1) Installing python and setting up environment. Simple statements like printing the names (“Hello World”), numbers, mathematical calculations, etc.

print("Hello World")

print("Welcome to Learn Python Programming")

print(10+20)#addition

print(20-10)#substraction

print(5\*3)#multiplication

print(10/5)#division

print("Arithmatic operator")

a=int(input("Enter 1st number"))

b=int(input("Enter 2nd number"))

print("Addition is=",a+b)

print("Substraction is=",a-b)

print("Multiplication is=",a\*b)

print("Modulus is=",a%b)

print("Divison is=",a/b)

print("Power is=",a\*\*b)

print("comparison operator")

a=10

b=20

print("a>b",a>b)

print("a<b",a<b)

print("a<=b",a<=b)

print("a>=b",a>=b)

print("a==b",a==b)

print('a != b =', a != b)

print("Assignment operator")

# assign 10 to a

a = 10

# assign 5 to b

b = 5

# assign the sum of a and b to a

a += b # a = a + b

print(a)

# Output: 15

print("Logical operator")

a = 5

b = 6

print((a > 2) and (b >= 6)) # True

print((a > 8) or (b >= 10)) #false

print("basics of logical opr")

# logical AND

print(True and True) # True

print(True and False) # False

# logical OR

print(True or False) # True

# logical NOT

print(not True) # False

print("Bitwise operator")

#remember this formula 8 4 2 1

a=10 #1 0 1 0

Naina sonar

b=7 #0 1 1 1

print(a&b) #0 0 1 0=2

print(a|b) #1 1 1 1=15

print(a^b)

#Output:

Hello World

Welcome to Learn Python Programming

30

10

15

2.0

\*\*\*\*Arithmatic operator\*\*\*\*

Enter 1st number5

Enter 2nd number8

Addition is= 13

Substraction is= -3

Multiplication is= 40

Modulus is= 5

Divison is= 0.625

Power is= 390625

\*\*\*\*comparison operator\*\*\*\*

a>b False

a<b True

a<=b True

a>=b False

a==b False

a != b = True

\*\*\*\*Assignment operator\*\*\*\*

15

\*\*\*\*Logical operator\*\*\*\*

True

False

basics of logical opr

True

False

True

False

\*\*\*\*Bitwise operator\*\*\*\*

2 15 13

**2 Write a program to find all prime numbers within a given range.**

lower = int(input("Give lower limit: "))

upper = int(input("Give upper limit: "))

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):

   if num > 1:

       for i in range(2, num):

           if (num % i) == 0:

               break

       else:

           print(num , end=" ")

**#Output**

"""Give lower limit: 5

Give upper limit: 50

Prime numbers between 5 and 50 are:

5 7 11 13 17 19 23 29 31 37 41 43 47"""

3 **Write a python program to print "n" terms of Fibonacci Series using Iteration**

nterms = int(input("How many terms? "))

n1, n2 = 0, 1

count = 0

if nterms <= 0:

   print("Please enter a positive integer")

elif nterms == 1:

   print("Fibonacci sequence upto",nterms,":")

   print(n1)

else:

   print("Fibonacci sequence:")

   while count < nterms:

       print(n1, end=" ")

       nth = n1 + n2

       n1 = n2

       n2 = nth

       count += 1

**#Output**

"""How many terms? 10

Fibonacci sequence:

0 1 1 2 3 5 8 13 21 34"""

**Write a python program to demonstrate the use of slicing in string.**

text = "Hello, Python World!"

substring1 = text[0:5]

print(f"Substring from index 0 to 4: '{substring1}'")

substring2 = text[:5]

print(f"Substring from start to index 4: '{substring2}'")

substring3 = text[7:]

print(f"Substring from index 7 to end: '{substring3}'")

substring4 = text[-6:]

print(f"Last 6 characters: '{substring4}'")

substring5 = text[0:13:2]

print(f"Every 2nd character from index 0 to 12: '{substring5}'")

reversed\_text = text[::-1]

print(f"Reversed string: '{reversed\_text}'")

substring6 = text[12:5:-2]

print(f"Every 2nd character in reverse order from index 12 to 6: '{substring6}'")

substring7 = text[-4:-1]

print(f"Substring from the second-to-last to the fourth-to-last character: '{substring7}'")

empty\_string = ""

empty\_slicing = empty\_string[1:5]

print(f"Slicing an empty string: '{empty\_slicing}'")

out\_of\_range\_slicing = text[100:200]

print(f"Slicing with out-of-range indices: '{out\_of\_range\_slicing}'")

**#Output:**

"""Substring from index 0 to 4: 'Hello'

Substring from start to index 4: 'Hello'

Substring from index 7 to end: 'Python World!'

Last 6 characters: 'World!'

Every 2nd character from index 0 to 12: 'Hlo yhn'

Reversed string: '!dlroW nohtyP ,olleH'

Every 2nd character in reverse order from index 12 to 6: 'nhy '

Substring from the second-to-last to the fourth-to-last character: 'rld'

Slicing an empty string: ''

Slicing with out-of-range indices: ''

"""

**Write a python Program related to Functions & Modules.**

# mymodule.py

def greet(name):

    return f"Hello, {name}!"

def add(x, y):

    return x + y

# main.py

import mymodule

print(mymodule.greet("Alice"))

print(“Addition is=”,mymodule.add(5, 7))

**#Output:**

Hello, Alice!

Addition is=12

**Write a python  program that demonstrate concept of Functional Programming.**

from functools import reduce

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

squared\_numbers = list(map(lambda x: x \* x, numbers))

print("Squared Numbers:", squared\_numbers)

even\_numbers = list(filter(lambda x: x % 2 == 0, numbers))

print("Even Numbers:", even\_numbers)

sum\_of\_numbers = reduce(lambda x, y: x + y, numbers)

print("Sum of Numbers:", sum\_of\_numbers)

**#Output:**

"""Squared Numbers: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

Even Numbers: [2, 4, 6, 8, 10]

Sum of Numbers: 55"""

**Write a program to demonstrate the use of list & related functions.**

fruits = ["apple", "orange", "cherry","apple"]

print("Initial List:", fruits)

fruits.append("date")

print("After Append:", fruits)

fruits.remove("orange")

print("After Remove:", fruits)

removed\_fruit = fruits.pop(2)

print("Removed Fruit:", removed\_fruit)

print("After Pop:", fruits)

slice\_of\_fruits = fruits[1:4]

print("Slice of Fruits:", slice\_of\_fruits)

fruits.sort()

print("After Sort:", fruits)

fruits.reverse()

print("After Reverse:", fruits)

length\_of\_list = len(fruits)

print("Length of List:", length\_of\_list)

**output:**

Initial List: ['apple', 'orange', 'cherry', 'apple']

After Append: ['apple', 'orange', 'cherry', 'apple', 'date']

After Remove: ['apple', 'cherry', 'apple', 'date']

Removed Fruit: apple

After Pop: ['apple', 'cherry', 'date']

Slice of Fruits: ['cherry', 'date']

After Sort: ['apple', 'cherry', 'date']

After Reverse: ['date', 'cherry', 'apple']

Length of List: 3

**Write a python program to demonstrate the use of Dictionary & related functions.**

person = {

    "name": "Alexa",

    "age": 30,

    "city": "New York"

}

print("Initial Dictionary:", person)

print(person["name"])

person["occupation"] = "Engineer"

print("After Adding Occupation:", person)

del person["city"]

print("After Removing City:", person)

occupation = person.pop("occupation", "Not Found")

print("Popped Occupation:", occupation)

print("After Popping Occupation:", person)

print(person.keys())

print(person.values())

print(person.items())

**Output**:

Initial Dictionary: {'name': 'Alexa', 'age': 30, 'city': 'New York'}

Alexa

After Adding Occupation: {'name': 'Alexa', 'age': 30, 'city': 'New York', 'occupation': 'Engineer'}

After Removing City: {'name': 'Alexa', 'age': 30, 'occupation': 'Engineer'}

Popped Occupation: Engineer

After Popping Occupation: {'name': 'Alexa', 'age': 30}

dict\_keys(['name', 'age'])

dict\_values(['Alexa', 30])

dict\_items([('name', 'Alexa'), ('age', 30)])

**Write a python  program to demonstrate the use of Tuple.**

country=("India","US","UK","Bangladesh","china")

print(country)

print("use of length function=",len(country))

print("print using index=",country[3])

print("print using slicing",country[0:4])

print("print the count of tuple elements",country.count("US"))

print("print Index No. of element",country.index("UK"))

print("print tuple Elements multiple times ",country\*2)

**Output:**

('India', 'US', 'UK', 'Bangladesh', 'china')

use of length function= 5

print using index= Bangladesh

print using slicing ('India', 'US', 'UK', 'Bangladesh')

print the count of tuple elements 1

print Index No. of element 2

print tuple Elements multiple times  ('India', 'US', 'UK', 'Bangladesh', 'china', 'India', 'US', 'UK', 'Bangladesh', 'china')

**Write a python program to demonstrate Regular Expression.**

import re

string = 'hello 12 hi 89. Howdy 34'

pattern = '\d+'

result = re.findall(pattern, string)

print("finadall",result)

result = re.split(pattern, string)

print("Split",result)

txt = "The rain in india"

x = re.search("india", txt)

print(x)

x = re.split('\s', txt)

print(x)

**Output:**

finadall ['12', '89', '34']

Split ['hello ', ' hi ', '. Howdy ', '']

<re.Match object; span=(12, 17), match='india'>

['The', 'rain', 'in', 'india']

**Write a python program to demonstrate the working of Class and Objects.**

class Person:

    def \_\_init\_\_(self, name, age):

        self.name = name

        self.age = age

    def display\_info(self):

        print("Name:",self.name)

        print("Age:",self.age)

    def celebrate\_birthday(self):

        self.age += 1

        print(f"Happy Birthday, {self.name}! You are now {self.age} years old.")

person1 = Person("Alexa", 30)

print("Initial Information:")

person1.display\_info()

print("\nCelebrating Birthday...")

person1.celebrate\_birthday()

print("\nUpdated Information:")

person1.display\_info()

**Output**

Initial Information:

Name: Alexa

Age: 30

Celebrating Birthday...

Happy Birthday, Alexa! You are now 31 years old.

Updated Information:

Name: Alexa

Age: 31

**Write a python program to demonstrate the working of Inheritance**

class Person:

    def \_\_init\_\_(self, name, age):

        self.name = name

        self.age = age

    def greet(self):

        return f"Hello, my name is {self.name} and I am {self.age} years old."

class Student(Person):

    def \_\_init\_\_(self, name, age, student\_id):

        super().\_\_init\_\_(name, age)

        self.student\_id = student\_id

    def student\_info(self):

        return f"{self.greet()} My student ID is {self.student\_id}."

student = Student("Alexa", 20, "S12345")

print(student.student\_info())

**Output**

Hello, my name is Alexa and I am 20 years old. My student ID is S12345.

**Write a python program to demonstrate the working of Overloading Methods**

class MathOperations:

    def add(self, a, b, c=0):

        """Adds two or three numbers."""

        return a + b + c

m=MathOperations()

print("Addition with 2 Arguments=",m.add(2,3))

print("Addition with 3 Arguments=",m.add(2,3,4))

**Output**

Addition with 2 Arguments= 5

Addition with 3 Arguments= 9

**Output**

Error: You cannot divide by zero.

Execution of divide\_numbers() completed.

None

**Write a python program to demonstrate Exception Handling Mechanism**

def divide\_numbers(num1, num2):

    """Divides two numbers and handles division by zero."""

    try:

        result = num1 / num2

    except ZeroDivisionError:

        print("Error: You cannot divide by zero.")

    except TypeError:

        print("Error: Please provide numbers (integers or floats) for division.")

    else:

        print("Division successful.")

        return result

    finally:

        print("Execution of divide\_numbers() completed.")

num1=10

num2=0

print(divide\_numbers(num1, num2))