**Program No.1**

**Que. PythonProgramtoAddtwonumbers.**

num1= 15

num2 = 12

sum = num1+ num2

print("the addition of two number is",sum)

**Output :**

Graphical user interface, text

Description automatically generated

**Program No.2**

**Que. Python Program Add two number provided by user input**

number1= input("First number: ")

number2 = input("\ nSecond number: ")

sum = float(number1) + float(number2)

print("The sum of {0} and {1} is {2}" .format(number1, number2, sum))

**Output :**

Graphical user interface

Description automatically generated

**Program No.3**

**Que. Python program to find the factorial of a number provided by theuser.**

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

factorial = 1

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:print("The factorial of 0 is 1")

else:

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

factorial = factorial\* i

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

**Output :**

Text

Description automatically generated

**Program No.4**

**Que. Python Program for simple interest**

P = 1000

R = 1

T = 2

SI = (P \* R \* T) / 100

print("simple interest is", SI)

**Output :**Graphical user interface, text

Description automatically generated

**Program No.5**

**Que. Python Program for simple interest**

def simple\_ interest(p,t,r):

print('The principal is', p)

print('The time period is', t)

print('The rate of interest is',r)

si = (p \* t \* r)/100

print('The Simple Interest is', si)

return sisimple\_ interest(3000, 7, 1)

**Output :**

Graphical user interface, text

Description automatically generated

**Program No.6**

**Que. Python program to check if a string is palindrome or not**

my\_ string=input("Enter string:")

if(my\_ string==my\_ string[::- 1]

):

print("The string is a palindrome")

else:print("The string isn't a palindrome")

my\_ string=input("Enter string:")

if(my\_ string==my\_ string[::- 1]

):

print("The string is a palindrome") eles:

print("The string isn't a palindrome")

**Output :**

Text

Description automatically generated

**Program No.7**

**Python program to check if a string is palindrome or not using reverse**

my\_ str = 'dad'

# reverse the string

rev\_ str = reversed(my\_ str)

# check if the string is equal to its reverse

if list(my\_ str) == list(rev\_ str):

print("The string is a palindrome.")

else:

print("The string is not a palindrome.")

**Output :**

Graphical user interface, text

Description automatically generated

**Program No.8**

**Que. Python Program to Revese word in given string in python**

txt = "Hello World"[::- 1]

print(txt)

**Output**

Graphical user interface, text

Description automatically generated

**Program No.9**

**Que. Python Program to Revese word in given string in python using function**

def my\_ function(x):

return x[::- 1]

mytxt = my\_ function("I wonder how this text looks like backwards")

print(mytxt)

**Output:** A picture containing text, screenshot, monitor

Description automatically generated

**Program No.10**

**Que. Python Program to Find out way to remove i’tfrom string**

test\_ str = "Computerscience”

print ("The original string is : " + test\_ str)

new\_ str = test\_ str[:2]+ test\_ str[3:]

print ("The string after removal of i'th character : " + new\_ str)

**Output :**

Text

Description automatically generated

**Program No.11**

**Que. Python program to find out way to remove l'th character from string.using loop .**

**Output :**

A picture containing text, screenshot, monitor

Description automatically generated

**Program No.12**

**Que. Python Program to check if substring is present in string python**

txt = "The rain in Spain stays mainly in the plain"x = "ain" in txt

print(x)

**Output:**

Graphical user interface, text

Description automatically generated

**Program No.13**

**Que. Python Program to interchange first and last element in list**

def swapList(newList):

size = len(newList)

temp = newList[0]

newList[0]= newList[size - 1]

newList[size - 1]= temp

return newList

newList = [12, 35, 9, 56, 24]print(swapList(newList))

**output:**

A screenshot of a computer

Description automatically generated

**Program No.14**

**Que. Python Program for insertion sort.**

def insertion\_ sort(list1):

for i in range(1, len(list1)):

value = list1[i]

j = i - 1

while j >= 0 and value < list1[j]

:

list1[j + 1]= list1[j]

j - = 1

list1[j + 1]

= value

return list1

list1= [10, 5, 13, 8, 2]

print("The unsorted list is:", list1)

print("The sorted list1is:", insertion\_ sort(list1))

**output :**

Text

Description automatically generated

**Program No.15**

**Que. Python Program for Linear search.**def linearsearch(arr, x):

for i in range(len(arr)):

if arr[i]== x:

return i

return - 1

arr = ['t','u','t','o','r','i','a','l']

x = 'a'

print("element found at index "+str(linearsearch(arr,x)))

**Output:**

A screenshot of a computer

Description automatically generated

**Program No.16**

**Que. python program to remove key from dictionar y**

thisdict ={

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict.pop("model")

print(thisdict)

**output :**

A picture containing text, screenshot, monitor, screen

Description automatically generated

**Program No.17**

**Que. python program to reversing a list**

apple = ['a','p','p','l','e']

# Method calling

apple.reverse() # Reverse elements of the list

# Displaying result

print(apple)

**output :**

A picture containing text, screenshot, monitor

Description automatically generated

**Program No.18**

**Que. Python program to swap two elements in a list**

def swapPositions(list, pos1, pos2):

list[pos1]

, list[pos2]= list[pos2]

, list[pos1]

return list# Driver function

List = [23, 65, 19, 90]

pos1, pos2 = 2, 1

print(swapPositions(List, pos1- 1, pos2- 1))

**output :**

A picture containing text, screenshot, monitor

Description automatically generated

**Program No.19**

**Que. Python program to swap two elements in a list**

def swapPositions(list, pos1, pos2):

first\_ ele = list.pop(pos1)

second\_ ele = list.pop(pos2- 1)

list.insert(pos1, second\_ ele)

list.insert(pos2, first\_ ele)return list

List = [23, 65, 19, 90]

pos1, pos2 = 1, 3

print(swapPositions(List, pos1- 1, pos2- 1))

**output :**

Text

Description automatically generated

**Program No.20**

**Que. Python program to find out different way to clear list in python**

GEEK = [6, 0, 4, 1]

print('GEEK before clear:', GEEK)

# Clearing list

GEEK.clear()print('GEEK after clear:', GEEK)

**output :**

A picture containing text, screenshot, monitor

Description automatically generated

**Program No.21**

**Que. Python program to find out different way to clear list in python**

list1= [1, 2, 3]

print ("List1before deleting is : " + str(list1))

# deleting list1using del

del list1[:]

print ("List1after clearing using del : " + str(list1))

**output :**

A picture containing text, screenshot, monitor, computer

Description automatically generated

**Program No.22**

**Que. Python program to demonstrate use dictionaries by key or value**

Dict = {}

print("Empty Dictionary: ")

print(Dict)

Dict = dict({1: 'Java', 2: 'T', 3:'Point'})

print("\ nCreate Dictionary by using dict(): ")

print(Dict)

Dict = dict([(1, 'Devansh'), (2, 'Sharma')]

)

print("\ nDictionary with each item as a pair: ")

print(Dict)

**output :**

A picture containing text, screenshot, monitor

Description automatically generated

Program No 23

Q. write a simple python function to check whether X is even or odd.

num=int(input("Enteranumber:"))if(num%2)==0:print("{0}isEvennumber".format(num))else:print("{0}isOddnumber".format(num))

Output: A picture containing text, screenshot, monitor

Description automatically generated

2.writeasimplepythonfunctiontodemonstratedefaultdefmy\_function(country="Canada"):print("Iamfrom"+country)my\_function("England")my\_function("India")my\_function()my\_function("Japan")

Output:

A picture containing text, screenshot, monitor

Description automatically generated3.writeasimplepythonmoduleforsubtractionandaddition.

Calc.pydefadd(x,y):return(x+y)defsubtract(x,y):return(x-y)main.pyfromcalcimportadd,subtracta=int(input("Enterthefirstnumber"))b=int(input("Enterthesecondnumber"))print("Summationis=",add(a,b))print("substractionis=",subtract(a,b))

Output:

Graphical user interface, text

Description automatically generated4.writeaprogramtoimportingsqrt()fromthemodulemath.

importmath

print(math.sqrt(12))print(math.sqrt(4))print(math.sqrt(50))

Output Graphical user interface, text

Description automatically generated

5.writeaprogramtoimportingFactorialfromthemodulemath.importmathx=5print("Thefactorialof5is:",end="")print(math.factorial(x))

Output A picture containing text, screenshot, monitor

Description automatically generated

6.Write a program to demonstrate to open and close file.

My\_file = open("my\_file.txt", "r") # Open a file

print ("Name of the file: ", my\_file.name)

print ("Opening mode : ", my\_file.mode)

output

A picture containing text, screenshot, monitor

Description automatically generated

7.Write a python program to handle simple runtime error.

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

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

c = a/b

print("a/b = %d" %c)

print("Hi I am other part of the program")

Output:

A picture containing text, screenshot, monitor

Description automatically generated

8.Write a program to handle multiple errors with one except statement.

try:

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

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

c = a/b

except:

print("Can't divide with zero")

Output:

Graphical user interface, text

Description automatically generated

9.Write a python program to create user defined exception.

# class MyError is extended from super class Exception

class User\_Error(Exception):

def \_\_init\_\_(self, value):

self.value = value

def \_\_str\_\_(self):

return(repr(self.value))

try:

raise(User\_Error("User defined error"))

except User\_Error as error:

print('A New Exception occured:',error.value)

Output

Graphical user interface, text

Description automatically generated with medium confidence

10.Write python code to illustrate clean-up final actions.

file = open('finally.txt', 'w')

try:

file.write("Testing1 2 3.")

print("Writing to file.")

except IOError:

print("Could not write to file.")

else:

print("Write successful.")

finally:

file.close()

print("File closed.")

Output

Graphical user interface, text

Description automatically generated

11.Write a program to demonstrate the use of class .

class Human():

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

self.name = name

self.age = age

def sayMyName(self):

print("My name is ", self.name)

def tellAge(self):

print("My age is ", self.age)

Mayur = Human("Mayur", 22)

Mayur.sayMyName()

Mayur.tellAge()

A screenshot of a computer

Description automatically generated

12.Write a python program to demonstrate inheritance.

class Animal:

def speak(self):

print("Animal Speaking")

#child class Dog inherits the base class Animal

class Dog(Animal):

def bark(self):

print("dog barking")

d = Dog()

d.bark()

d.speak()

Output:

A screenshot of a computer

Description automatically generated

13.Write a python program to demonstrate overloading.

#!/usr/bin/env python

class Human:

def sayHello(self, name=None):

if name is not None:

print('Hello ' + name)

else:

print('Hello ')

# Create instance

obj = Human()

# Call the method

obj.sayHello()

# Call the method with a parameter

obj.sayHello('Guido')

Output:

A screenshot of a computer

Description automatically generated

14.Write a python program to demonstrate overriding.

class Parent(object):

class Robot:

def action(self):

print('Robot action')

class HelloRobot(Robot):

def action(self):

print('Hello world')

r = HelloRobot()

r.action()

Output:

Text

Description automatically generated