CERTIFICATE

This is to certify that Mr./Ms. Hemil Chovatiya

with enrolment no 200303108003 *has successfully completed his*

laboratory experiments In Python Programing Workshop laboratory

during the academic year 2020-2021 *in branch* 3ITA1 *Batch*

Date: *Signature of lab teacher:*

Signature of HOD:

PRACTICAL – 1

AIM : What is Python? Installation of the python environment (anaconda),.python keywords, comments, and indentation, Python data types, Math and numbers in python.

(a) keywords

1. import keyword

print(keyword.kwlist)

ans:

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

2.

print("Sahil")

Ans: Sahil

3.

print('yo')

Ans: yo

 # print("Sahil") : use # for comment.

4.

'''print("Sahil")'''

Ans: 'print("Sahil")'

5.

"""print("Ram")"""

Ans: 'print("Ram")'

6.

print("Sahil") print("Sahil")

Ans:Sahil

7.

'''print("Sahil") print("Sahil")

print("Sahil")

print("Sahil")

'''

Ans: 'print("Sahil")\n print("Sahil")\n print("Sahil")\n print("Sahil")\n'

8.

'''print("Ram"); print("Ram"); print("Ram")

print("Ram") print("Ram")

print("Ram")

'''

Ans: 'print("Ram"); print("Ram"); print("Ram")\n print("Ram")\n print("Ram")\n print("Ram")\n'

9.

'''print("Sahil");

print("Sahil");

print("Sahil");

print("Sahil");

'''

Ans: 'print("Sahil");\n print("Sahil");\n print("Sahil");\n print("Sahil");\n'

10.

```
# Indentation:
print('Indentation') for
i in range(10):
    print(i)
```

Ans:

Indentation
0
1
2
3
4
5
6
7
8
9

(b) Indentation:

1.

```
print('Indentation')

for i in range(10):

print(i)  print(i)
```

Ans:

```
Indentation
0
0
1
1
2
2
3
3
4
4
5
5
6
6
7
7
8
8
9
9
```

(c) Data types

1. # Data types

```
a=2

print(a)
```

Ans: 2

2.

```
a=2.2
print(a)
```

Ans: 2.2

3.

```
# Data types
#(type(): used know which class a variable or a value belongs to.)

a=2.2

print(a)  print(type(a))
```

Ans:

```
2.2
<class 'float'>
```

4.

```
# Data types

a=2

print(a)  print(type(a))
```

Ans: 2

```
<class 'int'>
```

5.

a="sahil" print(a)

print(type(a))

Ans: sahil

<class 'str'>

6.

a='s'

print(a)

print(type(a))

Ans: s
<class 'str'>

(d) variables

1. a=2 b=3

print(a,b)

Ans: 2 3

2.

a,b,c=2,1.5,'kr' print(a,b,c)

Ans: 2 1.5 kr

(e) memory location 1.

p=2 print(id(p)) Ans:

140705156441936

(f) numbers:

1.

numbers in python:integers,floating point numbers & complex numbers a=29.5

print(type(a))

Ans: <class 'float'>

2.

To be a complex number, it is compulsary to be in the form of i+j. a=5+29j

print(type(a)) Ans:

<class 'complex'>

3.

a=i+9j

print(type(a)) Ans:

<class 'complex'>

4.

a=i+5

print(type(a)) Ans:

<class 'int'>

5.

input (from user) input('enter

a number: ')

Ans:

```
enter a number: 5

'5'
```

6.

```
# python input & output
a=input('enter a number: ')  b=input('enter

another number: ')  print(a+b)
```

Ans:

```
enter a number: 5
enter another number: 3
53
```

7.

```
a=int(input('enter a number: '))

b=int(input('enter anther number: '))  print(a+b)
```

Ans:

```
enter a number: 5
enter anther number: 3
8
```

8.

```
a=float(input('enter a number: '))

b=float(input('enter anther number: '))  print(a+b)
```

Ans:

```
enter a number: 5 enter
anther number: 29
34.0
```

9.

.format

a=2; b=5

print("the value of a is {} and b is {}".format(a,b))

Ans: the value of a is 2 and b is 5

10.

a=2; b=5

print("the value of a is {} and b is {}".format(b,a))

Ams: the value of a is 5 and b is 2

PRACTICAL – 2

AIM : CONTROL STATEMENTS (Include nested and at least 3 examples for A, B, C) A. If-else and if-elifelse B. While-loops C. For loops D. Break, continue, pass, and return E.

(a) Control statement

1.

 # Control statement: (if-else):use for take disition

```
x=int(input('enter a number: '))  if(x%2==0):

   print(x," is even")
else: print(x," is
   odd")
```

Ans:

```
enter a number: 5
5 is odd
```

(b) nested if-else: elif

1.

 # use elif between if & else

```
x=int(input('enter a number: '))  if(x>0):

   print("%d is positive"%x)
```

```
elif(x<0): print("%d is
nagative"%x)  else:

   print("%d is zero"%x)
```

Ans:

```
enter a number: 0
0 is zero
```

2.

```
a,b,c=7,6,5 if(a>=b)
and (a>=c):

   largest=a

elif(b>=a) and (b>=c):  largest=b

else:

   largest=c print("largest no
is:{}".format(largest))
```

Ans: largest no is:7

3.

```
x=29

if(x==0):

   print("%d is zero"%x)

else:

   print("%d is decimal no"%x)
```

Ans: 29 is decimal no

(c) while loop : condition

1.

```
a=[5,8,15,4,2]
i=0

mul=1


while i<len(a):
   mul*=a[i]

   i+=1 print('product is:{}'.format(mul)) # print statement in while


   loop
```

Ans:

```
product is:5
product is:40
product is:600
product is:2400
product is:4800
```

2.

```
# while loop: condition

a=[5,8,15,4,2] i=0

mul=1


while i<len(a):

   mul*=a[i]

   i+=1

print('product is:{}'.format(mul))     # print statement is out of while loop
```

Ans: product is:4800

3.

```
# while loop: condition

a=[5,8,15,4,2] i=0

add=0

while i<len(a):  add+=a[i]


   i+=1

   print('product is:{}'.format(add))
```

Ans:

```
product   is:5
product is:13
product is:28
product is:32
product is:34
```

4.

```
# while loop: condition

a=[5,8,15,4,2] i=0


add=0 while

i<len(a):

add+=a[i]

i+=1


print('addition of array is:{}'.format(add))
```

Ans: addition of array is:34

5.

```
# prime or not  n=int(input("enter

a number: "))

i=2

j=0  while(i<n):


   if(n%i==0): print("{} is not
      prime".format(n))

      j+=1

      break

   else:


      i+=1
      continue

   i+=1


if j==0:

   print("{} is prime".format(n))
```

Ans:

```
enter a number: 89
89 is prime
```

6.

```
# prime or not  n=int(input("enter

a number: "))

i=2

j=0  while(i<n):
```

```
    if(n%i==0): print("{} is not
       prime".format(n))
       j+=1
       break
    i+=1

if j==0: print("{} is
   prime".format(n))
```

Ans:

```
enter a number: 37
37 is prime
```

7.

```
# using while loop to check number is prime or not

n=int(input("enter a number:")) if n>1:

    for i in range(2,n):
       if(n%i)==0: print("{} is not a prime
          number",n)  break

    else: print("{} is a prime
number",n) else: print("{} is not a
prime number",n)
```

Ans:

```
enter a number:13 {} is
a prime number 13
```

8.

```
# using while loop to check number is prime or not  n=int(input("enter

a number:"))


if n>1: for i in
   range(2,n):

      if(n%i)==0: print("{} is not a prime

         number",n)  break


   else: print("{} is a prime

number",n) else:

   print("{} is not a prime number",n)
```

Ans:

```
enter a number:-23 {} is not
a prime number -23
```

(d) for loop

#used to interate over a sequence(list,tuple,string) or other interable objects #SYNTAX:

for i in sequence:

1.

```
l=[5,12,29,15,40]
mul=1

for i in l:


   mul*=i

print('product is {}'.format(mul))
```

Ans: product is 1044000

(e) break statement

 # use to terminate the loop containing it.

 # SYNTAX: break

1.
```
for i in
"university":
    if(i=='r'):
        break
    print(i)
```

Ans:

u n
i
v
e

(f)continue statement 1.

```
for i in 'university':
    if(i=='r'):
        continue
    print(i)
```

Ans:

u
n
i
v
e
s
i
t
y

(g) pass

 # pass (it means nothing) # use

to perform null operation.

1. for i in
'university':

pass

print(i)

Ans: y

(h). return statement

 # used to end exicution of the function call and return the result to the caller #

def fun():

1. def fun():

return 5+10

print("hi")

print(fun())

Ans: 15

2.

def s(): return

5-10

print("hi")

print(s())

Ans: -5

3.

def sahil():

return 5*10

print("hi") print(sahil())

Ans: 50

4. def fun():

return 5/10

print("hi")

print(fun())

Ans: 0.5

PRACTICAL – 3

AIM : Functions in python - Inbuilt functions in python, User-defined functions (with and without argument & with and without return type), Lambda faction in python.

function

(i)it is a group of related state that perform a specific task.

def fun()

(ii)with the help of function, we break a program into smaller chuncks.

(iii)it avoid repetation of code & make code reuseable.

(iv)SYNTAX:

def function-name(perameters):

doc string

//documentation

statement(s)

(a) range function

 # to generate avsequence of numbers

 # (i) range(n)

 # (ii) we can define the start,stop & step size: range(start,stop,step size)

1.

```
for i in range(10):
    print(i)
```

Ans:
0
1
2
3
4
5
6
7
8
9

2.

for i in range(0,10,2): print(i)

Ans:
0
2
4
6
8

3.

for i in range(1,25,5): print(i)

Ans:
1
6
11
16
21

4.

l=['sahil','ram','shayam','pr','t'] for

i in range(len(l)):

```
    print(i)
```

Ans:

```
0
1
2
3
4
```

5.

```
l=['sahil','ram','smit','pr','t']

for i in range(len(l)):

    print(l[i])
```

Ans:

```
sahil
ram
smit
pr
t
```

6.

```
def func_name(parameter):

    '''Doc string

    '''

    Statements(s) def

print_name(name):

print("HELLO"+str(na

me))
```

```
print_name('sahil')
```

Ans: HELLOsahil

7.

```
def sum1(lst):
    '''
    This function return the sum of all the elements of lst
    '''

    s=0 for n
    in lst:

        s=s+n

    return s

p=sum1([2,4,8,10])

print(p)
```

Ans: 24

8. # HCF: Highest Common Factor

```
def HCF(a,b):

    if(a>b):
        smaller=b

    else:

        smaller=a

    HCF=0

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

```
    if (a%i==0)and(b%i==0):

        HCF=i
  return HCF

m= HCF(4,12)

print("HCF of a and b is:",m)
```

Ans: HCF of a and b is: 4

10.

```
a=int(input("enter a number"))

b=int(input("enter a second number"))  def

HCF(a,b):

  if(a>b):
    smaller=b

  else:

    smaller=a

  HCF=0
  for i in range(1,smaller+1):
    if (a%i==0)and(b%i==0):

      HCF=i

return HCF  m=

HCF(a,b)

print("HCF of a and b is:",m)
```

Ans:

enter a number34 enter
a second number4
HCF of a and b is: 2

(b). divmod:

 # EX: (11,2)=> ans: (5,1) # where 5

quotiont is & 1 is reminder

 # reduce:

 # to perform some computation on a list. # EX:

[1,2,3,4,5]=> ans:120 (multiply one by one)

1.

```
from functools import reduce  def

mul(x,y):

   return x*y  l=[1,2,3,4,5]


m=reduce(mul,l)
print(m)
```

Ans: 120

 # USER DEFINE FUNCTION:

 # TO DO A CERTAIN A SPECIFIC TASK.

2.

```
def mul(a,b):

   p=a*b  return

   p
```

```
a=int(input("enter a number"))
b=int(input("enter a second number"))
print(mul(a,b))
```

Ans:

```
enter a number6
enter a second number7
42
```

(c) LAMBDA/ ANONYMOUS FUNCTION

 # SYNTAX: lambda arguments : expression

 # ex: lambda a : a*5

 # where lambda is a keyword

1.

```
mul=lambda a: a*5  print(mul(2))
```

Ans: 10

2.

```
mul=lambda a: a/5  print(mul(20))
```

Ans: 4.0

PRACTICAL – 4

AIM : List, Tuples, Dictionaries & Sets in python Input and output in python.

(a) list

 # (1) collection of data elements(number/float/characters etc.)

 # (2) it is one of the sequance data structure.

 # (3) it is incluese denoted with [] sign.

 # (4) each items seperated with (,)

 # (5) list is a mutable(changable) data structure

 # List creation

 # empty list--> a=[]
 # list--> a=['ram',"computer","mobile"]

 # list--> a=[1,2,3,8]

 # nested list--> a=[[1,2],[3,8]]

 # list of different datatypes a=[4,"computer",9.5,"mobile"]

 # list length

 # l=[4,"computer",9.5,"mobile"] --> length=4

 # print(len(l)) -->4 # where l --> is list name.

1.

l=[4,"computer",9.5,"mobile"]

a=len(l) print(a)

Ans: 4

 # l.append('bi')# where l --> is list name. #

print((l)) -->[4,"computer",9.5,"mobile",'bi']

2.

l=[4,"computer",9.5,"mobile"]

l.append('bi')

print((l))

Ans: [4, 'computer', 9.5, 'mobile', 'bi']

#add element

 # # 0 1 2 3 <--

array

#l=[4,"computer",9.5,"mobile"]

#l.insert(2,'sahil') # where l --> is list name. # #ans -->

l=[4,"computer",'sahil',9.5,"mobile"]

3.

l=[4,"computer",9.5,"mobile"]

l.insert(2,'sahil')

print(l)

Ans: [4, 'computer', 'sahil', 9.5, 'mobile']

 # remove any element

 # l=[4,"computer",'Sahil',9.5,"mobile"]

 # l.remove(9.5) # where l --> is list name.

 # print(l) # where l --> is list name.

 # #ans--> l=[4,"computer",'Sahil',"mobile"]

4.

l=[4,"computer",'Sahil',9.5,"mobile"]

l.remove(9.5)

print(l)

Ans: [4, 'computer', 'Sahil', 'mobile']

5.

 # only able to remove one element

l=[4,"S",'Sahil',4,3.5,"Hi",5,4]
l.remove(4)

print(l)

Ans: ['S', 'Sahil', 4, 3.5, 'Hi', 5, 4]

6.

 # reverse function

l=[1,2,3,4,8,6]

l.reverse() print(l)

Ans: [6, 8, 4, 3, 2, 1]

7.

sorted in increasing element l=[1,2,3,4,8,6]

l.sort() print(l)

Ans: [1, 2, 3, 4, 6, 8]

 # # count 1 in list

 # l=[1,2,1,3,4,1,8,1,6,1]
 # l.count(1) -->ans=5

 # print(l)

8.

l=[1,2,1,3,4,1,8,1,6,1]

a=l.count(1) print(a)

Ans: 5

(b) tuple

 # (1)it is similler data structure

#(2)--> ()
 # (3) we cant change the element of tuple once it is assigned

 # (4) immutable data structure

 # a=()--> empty tuple

 # t=(1,3,5,6) #

t=(2,'D',6,'t')

1.

```
a=(1,3,5,6)
print(a[2])
```

Ans: 5

2.

 # concatination

```
a=(1,3,5,6)+ (6,5,2)

print(a)
```

Ans: (1, 3, 5, 6, 6, 5, 2)

3.

```
a=(1,3,5,6)+ (6,5,2)
print(a.count(5))
```

Ans: 2

4.

 # index

```
a=(1, 3, 5, 6, 6, 5, 2)

# 0 1 2 3 4 5 6

print(a.index(5))
```

Ans: 2

5.

 # length

```
a=(1, 3, 5, 6, 6, 5, 2)

# 0 1 2 3 4 5 6
print(len(a))
```

Ans: 7

6.

 # find max

```
t=(1, 3, 5, 6, 6, 5, 2)
print(max(t))
```

Ans: 6

7.

 # find min

```
t=(1, 3, 5, 6, 6, 5, 2)
print(min(t))
```

Ans: 1

 # set --> it is a collection of unique elements

 # it is an unordered collection of items or elmet. and all the elements should be are unique.

 # it is a mutable data structure

 # set creation
 # s={} --> empty tuple

 # s={1,3,5,6}

 # print(type(s))--> ans='set'

8.

```
s={1,3,5,6}
print(type(s))
```

Ans: <class 'set'>

 # covert into set

 # s=set([1,3,5,6])

 # print(s)--> ans={1, 3, 5, 6}

9.

```
s=set([1,3,5,6
]) print(s)
```

Ans: {1, 3, 5,
6}

10.

 # convert into

list

s=list({1,3,5,6})

print(s) Ans: [1, 3,

5, 6]

12.

 # discard(remove) element from set

s={1,3,5,6}
s.discard(3)

print(s)

Ans: {1, 5,

6}

13.

 # clear all element from set.... and set do empty--> ans=set()

s={1,3,5,6}

s.clear()

print(s)

Ans: set()

 # find element from set

 # s={1,3,5,6}

 # print(s[2])

 # ans-->error

 # because set does not arrat type

14.

add element in set --> it do in increase order s={1,3,5,6}

s.add(4) print(s) Ans:

{1, 3, 4, 5, 6}

15.

add element in set s={1,3,5,6}

s.add(2.5) print(s) Ans:

{1, 2.5, 3, 5, 6}

16.

add element in set s={1,7,5,6}

s.add(4) print(s)

Ans: {1, 4, 5, 6, 7}

(c) Dictionary

 # it is an unordered collection of items: # d={} -> empty dictionary

 # d={1:'xyz',2:'ab'} -> where 1& 2 are key and 'xyz' & 'ab' are values

 # d={'name':'kt','age':'19','address':'PU'} 1.

d={1:'xy',2:'kt',3:'pr'} print(d[2])

Ans: kt

2.

d={1:'xy',2:'kt',3:'pr'} print(d.get(2))

Ans: kt

3.

 # adding an element in the dissending

d={1:'xy',2:'kt',3:'pr'}
d[4]='kr' print(d)

Ans: {1: 'xy', 2: 'kt', 3: 'pr', 4: 'kr'}

4.

adding an element in the dissending d={1:'xy',2:'t',4:'pr'}

d[3]='k' print(d)

Ans: {1: 'xy', 2: 't', 4: 'pr', 3: 'k'}

5.

adding an element in the dissending d={1:'xy',2:'t',3:'pr'}

d[5]='k' print(d)

Ans: {1: 'xy', 2: 't', 3: 'pr', 5: 'k'}

6.

 # update element

d={1:'xy',2:'t',3:'pr'} d[2]='k'

print(d) Ans: {1: 'xy', 2: 'k', 3:

'pr'}

7.

update element

d={1:'xy',2:'t',3:'pr'}

d[1]='k' print(d)

Ans: {1: 'k', 2: 't', 3: 'pr'}

8.

update element

d={1:'xy',2:'t',3:'pr'}

print(d)

i=d[2]

d[2]=d[3]

d[3]=i

print(d)

Ans:

{1: 'xy', 2: 't', 3: 'pr'}
{1: 'xy', 2: 'pr', 3: 't'}

9.

```
# update element

d={1:'xy',2:'t',3:'pr'

}  d[4]='p' print(d)


d[2]='kt'

print(d)


i=d[2]

d[2]=d[3]

d[3]=i print(d)
```

Ans:

```
{1: 'xy', 2: 't', 3: 'pr', 4: 'p'}
{1: 'xy', 2: 'kt', 3: 'pr', 4: 'p'}
{1: 'xy', 2: 'pr', 3: 'kt', 4: 'p'}
```

10.

 # delete element from dictionary

```
d={'name':'kt','age':'19','address':'PU'}

print(d.pop('age'))
```

Ans: 19

12.

```
# delete element from

dictionary

d={1:'kt',2:'19',3:'PU'}

print(d.pop(2))  print(d)
```

Ans:
19
{1: 'kt', 3: 'PU'}

13.

```
# delete element from dictionary

d={'name':'sg','age':'19','address':'PU'

}  print(d.pop('age')) print(d)
```

Ans:

19
{'name': 'sg', 'address': 'PU'}

14.

```
# delete a perticular

key

d={1:'xy',2:'kt',3:'pr'}

del d[2]  print(d)
```

Ans: {1: 'xy', 3: 'pr'}

RAM
Random access memory

volatile memory

array

15.

```
a=["sahil",4,5,[56,74,29]]
print(a[3][1])
```

Ans: 74

16.

```
a=["sahil",4,5,[56,74,29]] print(a[-
3])
```

Ans: 4

17.

```
a=["sahil",4,5,[56,74,29]] print(a[-
1][1])
```

Ans: 74

18.

```
a=["sahil",'smit',4,5,2,29,[56,74,29]
] print(a[1:4])
```

Ans: ['smit', 4, 5]

19.

```
a=["sahil",'ram',4,5,2,29,[56,74,29]
] print(a[1:5+1])
```

Ans: ['ram', 4, 5,
2, 29]

20.

```
a=["sahil",'ram',4,5,2,29,[56,74,29]
] print(a[1:5+2])
```

Ans: ['ram', 4, 5,
2, 29, [56, 74, 29]]

21.

```
a=["sahil",'smit',4,5,2,29,[56,74,29]]

print(a[:])
```

Ans: ['sahil', 'smit', 4, 5, 2, 29, [56, 74, 29]]

22.

```
a=["sahil",'smit',4,5,2,29,[56,74,29]] print(a)
```

Ans: : ['sahil', 'smit', 4, 5, 2, 29, [56, 74, 29]]

23.

```
a=["sahil",'raj',4,5,2,29,[56,74,29]]

a[2]=2002


print(a)
```

Ans: ['sahil', 'raj', 2002, 5, 2, 29, [56, 74, 29]]

24.

```
a=["sahil",'smit',4,5,2,29,[56,74,29]]

a[2]=2002

a.append("rishi")

print(a)
```

Ans: ['sahil', 'smit', 2002, 5, 2, 29, [56, 74, 29], 'rishi']

25.

```
a=["sahil",'raj',4,5,2,29,[56,74,29]]

a[2]=2002


a.append("smit")
```

```
a.insert(4,"sg")

print(a)
```

Ans: ['sahil', 'raj', 2002, 5, 'sg', 2, 29, [56, 74, 29], 'smit']

26.

```
a=["sahil",'ram',4,5,2,29,[56,74,29]]

a[2]=2002
print(a)

a.append("smit")

print(a)

a.insert(4,"ss"

)  print(a) del

a[2]  print(a)

del a[1:3]

print(a)
```

Ans:

['sahil', 'ram', 2002, 5, 2, 29, [56, 74, 29]]

['sahil', 'ram', 2002, 5, 2, 29, [56, 74, 29], 'smit']

['sahil', 'ram', 2002, 5, 'ss', 2, 29, [56, 74, 29], 'smit']

['sahil', 'ram', 5, 'ss', 2, 29, [56, 74, 29], 'smit']

['sahil', 'ss', 2, 29, [56, 74, 29], 'smit']

PRACTICAL - 5

AIM : Inputs and outputs from the file, Operations on files, Modes and methods of files.

(a) file(write & read)

1.

```
f = open('C:\\Users\\DELL\\Desktop\\Sahil\\link.txt')

f.read()
```

Ans: 'DE LAB:\n https://circuitverse.org/simulator\n https://www.multisim.com/create/\n\nMY FACE BOOK LINK:\n https://www.facebook.com/profile.php?id=100053302174534\n\nMSN:\nhttps://www.msn.com/en-in/?ocid=wispr&pc=u477\n\nPU-WIFI:\nhttp://10.0.0.11:8090/httpclient.html?u=http://www.msftconnecttest.com/redirect'

2.

```
f = open('C:\\Users\\DELL\\Desktop\\Sahil\\link.txt')

f.read(10)
```

Ans:

'DE LAB:\n '

3.

```
f = open('C:\\Users\\DELL\\Desktop\\Sahil\\link.txt','w')

f.write('I am Sahil Gandhi')
```

Ans: 16

4.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link.txt','r')
f.read()
```

Ans: 'I am Sahil Gandhi'

5.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link.txt','r')
f.read(6)
```

Ans: 'I am S'

6.

```
f.tell()
```

Ans: 16

7.

```
f.seek(0)
```

Ans: 0

8.

```
f.seek(5)
```

Ans: 5

9.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link.txt','r')
f.read()
```

Ans: 'I am Sahil Ganadhi'

10.

f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link.txt','w')

f.write('sahil\n')

f.write('shyam')

Ans: 5

11.

f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link.txt','r')

f.readline()

Ans: 'sahil\n'

12.

f.readline()

Ans: 'smit'

13.

f.seek(0)

Ans: 0

14.

f.readlines() Ans:

['sahil\n', 'shyam']

15.

f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link.txt','w')

f.write('sahil\n')

f.write('shyam hii')

Ans: 9

16.

import os

os.rename('C:\\Users\\DELL\\Desktop\\SAHIL\\lec.link.txt','C:\\Users\\DELL\\Desktop\\SAHIL\

\link.txt')

17.

f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link1.txt','w')

f.write('sahil\n')

f.write('shyam hii')

Ans: 9

18.

import os

os.rename('C:\\Users\\DELL\\Desktop\\SAHIL\\link1.txt','C:\\Users\\DELL\\Desktop\\SAHIL\\li
nkt.txt')

19.

f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link1.txt','r') f.read()

Ans: 'sahil\nshyam hii'

20.

```
os.remove('link1.txt')
```

21.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link4.txt','a')

f.write('sahil\n')

f.write('shyam')
```

Ans: 5

22.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link4.txt')

f.read()
```

Ans: 'sahil\nshyam hiisahil\nshyam hiisahil\nshyam'

23.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link4.txt','w')

f.write('hellosahil\n')

f.write('shyam')
```

Ans: 5

24.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link4.txt')

f.read()
```

Ans: 'hellosahil\nshyam'

25.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link8.txt','w')

f.write('Hellosahil\n')

f.write('shyam hii')
```

Ans: 9

26.

```
f.close()
```

27.

```
import os

os.rename('C:\\Users\\DELL\\Desktop\\SAHIL\\link8.txt','C:\\Users\\DELL\\Desktop\\SAHIL\\link10.txt')
```

28.

```
f = open('C:\\Users\\DELL\\Desktop\\SAHIL\\link10.txt')

f.read()
```

Ans: 'Hellosahil\nshyam hii'

29.

```
f.close()
```

30.

```
os.remove('C:\\Users\\DELL\\Desktop\\SAHIL\\link10.txt')
```

PRACTICAL – 6

AIM : EXCEPTION HANDLING (compile time and run time) Try - except -finally Try -except - else. (a) python built in expectation 1.

dir(_builtins_)

Ans: ['ArithmeticError',
'AssertionError',
'AttributeError',
'BaseException',
'BlockingIOError',
'BrokenPipeError',
'BufferError',
'BytesWarning',
'ChildProcessError',
'ConnectionAbortedError',
'ConnectionError',
'ConnectionRefusedError',
'ConnectionResetError',
'DeprecationWarning',
'EOFError',
'Ellipsis',
'EnvironmentError',
'Exception',
'False',
'FileExistsError',
'FileNotFoundError',
'FloatingPointError',
'FutureWarning',
'GeneratorExit',
'IOError',
'ImportError',
'ImportWarning',
'IndentationError',
'IndexError',
'InterruptedError',
'IsADirectoryError',
'KeyError',
'KeyboardInterrupt',
'LookupError',

'MemoryError',
'ModuleNotFoundError',
'NameError',
'None',
'NotADirectoryError',
'NotImplemented',
'NotImplementedError',
'OSError',
'OverflowError',
'PendingDeprecationWarning',
'PermissionError',
'ProcessLookupError',
'RecursionError', 'ReferenceError',
'ResourceWarning',
'RuntimeError',
'RuntimeWarning',
'StopAsyncIteration',
'StopIteration',
'SyntaxError',
'SyntaxWarning',
'SystemError',
'SystemExit',
'TabError',
'TimeoutError',
'True',
'TypeError',
'UnboundLocalError', 'UnicodeDecodeError',
'UnicodeEncodeError',
'UnicodeError',
'UnicodeTranslateError',
'UnicodeWarning',
'UserWarning',
'ValueError',
'Warning',
'WindowsError',
'ZeroDivisionError',
'_IPYTHON_',
"_loader build_c__',lass_', '
debug_',
'_doc__',
'_import_',

'_name__',
'_package__', '
spec_', 'abs',
'all',
'any',
'ascii',
'bin',
'bool',
'breakpoint',
'bytearray',
'bytes',
'callable',
'chr',
'classmethod',
'compile',
'complex',
'copyright',
'credits',
'delattr',
'dict',
'dir',
'display', 'divmod',
'enumerate',
'eval',
'exec',
'filter',
'float',
'format',
'frozenset',
'get_ipython',
'getattr',
'globals',
'hasattr',
'hash',
'help',
'hex',
'id',
'input',
'int',
'isinstance',
'issubclass',
'iter',

'len',
'license',
'list',
'locals',
'map',
'max',
'memoryview',
'min',
'next',
'object',
'oct',
'open',
'ord',
'pow',
'print',
'property',
'range',
'repr',
'reversed',
'round',
'set',
'setattr', 'slice',
'sorted',
'staticmethod',
'str',
'sum',
'super',
'tuple',
'type',
'vars',
'zip']

2. import
sys

lst=['a',0,5] for

i in lst:

```
try: print("The entered value
   is",i)  p=1/int(i)

except:
   print("Oops!",sys.exc_info()[0],"occurred."
   )  print("Next entry is:") print("  ")
   print("The
reciprocal of 5 is 0.2")
```

Ans:

```
The entered value is a Oops! <class
'ValueError'> occurred.  Next entry
is:
--------------------

The reciprocal of 5 is 0.2
The entered value is 0
Oops! <class 'ZeroDivisionError'> occurred.
--------------------
Next entry is:

The reciprocal of 5 is 0.2
The entered value is 5
The reciprocal of 5 is 0.2
```

3.

```
import sys

lst=['a',0,5] for

i in lst:

   try: print("The entered value
      is",i)  p=1/int(i)

   except(valueError): print("This
   is a value error.")
```

except(ZeroDivisionError):

print("This is a zero error.")

except: print("Some other error.") print("The reciprocal of",l,"is",p)

Ans:

```
The entered value is a
---------------------------------------------------------------------------
ValueError    Traceback (most recent call last)
<ipython-input-9-47df665b8035> in <module>
     5    print("The entered value is",i)
----> 6 p=1/int(i) 7
          except(valueError)
     :

ValueError: invalid literal for int() with base 10: 'a'

During handling of the above exception, another exception occurred:

NameError     Traceback (most recent call last)
<ipython-input-9-47df665b8035> in <module>
     5        print("The entered value is",i)
     6        p=1/int(i)
----> 7 except(valueError): 8
          print("This is a value error.") 9
          except(ZeroDivisionError):

NameError: name 'valueError' is not defined
```

PRACTICAL – 7

AIM : Regular expression (REGEX) in python.

(a) re.match

SYNTAX: re.match(pattern,string)

where re ia a regular expression

where match is a function

1.

```
import re

re.match('a','parul')
```

2.

```
import re

re.match('pa','parul')
```

ans: <re.Match object; span=(0, 2), match='pa'>

3.

```
import re

re.match('p','parul')
```

ans: <re.Match object; span=(0, 1), match='p'>

4.

```
import re

re.match('par','parul')
```

ans: <re.Match object; span=(0, 3), match='par'>

(b) search function

SYNTAX: re.search(pattern,string)

1.

```
import re

re.search('r','university')
```

ans: <re.Match object; span=(5, 6), match='r'>

2.

```
import re

re.search('univ','university')
```

ans: <re.Match object; span=(0, 4), match='univ'>

3.

```
import re

re.search('ers','university')
```

ans: <re.Match object; span=(4, 7), match='ers'>

4.

bool(re.match('a','parul'))

Ans: False

5.

bool(re.match('pa','parul'))

Ans: True

6.

bool(re.match('p','parul'))

Ans: True

7.

re.search('v','univervvsity') Ans:

<re.Match object; span=(3, 4), match='v'>

8.

re.search('r','sahilgandhi') Ans: <re.Match

object; span=(2, 3), match='i'>

9.

re.search('u','niver\nsi\nuty')

Ans: <re.Match object; span=(9, 10), match='u'>

10.

re.search('v.+','university').group()

Ans: 'versity'

Practical-8

Aim:- Decorators in python.

Syntax for Decorator:

```
@gfg_decorator def
hello_decorator():

    print("Gfg")

'''Above code is equivalent to -


def hello_decorator():


    print("Gfg")

hello_decorator = gfg_decorator(hello_decorator)'''

# defining a decorator:
def hello_decorator(func):

       # inner1 is a Wrapper function in
       # which the argument is called

       # inner function can access the outer
       local  # functions like in this case
       "func" def inner1():
              print("Hello, this is before function execution")

              # calling the actual function now
```

```
            # inside the wrapper function.  func() print("This is
            after function execution")

        return inner1

# defining a function, to be called inside wrapper  def
function_to_be_used():
        print("This is inside the function !!")

# passing 'function_to_be_used' inside the  # decorator to control its
behavior function_to_be_used = hello_decorator(function_to_be_used)

# calling the function  function_to_be_used()
```

output:-

```
Hello, this is before function execution  This is inside the
function !!
This is after function execution.
```