



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

IT WORKSHOP ON PYTHON

LAB MANUAL

COURSE CODE: BCS23204P

Prepared by:-

RUBI KALITA

Laboratory Instructor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GUWAHATI-781017



Hathkhowapara, Azara, Guwahati 781017, Assam

	L	T	P	C
	0	0	2	1
Pre-requisite: Basic knowledge of Programming				
Course Outcome:				
After successful completion of the course, the students will be able to				
CO1: Write, Test and Debug Python Programs.				
CO2: Implement Conditionals and Loops for Python Programs				
CO3: Use functions and represent Compound data using Lists, Tuples and Dictionaries.				
CO4: Read and write data from & to files in Python and develop Application				
Suggested Practical List	15 hours			
WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:				
Grade A : if Percentage ≥ 80				
Grade B : if Percentage ≥ 60 and Percentage < 80				
Grade C : if Percentage ≥ 40 and Percentage < 60				
Grade D : if Percentage ≤ 40				
2. WAP to print factors of a given number.				
3. WAP to add N natural numbers and display their sum.				
4. WAP to print the following conversion table (use looping construct):				
Height (in Feet)	Height (in inches)			
5.0ft	60 inches			
5.1ft	61.2 inches			
5.8ft	69.6 inches			
5.9ft	70.8 inches			
6.0ft	72 inches			
5. WAP that takes a positive integer n and the produces lines of output as shown:				
* * * * * *				



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.
7. Write a function that calculates factorial of a number.
8. WAP to print the series and its sum: (use function)
 $1/1! + 1/2! + 1/3! + \dots + 1/n!$
9. WAP to perform the following operations on an input string
 - a. Print length of the string
 - b. Find frequency of a character in the string
 - c. Print whether characters are in uppercase or lowercase
10. WAP to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.
11. WAP to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.
12. WAP to create a tuple $t1 = (1, 2, 5, 7, 2, 4)$. The program should perform the following:
 - a. Print tuple in two lines, line 1 containing the first half of the tuple and second line having the second half.
 - b. Concatenate tuple $t2 = (10, 11)$ with $t1$.



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

1. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:

Grade A : if Percentage ≥ 80
Grade B : if Percentage ≥ 60 and Percentage < 80
Grade C : if Percentage ≥ 40 and Percentage < 60
Grade D : if Percentage ≤ 40

Code:

```
m1 = float(input("Enter marks of subject 1: "))
m2 = float(input("Enter marks of subject 2: "))
m3 = float(input("Enter marks of subject 3: "))

total = m1 + m2 + m3
percentage = total / 3

print(f"Total Marks = {total}")
print(f"Percentage = {percentage:.2f}")

if percentage >= 80:
    print("Grade: A")
elif percentage >= 60:
    print("Grade: B")
elif percentage >= 40:
    print("Grade: C")
else:
    print("Grade: D")
```

Output:

```
Enter marks of subject 1: 85
Enter marks of subject 2: 90
Enter marks of subject 3: 80
Total Marks = 255.0
Percentage = 85.00
Grade: A
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

2. WAP to print factors of a given number.

Code:

```
# Program 2: Print factors of a given number
```

```
n = int(input("Enter a number: "))
print(f"Factors of {n} are:")
for i in range(1, n + 1):
    if n % i == 0:
        print(i, end=" ")
```

Output:

```
Enter a number: 12
Factors of 12 are:
1 2 3 4 6 12
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

3. WAP to add N natural numbers and display their sum.

Code:

Program 3: Sum of N natural numbers

```
n = int(input("Enter the value of N: "))  
sum_n = 0
```

```
for i in range(1, n + 1):  
    sum_n += i
```

```
print(f"Sum of first {n} natural numbers = {sum_n}")
```

Output:

```
Enter the value of N: 10  
Sum of first 10 natural numbers = 55
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

4. WAP to print the following conversion table (use looping constructs):

Height(in Feet)	Height(in inches_)
5.0ft	60 inches
5.1ft	61.2inches
5.8ft	69.6inches
5.9ft	70.8inches
6.0ft	72inches

Code:

Program 4: Print conversion table (Feet to Inches)

```
print("Height Conversion Table (Feet to Inches)")
print("Feet\tInches")
```

```
ft = 5.0
while ft <= 6.0:
    inches = ft * 12
    print(f"{ft:.1f}\t{inches:.1f}")
    ft += 0.1
```

Output:

Height Conversion Table (Feet to Inches)

Feet	Inches
5.0	60.0
5.1	61.2
5.2	62.4
5.3	63.6
...	
5.9	70.8
6.0	72.0



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

5. WAP that takes a positive integer and produces lines of output as shown:

```
*  
* *  
* * *
```

Code:

Program 5: Print star pattern

```
n = int(input("Enter number of lines: "))
```

```
for i in range(1, n + 1):  
    print("* " * i)
```

Output:

Enter number of lines: 3

```
*  
* *  
* * *
```




GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

6. Write a menu driven program using user defined function to print the area of rectangle, square, circle and triangle by accepting suitable input from user.

Code:

Program 6: Menu driven program using functions

```
import math
```

```
def area_rectangle(l, b):  
    return l * b
```

```
def area_square(s):  
    return s * s
```

```
def area_circle(r):  
    return math.pi * r * r
```

```
def area_triangle(b, h):  
    return 0.5 * b * h
```

```
while True:  
    print("\n--- AREA CALCULATOR ---")  
    print("1. Rectangle")  
    print("2. Square")  
    print("3. Circle")  
    print("4. Triangle")  
    print("5. Exit")
```

```
    choice = int(input("Enter your choice: "))
```

```
    if choice == 1:  
        l = float(input("Enter length: "))  
        b = float(input("Enter breadth: "))  
        print("Area of Rectangle =", area_rectangle(l, b))
```

```
    elif choice == 2:
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
s = float(input("Enter side: "))
print("Area of Square =", area_square(s))

elif choice == 3:
    r = float(input("Enter radius: "))
    print("Area of Circle =", area_circle(r))

elif choice == 4:
    b = float(input("Enter base: "))
    h = float(input("Enter height: "))

print("Area of Triangle =", area_triangle(b, h))

elif choice == 5:
    print("Exiting program...")
    break

else:
    print("Invalid choice! Try again.")
```

Output:

```
--- AREA CALCULATOR ---
1. Rectangle
2. Square
3. Circle
4. Triangle
5. Exit
Enter your choice: 3
Enter radius: 7
Area of Circle = 153.93804002589985
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

7. Write a function that calculates factorial of a number n.

Code:

Program 7: Function to calculate factorial

```
def factorial(n):
```

```
    fact = 1
```

```
    for i in range(1, n + 1):
```

```
        fact *= i
```

```
    return fact
```

```
n = int(input("Enter a number: "))
```

```
print(f"Factorial of {n} = {factorial(n)}")
```

Output:

Enter a number: 5

Factorial of 5 = 120



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

8. WAP to print the series and its sum:(use functions)

$1/1! + 1/2! + 1/3! + \dots + 1/n!$

Code:

Program 8: Series $1/1! + 1/2! + 1/3! + \dots + 1/n!$

def factorial(n):

fact = 1

for i in range(1, n + 1):

fact *= i

return fact

n = int(input("Enter the value of n: "))

sum_series = 0

print("Series: ", end="")

for i in range(1, n + 1):

print(f"1/{i}!", end=" ")

if i != n:

print("+", end=" ")

sum_series += 1 / factorial(i)

print(f"\nSum of series = {sum_series:.4f}")

Output:

Enter the value of n: 4

Series: $1/1! + 1/2! + 1/3! + 1/4!$

Sum of series = 1.7083



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

9. WAP to perform the following operations on an input string

- a. Print length of the string
- b. Find frequency of a character in the string
- c. Print whether characters are in uppercase or lowercase

Code:

Program 9: String operations

```
string = input("Enter a string: ")
```

```
print("Length of string =", len(string))
```

```
ch = input("Enter a character to find frequency: ")
```

```
count = string.count(ch)
```

```
print(f"Frequency of '{ch}' = {count}")
```

```
for c in string:
```

```
    if c.isupper():
```

```
        print(f"{c} is uppercase")
```

```
    elif c.islower():
```

```
        print(f"{c} is lowercase")
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Output:

Enter a string: PyThon

Length of string = 6

Enter a character to find frequency: o

Frequency of 'o' = 1

P is uppercase

y is lowercase

T is uppercase

h is lowercase

o is lowercase

n is lowercase



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati781017, Assam

AIM 10,11 & 12: IMPLEMENTATION OF STRING, LIST, TUPLE, ARRAY AND DICTIONARY

Strings in Python:

A string is a sequence of characters that can be a combination of letters, numbers, and special characters. It can be declared in python by using single quotes, double quotes, or even triple quotes. These quotes are not a part of a string, they define only starting and ending of the string. Strings are immutable, i.e., they cannot be changed. Each element of the string can be accessed using indexing or slicing operations.

Assigning string to a variable

```
a = 'This is a string'
```

```
print (a)
```

```
b = "This is a string"
```

```
print (b)
```

```
c= '''This is a string'''
```

```
print (c)
```

Output:

```
This is a string
```

```
This is a string
```

```
This is a string
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Lists in Python:

Lists are one of the most powerful data structures in python. Lists are sequenced data types. In Python, an empty list is created using list() function. They are just like the arrays declared in other languages. But the most powerful thing is that list need not be always homogeneous. A single list can contain strings, integers, as well as other objects. Lists can also be used for implementing stacks and queues. Lists are mutable, i.e., they can be altered once declared. The elements of list can be accessed using indexing and slicing operations.

Declaring a list

```
L = [1, "a" , "string" , 1+2]
```

```
print L
```

#Adding an element in the list

```
L.append(6)
```

```
print L
```

#Deleting last element from a list

```
L.pop()
```

```
print L
```

#Displaying Second element of the list



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
print L[1]
```

The output is:

```
[1, 'a', 'string', 3]
```

```
[1, 'a', 'string', 3, 6]
```

```
[1, 'a', 'string', 3]
```

```
a
```

Tuples in Python: A tuple is a sequence of immutable Python objects. Tuples are just like lists with the exception that tuples cannot be changed once declared. Tuples are usually faster than lists.

```
tup = (1, "a", "string", 1+2)  
print(tup)  
print(tup[1])
```

The output is :

```
(1, 'a', 'string', 3)
```

```
a
```

Dictionary in Python is a collection of keys values, used to store data values like a map, which, unlike other data types which hold only a single value as an element.

```
# Creating a Dictionary  
# with Integer Keys  
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}  
print("\nDictionary with the use of Integer Keys: ")  
print(Dict)
```

```
# Creating a Dictionary  
# with Mixed keys
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
Dict = {'Name': 'Geeks', 1: [1, 2, 3, 4]}
print("\nDictionary with the use of Mixed Keys: ")
print(Dict)
```

Output:

Dictionary with the use of Integer Keys:

```
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
```

Dictionary with the use of Mixed Keys:

```
{'Name': 'Geeks', 1: [1, 2, 3, 4]}
```

Array in Python An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).

Python program to demonstrate

Creation of Array

```
# importing "array" for array creations
import array as arr
```

creating an array with integer type

```
a = arr.array('i', [1, 2, 3])
```

printing original array

```
print ("The new created array is : ", end = " ")
```

```
for i in range (0, 3):
```

```
    print (a[i], end = " ")
```

```
print()
```

creating an array with double type

```
b = arr.array('d', [2.5, 3.2, 3.3])
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
# printing original array
print ("The new created array is : ", end = " ")
for i in range (0, 3):
    print (b[i], end = " ")
```

output:

The new created array is : 1 2 3

The new created array is : 2.5 3.2 3.3



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

OUT OF SYLLEBUS TOPICS

AIM : CREATION OF CLASS

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state..

Syntax: Class Definition

class ClassName:

Statement

Syntax: Object Definition

obj = ClassName()

print(obj.attr)

Python3 program to

demonstrate instantiating

a class



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

class Dog:

A simple class

attribute

attr1 = "mammal"

attr2 = "dog"

A sample method

def fun(self):

print("I'm a", self.attr1)

print("I'm a", self.attr2)

Driver code

Object instantiation

Rodger = Dog()



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Accessing class attributes

and method through objects

print(Rodger.attr1)

Rodger.fun()

Output

mammal

I'm a mammal

I'm a dog



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

AIM : IMPLEMENTATION OF PYTHON LIBRARIES SUCH AS NUMPY, SCIPY, DOCTEST, OS, TKINTER

NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

#Importing numpy

```
import numpy as np
```

```
x = np.array([4, 5])
```

```
y = np.array([7, 10])
```

```
print("Original vectors:")
```

```
print(x)
```

```
print(y)
```

```
print("Inner product of vectors:")
```

#Inner product (dot product)of arrays

```
print(np.dot(x, y))
```

OUTPUT:

Original vectors

[4,5]

[7,10]

Inner product of vectors:

78



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

SciPy

SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT (fast Fourier transform), signal and image processing, ODE (Ordinary differential equation) solvers and other tasks common in science and engineering.

#Importing scipy library

```
from scipy import linalg
```

```
import numpy as np
```

#define square matrix

```
two_d_array = np.array([ [4,5], [3,2] ])
```

#pass values to det() function

#Calculating determination of two matrix

```
linalg.det( two_d_array )
```

OUTPUT:

-7.0

OS

The os Python module provides a big range of useful methods to manipulate files and directories, basically an interaction with operating system. This module provides a portable way of using operating system dependent functionality.

This module has number of useful functions /methods to perform the operations.

#Importing os module

#name the current working directory



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
import os
```

```
print(os.getcwd())
```

```
#Name the operating system
```

```
import os
```

```
print(os.name)
```

```
c:\users\apexnt
```

Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and Python with tkinter is the fastest and easiest way to create the GUI applications.

```
#import tkinter
```

```
import tkinter as tk
```

```
r = tk.Tk() .
```

```
#title of the window
```

```
r.title('Demo')
```

```
#To stop the window
```

```
r.mainloop()
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Output:

Initial queue

['a', 'b', 'c']

Elements dequeued from queue

a

b

c

Queue after removing elements

[]
