**File Basics**

Modes of files:

“r”------open file for reading (default mode)

“w”-------open file for writing

“x”------creates a file only if doesn’t exist already

“a”-------add contents at the end of a file

“t”--------text mode (default mode)

“b”-----------binary mode

“+”------------update a file, both read and write.

**File creation and read**

1. For creating a file:

First make a new .txt file

Enter data

2. For reading a file:

F=open(“marks.txt”)

Content=f.read()

Print(content)

f.close()

3. F=open(“marks.txt” , “r”) #file will be opened in read mode

4. F=open(“marks.txt” , “rt”) #file will be opened in read, text mode

5. F=open(“marks.txt” , “rb”) #file will be opened in read, binary mode

6. F=open(“marks.txt”)

Content=f.read(3)

Print(content) #first three characters will be displayed

F=open(“marks.txt”)

Content=f.read(3) #next three characters will be displayed

Print(content)

7. F=open(“marks.txt”)

Content=f.read()

For line in content:

 Print(line) #each character will be displayed in a separate line.

8. F=open(“marks.txt”)

For line in f:

 Print(line) #each line will be displayed in a separate line.

9. F=open(“marks.txt”)

Print(f.readline()) #first line will be printed

Print(f.readline()) #second line will be printed

Print(f.readline()) #third line will be printed

Print(f.readlines()) #all lines will be printed in the form of a list

**File write and append**

1. F=open(“marks.txt” , “w”) # a new file will be created , “w” mode will replace all content in a file if it is already existing.

f.write(“he is a good man”) #this line will be added in the newly created file

f.close()

2. F=open(“marks.txt” , “a”) # marks will be opened in append mode

f.write(“he is a good man\n”) #this line will be added at the end of already existing file

f.close()

3. if we want to know how many characters are added in this write statement

F=open(“marks.txt” , “a”)

A=f.write(“he is a good man\n”)

Print(a) #total number of added characters will be printed

f.close()

4. when you want to open a file for both read and write

f=open(“marks.txt” , “r+”)

content=f.read()

print(content) or simply print(f.read())

f.write(“thank you”)

**Tell(), Seek()**

1. f.tell() # it will return where “f” is pointing now

2. f.seek(0) #it will take pointer”f” to the place u have entered

**Recursive vs iterative approach**

Recursion is when a function calls itself in its definition.

Iterative approaches for calculating factorial

num=int(input("enter the number"))
fac=1
i=1
while (i<=num):
 fac=fac\*(i)
 i+=1
print(fac)

num=int(input("enter the number"))

fac=1

i=1

for i in range(num):

 fac=fac\* (i+1)

print(fac)

factorial using recursive approach

def factorial(n):

 if n==1:

 return 1

 else:

 return n\*factorial(n-1)

num=int(input("enter the number"))

fac=factorial(num)

print(fac)

 Fibonacci series using iteration

num = int(input("enter the biggest number up to which u want to print fibonacci series \n"))

def fibonacci(num):

 n = 0

 m = 1

 print(n,m,end=" ")

 c = n + m

 while(c<num):

 c = n + m

 print(c,end=" ")

 n=m

 m=c

fibonacci(num)

This function takes the index and returns the number of Fibonacci series on that index.

def fibonacci(n):

 if(n==0):

 return 0

 elif(n==1):

 return 1

 else:

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

**Python built-in and external modules**

1. How to find the list of all python modules in your computer.

2. Import random

1. randomNumber = random.randint (0,5)

print(randomNumber) # it will print any number between 0 and 5

1. rand = random.random() # it will generate random numbers between 0 and 1.
2. List = [“apple” , “mango” , “orange”]

Choice=random.choice(list) # it will select any item from list and save it in choice

**Time module in python**

1. Execution time can be measured.
2. Import time

startTime=time.time()

i=1

while(i<45):

 print(“apple”)

 i+=1

print(“while loop takes”, time.time() – startTime , “seconds”)

startTime2 = time.time()

for I in range(45):

 print(“apple”)

print(“for loop takes”, time.time() – startTime2 , “seconds”)

1. localTime = time.asctime(time.localtime())

print(localTime) #it will print complete time with date and year

1. time.sleep(3) #it will wait for the execution of next line

**\*args and \*\*kwargs in python**

1. In Python, we can pass a variable number of arguments to a function using special symbols. We use \*args and \*\*kwargs as an argument when we are unsure about the number of arguments to pass in the functions.

There are two special symbols:

* \*args (Non Key value Arguments) (when we want to pass a list as argument)
* \*\*kwargs (Key value Arguments) (when we want to pass a dictionary as argument)

 def add1 (\*args):
 sum = 0
 for i in args:
 sum=sum+i
 return sum

print(add1(2,3))
print(add1(3,4,5))

list=["a","b","c"]
l=[1,2,3,4,5]
def printList (\*args):
 for l in args:
 print(l)

printList(\*list)

def printDictionary (\*\*kwargs):
 for key,value in kwargs.items():
 print(f"{key} is {value} ")
d1={"apple" : "fruit", "red": "colour"}
d2={"apple" : "fruit", "red": "colour", "cabbage": "vegetable"}
printDictionary(\*\*d1)
printDictionary(\*\*d2)

2. if you want to pass a normal and a \* argument then normal argument must come first.

3. Order of arguments…..normal, \*args, \*\*kwargs