**CM-507**

**PYTHON PROGRAMMING LAB**

1. Write and execute simple python Program.

n = int(input("Enter a positive integer: "))

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

print(i)

1. Write /execute simple ‘Python’ program: Develop minimum 2 programs using different data types (numbers, string, tuple, list, and dictionary).

**Program1: Using numbers**

a = 5

b = 3

c = a + b

print("Sum of", a, "and", b, "is", c)

d = a - b

print("Difference of", a, "and", b, "is", d)

e = a \* b

print("Product of", a, "and", b, "is", e)

f = a / b

print("Quotient of", a, "and", b, "is", f)p

**Program2: Using strings**

fruits = ['apple', 'banana', 'cherry']

greeting = "Hello, what fruit would you like to have?"

print(greeting)

for fruit in fruits:

print(fruit)

selected\_fruit = input("Enter the name of the fruit you would like to have: ")

if selected\_fruit in fruits:

print("You have selected", selected\_fruit)

else:

print("Sorry, we don't have", selected\_fruit)

1. Write /execute simple ‘Python’ program: Develop minimum 2 programs using Arithmetic Operators, exhibiting data type conversion.

**Program1: Using Arithmetic Operators**

a = 5

b = 3

c = a + b

print("Sum of", a, "and", b, "is", c)

d = a - b

print("Difference of", a, "and", b, "is", d)

e = a \* b

print("Product of", a, "and", b, "is", e)

f = a / b

print("Quotient of", a, "and", b, "is", f)

g = int(a / b)

print("Integer Quotient of", a, "and", b, "is", g)

**Program2: Using Data type conversion**

# Converting an integer to a string

integer = 42

string = str(integer)

print("String:", string)

# Converting a string to an integer

string = "42"

integer = int(string)

print("Integer:", integer)

# Converting a float to a string

float\_num = 3.14

string = str(float\_num)

print("String:", string)

# Converting a string to a float

string = "3.14"

float\_num = float(string)

print("Float:", float\_num)

# Converting a list to a tuple

lst = [1, 2, 3, 4]

tuple = tuple(lst)

print("Tuple:", tuple)

# Converting a tuple to a list

tuple = (1, 2, 3, 4)

lst = list(tuple)

print("List:", lst)

1. (i)Write simple programs to convert U.S. dollars to Indian rupees.

Program:

usd = **float**(**input**("Enter currency in USD: "))

inr = usd \* 73

print("The currency in INR is",**round**(inr,2))

(ii) Write simple programs to convert bits to Megabytes, Gigabytes and Terabytes.

# Python implementation of above program

# Function to calculates the bits

def Bits(kilobytes) :

# calculates Bits

# 1 kilobytes(s) = 8192 bits

Bits = kilobytes \* 8192

return Bits

# Function to calculates the bytes

def Bytes(kilobytes) :

# calculates Bytes

# 1 KB = 1024 bytes

Bytes = kilobytes \* 1024

return Bytes

# Driver code

if \_\_name\_\_ == "\_\_main\_\_" :

kilobytes = 1

print(kilobytes, "Kilobytes =",

Bytes(kilobytes) , "Bytes and",

Bits(kilobytes), "Bits")

1. Write simple programs to calculate the area and perimeter of the square, and the volume & perimeter of the cone.

import math

pi = math.pi

# Function to calculate Volume of Cone

def volume(r, h):

return (1 / 3) \* pi \* r \* r \* h

# Function To Calculate Surface Area of Cone

def surfacearea(r, s):

return pi \* r \* s + pi \* r \* r

# Driver Code

radius = float(5)

height = float(12)

slat\_height = float(13)

print( "Volume Of Cone : ", volume(radius, height) )

print( "Surface Area Of Cone : ", surfacearea(radius, slat\_height) )

1. Write program to: (i) determine whether a given number is odd or even. (ii) Find the greatest of the three numbers using conditional operators.

def odd\_even(num):

if num % 2 == 0:

return "Even"

else:

return "Odd"

def find\_greatest(a, b, c):

greatest = a

if b > greatest:

greatest = b

if c > greatest:

greatest = c

return greatest

num = int(input("Enter a number: "))

print("The number is", odd\_even(num))

a = int(input("Enter first number: "))

b = int(input("Enter second number: "))

c = int(input("Enter third number: "))

print("The greatest number is", find\_greatest(a, b, c))

1. Write a program to: i) Find factorial of a given number. ii) Generate multiplication table up to 10 for numbers 1 to 5.

def factorial(num):

if num == 0:

return 1

return num \* factorial(num - 1)

def multiplication\_table(n):

for i in range(1, 11):

print(n, "x", i, "=", n \* i)

num = int(input("Enter a number to find the factorial: "))

print("The factorial of", num, "is", factorial(num))

for i in range(1, 6):

print("\nMultiplication table for", i)

multiplication\_table(i)

1. Write a program to: i) Find factorial of a given number. ii) Generate multiplication table up to 10 for numbers 1 to 5 using functions.

def factorial(num):

if num == 0:

return 1

return num \* factorial(num - 1)

def multiplication\_table(n):

for i in range(1, 11):

print(n, "x", i, "=", n\*i)

num = int(input("Enter a number to find the factorial: "))

print("The factorial of", num, "is", factorial(num))

print("\nMultiplication table:")

for i in range(1, 6):

print("Table for", i)

multiplication\_table(i)

print("\n")

1. Write a program to: i) Find factorial of a given number using recursion. ii) Generate Fibonacci sequence up to 100 using recursion.

def factorial(num):

if num == 0:

return 1

return num \* factorial(num - 1)

def fibonacci(num):

if num <= 1:

return num

return fibonacci(num - 1) + fibonacci(num - 2)

num = int(input("Enter a number to find the factorial: "))

print("The factorial of", num, "is", factorial(num))

print("\nFibonacci sequence:")

for i in range(100):

fib = fibonacci(i)

if fib > 100:

break

print(fib, end=", ")

1. Write a program to: Create a list, add element to list, delete element from the lists.

my\_list = []

def add\_to\_list(element):

my\_list.append(element)

print(element, "added to the list")

def delete\_from\_list(element):

if element in my\_list:

my\_list.remove(element)

print(element, "deleted from the list")

else:

print(element, "not found in the list")

print("1. Add element to the list")

print("2. Delete element from the list")

print("3. Display the list")

print("4. Quit")

while True:

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

if choice == 1:

element = int(input("Enter an integer to add: "))

add\_to\_list(element)

elif choice == 2:

element = int(input("Enter an integer to delete: "))

delete\_from\_list(element)

elif choice == 3:

print("The list contains:", my\_list)

elif choice == 4:

print("Exiting the program...")

break

else:

print("Invalid choice. Try again.")

1. Write a program to: Sort the list, reverse the list and counting elements in a list.

my\_list = [3, 4, 1, 7, 5, 9, 2, 8, 6, 0]

def sort\_list(lst):

return sorted(lst)

def reverse\_list(lst):

return lst[::-1]

def count\_elements(lst):

count = {}

for element in lst:

count[element] = count.get(element, 0) + 1

return count

print("Original list:", my\_list)

sorted\_list = sort\_list(my\_list)

print("Sorted list:", sorted\_list)

reversed\_list = reverse\_list(my\_list)

print("Reversed list:", reversed\_list)

count = count\_elements(my\_list)

print("Count of elements in the list:", count)

1. Write a program to: Create dictionary, add element to dictionary, delete element from the dictionary.

my\_dict = {'a': 1, 'b': 2, 'c': 3}

def add\_element(d, key, value):

d[key] = value

return d

def delete\_element(d, key):

if key in d:

del d[key]

return d

print("Original dictionary:", my\_dict)

my\_dict = add\_element(my\_dict, 'd', 4)

print("Dictionary after adding an element:", my\_dict)

my\_dict = delete\_element(my\_dict, 'a')

print("Dictionary after deleting an element:", my\_dict)

1. Write a program to: To calculate average, mean, median, and standard deviation of numbers in a list.

import statistics

numbers = [3, 4, 1, 7, 5, 9, 2, 8, 6, 0]

def average(lst):

return sum(lst) / len(lst)

def mean(lst):

return statistics.mean(lst)

def median(lst):

return statistics.median(lst)

def standard\_deviation(lst):

return statistics.stdev(lst)

print("Numbers:", numbers)

avg = average(numbers)

print("Average:", avg)

mn = mean(numbers)

print("Mean:", mn)

md = median(numbers)

print("Median:", md)

stddev = standard\_deviation(numbers)

print("Standard Deviation:", stddev)

1. Write a program to: To print Factors of a given Number.

def factors(n):

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

if n % i == 0:

print(i)

num = int(input("Enter a number: "))

print("Factors of", num, "are:")

factors(num)

1. File Input/output: Write a program to: i) To create simple file and write “Hello World” in it. ii) To open a file in write mode and append Hello world at the end of a file.

# Step 1: Create a file and write "Hello World" in it

with open("hello.txt", "w") as file:

file.write("Hello World\n")

print("File 'hello.txt' created and written successfully.")

# Step 2: Open the file in append mode and append "Hello World" at the end

with open("hello.txt", "a") as file:

file.write("Hello World\n")

print("File 'hello.txt' appended successfully.")

1. Write a program to :i) To open a file in read mode and write its contents to another file but replace every occurrence of character ‘h’ ii) To open a file in read mode and print the number of occurrences of a character ‘a’.

# Step 1: Open a file in read mode and write its contents to another file with 'h' replaced

with open("input.txt", "r") as file1, open("output.txt", "w") as file2:

for line in file1:

modified\_line = line.replace('h', 'X') # Replace 'h' with 'X'

file2.write(modified\_line)

print("Contents of 'input.txt' copied to 'output.txt' with 'h' replaced.")

# Step 2: Open a file in read mode and count the number of occurrences of 'a'

char\_to\_count = 'a'

count = 0

with open("input.txt", "r") as file:

for line in file:

count += line.count(char\_to\_count) # Count occurrences of 'a' in each line

print(f"Number of occurrences of '{char\_to\_count}' in 'input.txt': {count}")

1. Write a Program to: Add two complex number using classes and objects.

class ComplexNumber:

def \_\_init\_\_(self, real, imag):

self.real = real

self.imag = imag

def \_\_add\_\_(self, other):

real\_sum = self.real + other.real

imag\_sum = self.imag + other.imag

return ComplexNumber(real\_sum, imag\_sum)

def display(self):

print(f"Sum: {self.real} + {self.imag}i")

# Input complex numbers

real1 = float(input("Enter real part of the first complex number: "))

imag1 = float(input("Enter imaginary part of the first complex number: "))

real2 = float(input("Enter real part of the second complex number: "))

imag2 = float(input("Enter imaginary part of the second complex number: "))

# Create ComplexNumber objects

complex1 = ComplexNumber(1, 1)

complex2 = ComplexNumber(2, 2)

# Add two complex numbers

result = complex1 + complex2

# Display the result

result.display()

18.Write a Program to: Subtract two complex number using classes and objects.

class ComplexNumber:

def \_\_init\_\_(self, real, imaginary):

self.real = real

self.imaginary = imaginary

def \_\_sub\_\_(self, other):

return ComplexNumber(self.real - other.real, self.imaginary - other.imaginary)

def \_\_repr\_\_(self):

return f"({self.real}+{self.imaginary}j)"

def main():

c1 = ComplexNumber(1, 2)

c2 = ComplexNumber(3, 4)

c3 = c1 - c2

print(c3)

if \_\_name\_\_ == "\_\_main\_\_":

main()

19.Write a Program to: Create a package and accessing a package.

# Create a package

package = \_\_import\_\_("my\_package")

# Import a module from the package

module = package.my\_module

# Call a function from the module

module.my\_function()

Procedure:

1. Launch the IDLE application.

2. Open a new file or an existing one, then write your Python code in the editor.

3. Save your file with a `.py` extension.

4. Press F5 or go to the "Run" menu and select "Run Module."

5. Your program's output will appear in the IDLE Shell window.