=eval(input('Enter start of the range : '))
2 e=eval(input('Enter end of the range : '))
3 for i in range(s,e+1):
4     if i > 1:
5         for n in range(2, (i//2+1)):
6             if(i%n)==0:
7                 break
8     else:
9         print(i,end=' ')
```

```
Enter start of the range : 2
Enter end of the range : 10
2 3 5 7
```

## 16) Count people aged between 60 and 90

```
In [ ]: 1 ageRange = {(60, 90): 0}
2 ages = map(int, input("Enter ages (seperated by spaces): ").split())
3
4 for age in ages:
5     if (60 <= age <= 90):
6         ageRange[(60,90)] += 1
7
8 for k, v in ageRange.items():
9     lower, upper = k
10    print(f"Employees in age group {lower} - {upper}: {v}")
```

```
Enter ages (seperated by spaces): 56 78 97 67
Employees in age group 60 - 90: 2
```

## 17 ) Transpose Matrix

```
In [ ]: 1 X = [[1,2,3],
2             [4 ,5,6],
3             [7 ,8,9]]
4
5 result = [[0,0,0],
6           [0,0,0],
7           [0,0,0]]
8 for i in range(len(X)):
9     for j in range(len(X[0])):
10        result[j][i] = X[i][j]
11 for r in result:
12     print(r)
```

```
[1, 4, 7]
[2, 5, 8]
[3, 6, 9]
```

## 18) Matrix Operation

```
1
2 X = [[1,2,3],
3       [4,5,6],
4       [7,8,9]]
5
6 Y = [[9,8,7],
7       [6,5,4],
8       [3,2,1]]
9
10
11 result = [[0,0,0],
12            [0,0,0],
13            [0,0,0]]
14
15 # ADDITION
16 for i in range(len(X)):
17     for j in range(len(X[0])):
18         result[i][j] = X[i][j] + Y[i][j]
19
20 for r in result:
21     print(r)
22
23 print("\n")
24
25 # SUBTRACTION
26 for i in range(len(X)):
27     for j in range(len(X[0])):
28         result[i][j] = X[i][j] - Y[i][j]
29
30 for r in result:
31     print(r)
32
33 print("\n")
34
35 # MULTIPLY
36 # iterating by row of A
37 for i in range(len(X)):
38     # iterating by column by B
39     for j in range(len(Y[0])):
40         # iterating by rows of B
41         for k in range(len(Y)):
42             result[i][j] += X[i][k] * Y[k][j]
43 for r in result:
44     print(r)
45
46 print("\n")
```

```
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
```

```
[-8, -6, -4]
[-2, 0, 2]
[4, 6, 8]
```

In

```
[22, 18, 14]
[82, 69, 56]
[142, 120, 98]
```

## 19) Vowels

In [ ]:

```
1 def countCharacter(str):
2
3     vowels = 0
4     consonant = 0
5     specialChar = 0
6     digit = 0
7     for i in range(0, len(str)):
8         ch = str[i]
9         ch = ch.lower()
10        if ( (ch >= 'a' and ch <= 'z') or (ch >= 'A' and ch <= 'Z') ):
11
12            if (ch == 'a' or ch == 'e' or ch == 'i' or ch == 'o' or ch == 'u'):
13                vowels += 1
14            else:
15                consonant += 1
16
17        elif (ch >= '0' and ch <= '9'):
18            digit += 1
19        else:
20            specialChar += 1
21
22        print("Vowels:", vowels)
23        print("Consonant:", consonant)
24        print("Digit:", digit)
25        print("Special Character:", specialChar)
26
27    str = "helloWorld"
28    countCharacter(str)
```

```
Vowels: 3
Consonant: 7
Digit: 0
Special Character: 0
```

## 21 ) palindrome or not?

In [ ]:

```
1 Pal = lambda s: s == s[::-1]
2
3 S = input("Input: ")
4 print("The '{}' is{}palindrome.".format(S, ' ' if (Pal(S)) else ' not '))
```

```
Input: hello hello
The 'hello hello' is not palindrome.
```

## 22 ) list of numbers:

1) Insert an element 2) delete an element 3) sort the list 4) delete

```
1 stack = []
2
3 stack.append('a')
4 stack.append('b')
5 stack.append('c')
6
7 print('Initial stack')
8 print(stack)
9
10 print('\nElements popped from stack:')
11 print(stack.pop())
12 print(stack.pop())
13 print(stack.pop())
14
15 print('\nStack after elements are popped:')
16 print(stack)
```

Initial stack  
['a', 'b', 'c']

Elements popped from stack:  
c b  
a

Stack after elements are popped: []

In [ ]:

```
1 ME = ["Lesson", "was ", "Short?"]
2 print(ME)
```

['Lesson', 'was ', 'Short?']

**delete**

In [ ]:

```
1 ME2 = ["Lesson", "Felt", "Long?"]
2 print(ME2)
```

['Lesson', 'Felt', 'Long?']

In [ ]:

```
1 del ME2
```

In [ ]:

```
1 ME2
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-5-161024c480b6> in <module>()
----> 1 ME2
```

NameError: name 'ME2' is not defined

In

In [ ]:

```
1 ME.remove("Lesson")
2 print(ME)
```

['was ', 'Short?', 'Nope!']

In [ ]:

```
1 ME.clear()
2 print(ME)
```

[]

In [ ]:

```
1 #Sort
2 vowels.sort()
```

In [ ]:

```
1 vowels
```

Out[21]: ['a', 'e', 'o', 'u']

In [ ]:

```
1 # Combine
2 test1 = [1, 2, 3]
3 test2 = [4, 5, 6]
4
5 test3 = [*test1, *test2]
6 print(test3)
```

[1, 2, 3, 4, 5, 6]

## 23 ) sort words in alphabetical order

In [ ]:

```
1 #25
2 str = input("Enter a string: ")
3
4 words = [word.lower() for word in str.split()]
5 words.sort()
6
7 print("The sorted words are:")
8 for word in words:
9     print(word)
```

Enter a string: Hello who is this typing for the lab file

The sorted words are:

file

for

hello

is lab

the

this

typing

who

## 24 ) Linear Search

```
1 def Linear(array, n, x):
2     for i in range(0, n):
3         if (array[i] == x):
4             return i
5     return -1
6
7
8 array = [2, 4, 0, 1, 9]
9 x = 1
10 n = len(array)
11 result = Linear(array, n, x)
12 if(result == -1):
13     print("Element not found")
14 else:
15     print("Element found at index: ", result)
```

Element found at index: 3

## 25 Linear search - sorted

```
In [ ]: 1 def LinearOr(array, n, x):
2         for i in range(0, n):
3             if (array[i] == x):
4                 return i
5         return -1
6
7
8 array = [2, 4, 0, 1, 9]
9 array.sort()
10 print(array)
11 x = 1
12 n = len(array)
13 result = LinearOr(array, n, x)
14 if(result == -1):
15     print("Element not found")
16 else:
17     print("Element found at index: ", result)
```

[0, 1, 2, 4, 9]

Element found at index: 1



In

## 26 ) Maintain practical note book as per their serial numbers - dict

```
In [13]: 1 n=int(input("Enter number of books: \n"))
2 d={}
3 for i in range(1,n+1):
4     key=i
5     value=input("Enter Name :")
6     d.update({key: value})
7
8 print(d)
```

Enter number of books:

5

Enter Name :Deep Learning with CUDA

Enter Name :Spacy

Enter Name :AutoNLP

Enter Name :CS Manual

Enter Name :The AWK and Sed

{1: 'Deep Learning with CUDA', 2: 'Spacy', 3: 'AutoNLP', 4: 'CS Manual', 5: 'The A WK and Sed'}

## 27 ) Dictionary

1) Insert 2) delete 3) change

```
In [ ]: 1 dict1={'Name':'Me', "Age": 1000, "Country": "India"}
2 print(dict1)
3
4 #insert
5 dict1['Gender']='Male'
6 print(dict1)
7 dict1['Initial']='A'
8 print(dict1)
9
10 # delete item
11 dict1.pop('Country')
12 print(dict1)
```

{'Name': 'Me', 'Age': 1000, 'Country': 'India'}

{'Name': 'Me', 'Age': 1000, 'Country': 'India', 'Gender': 'Male'}

{'Name': 'Me', 'Age': 1000, 'Country': 'India', 'Gender': 'Male', 'Initial': 'A'}

{'Name': 'Me', 'Age': 1000, 'Gender': 'Male', 'Initial': 'A'}

```
In [ ]: 1 #delete entire
2 del dict1
```

```
In [ ]: 1 dict1# It will give error
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-57-e36219336d90> in <module>()
----> 1 dict1
```

NameError: name 'dict1' is not defined

## 28) Number is in a given range using functions.

```
1 def count(list1, l, r):
2     c = 0
3     for x in list1:
4         if x >= l and x <= r:
5             c += 1
6     return c
7
8 list1 = [1,2,3,4,5,6,7,8]
9 l = 4
10 r = 8
11 print(count(list1, l, r))
```

5

## 29 ) Function that accepts a string and calculates upper and lower case

```
In [ ]: 1 str1=input('Please enter the sentence : ')
2 upper=lower=0
3 for i in range(len(str1)):
4     if(str1[i].isupper()):
5         upper+=1
6     elif(str1[i].islower()):
7         lower+=1
8 print(f"Upper :\t{upper} \nLower :\t{lower}")
```

```
Please enter the sentence : Hello World
Upper : 2
Lower : 8
```

## 30 ) Max of three

```
In [ ]: 1 a=(input("Enter List Elements:")).split()
2 l=list(map(int,a))
```

```
Enter List Elements:2 5 6
```

```
In [ ]: 1 print(max(l))
```

6

In

## 31 ) Multiply all the numbers in a list using functions.

```
In [ ]: 1 total = 1
2 A = [1,10,10,10,10]
3
4 for i in range(0, len(A)):
5     total = total*A[i]
6
7 print("Output: ", total)
```

Output: 10000

## 32 ) Fibonacci sequence - 33 Factorial

- using recursion.

```
In [ ]: 1 # Fibonacci
2 def fib(n): #Recursive
3     if n <= 0:
4         return 0
5     if n == 1:
6         return 1
7     return fib(n-1) + fib(n-2)
8
9 # Factorial
10 def factorial(n):
11     if n <= 1:
12         return 1
13     return n*factorial(n-1)
14
15 num = int(input("Enter a number: "))
16 print("{}th term of fibonacci series: ".format(num), fib(num))
17 print("Factorial of {}: ".format(num), factorial(num))
```

Enter a number: 4

4th term of fibonacci series: 3

Factorial of 4: 24

## 34 ) create a module of factorial in Python.

```
In [ ]: 1 #36
2 #Importing factorial function created previously
3 from facto import factorial
4
5 print(factorial(4))
```

24

```
Out[45]: '\nfacto.py\n\ndef factorial(n):\n    if n <= 1:\n        return 1\n    return n*f\n    actorial(n-1)\n\nnum = int(input("Enter a number: "))\nprint("{}th term of\nfibona cci series: ".format(num), fib(num))\nprint("Factorial of {}:\n".format(num), facto rial(num))'
```

**35 ) class named Rectangle, constructed by a length& width, also design a method which will compute the area of a rectangle.**

```
1  # 35
2  class Rectangle(Shape):
3      length = 0.0
4      breadth = 0.0
5
6      def __init__(self, l, br, co):
7          self.length = l
8          self.breadth = br
9
10     def Calc_Area(self):
11         Area = (self.length)*(self.breadth)
12         return Area
13
14
15 a = eval(input(" Enter the Length of the Rectangle : "))
16 b = eval(input(" Enter the Breadth of the Rectangle : "))
17
18 ob1=Rectangle(a,b,c)
19 print("Area of the Rectangle : ", ob1.Calc_Area())
```

Enter the Length of the Rectangle : 6  
Enter the Breadth of the Rectangle : 7  
Area of the Rectangle : 42

**36 ) Class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.**

```
In [ ]: 1  class Circle:
2      radius=0.0
3      def __init__(self,r):
4          self.radius = r
5
6      def Get_Radius(self):
7          x = self.radius
8          return x
9
10     def Calc_Area(self):
11         Area = (3.14)*(self.radius)*(self.radius)
12         return Area
13
14     def Calc_Peri(self):
15         Peri = (2)*(3.14)*(self.radius)
16         return Peri
17
18 a = eval(input('Enter Radius of your Choice : '))
19 ob1 = Circle(a)
20 print("The Area of the Circle Is : ", ob1.Calc_Area())
21 print("The Area of the Circle Is : ", ob1.Calc_Peri())
```

Enter Radius of your Choice : 3

In

```
The Area of the Circle Is : 28.259999999999998
The Area of the Circle Is : 18.84
```

## 37 ) Class to reverse a string word by word

```
In [21]: 1 class reverser():
2         def reverse_string(self,my_string):
3             if len(my_string) == 0:
4                 return my_string
5             else:
6                 return self.reverse_string(my_string[1:]) + my_string[0]
7
8 obj = reverser()
9 my_str = str(input(">>>> : "))
10 print("The string is :", my_str)
11 print("The reversed string is :")
12 print(obj.reverse_string(my_str))
```

```
>>>> : wow it worked
The string is : wow it worked
The reversed string is : dekrow
ti wow
```

## 38 ) Read a file

```
In [ ]: 1 x = input("Enter file name : ")
2 f = open(x,"w")
3 f.write("HelloWorld")
4 f = open(x,"r")
5 for i in f:
6     print(i[ : :-1])
```

```
Enter file name : h
dlroWolleH
```

## 39 ) Read first n lines of a text file.

```
In [10]: 1 f = open("h.txt","w")
2 f.write("\n It's me\n")
3 f.write(" Yup me\n")
4 f.write(" OK\n")
5 f.write(" Bye\n")
6 f.close()
7
8 N = int(input("Enter number of lines : "))
9 fptr = open("h.txt", "r")
10 line = (fptr.readlines())[0:N]
11 print(line)
```

```
Enter number of lines : 3
['\n', " It's me\n", ' Yup me\n']
```