St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology
A.Y. 2023-24
Class: SE-ITA/B, Semester: IV
Subject: **Python Lab**.

Experiment – 6: Functions and Exception Handling in Python

1. **Aim:** To implement a python program to define functions for the following:
   a) Linear Search Algorithm
   b) To implement a python program to define functions for the following:
.   I. To handle multiple exceptions.
       II. Create a list of Strings,if the string ends with ‘ing’ then replace it by ‘ly’.
       III. To implement anonymous functions(lambda,map,reduce,filter)

2. Prerequisite: Knowledge of Exceptions and functions in python.

3. **Objectives:** After performing this experiment, the student will be able to understand and write conditional statements in Python.

4.**Requirements:** Personal Computer (PC), Windows /Linux Operating System, IDLE 3.6 for Python3.

5. Pre-Experiment Exercise:

Theory:

Python Functions:

In Python, function is a group of related statements that perform a specific task. Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable. Furthermore, it avoids repetition and makes code reusable.

Syntax of Function:

def function_name (arguments):

‘”function_body’”

Statement(s)

Keyword **def** marks the start of function header.

A function name to uniquely identify it. Function naming follows the same rules of writing identifiers in Python.

Parameters (arguments) through which we pass values to a function. They are optional.

A colon (**:**) to mark the end of function header.

Optional documentation string (function_body) to describe what the function does.

One or more valid python statements that make up the function body. Statements must have same indentation level (usually 4 spaces).

An optional return statement to return a value from the function.

6. Laboratory Exercise

A. Procedure:

- Open Idle for python
- Open editor in Idle from menu file-new
- Type python code with proper syntax
- Save file with .py extension
- Execute the code inside the saved file using shortcut key F5 or using menu: Run-Run module

B. Program code with comments:

Write and execute your program code to achieve the given aim and attach it **with your own comments with neat indentation**.

7. Post-Experiments Exercise

A. Extended Theory:

1. Explain the concept of Recursion in Python with one example.
2. Explain exception-handling in Python along with example and **list down** all the types of errors that can be handled in try-except block
3. Explain about types of parameters and Arguments in functions.

B. Questions/Programs:

1. Write a Python program to demonstrate the use of iterator and generator functions.
2. Write a Python program to calculate sum of first 5 natural numbers using recursion.

C. Conclusion:

1. Write what was performed in the program(s).
2. What is the significance of the program(s).

D. References:

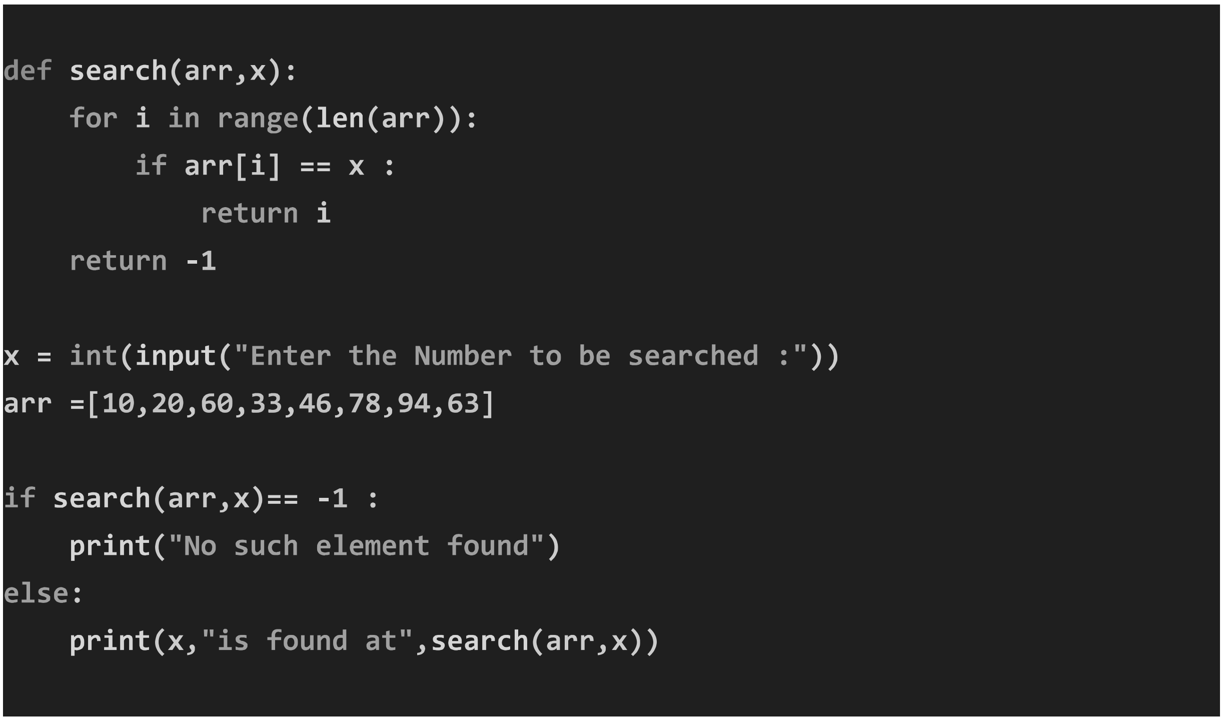
1. James Payne, ”Beginning Python: Using Python 2.6 and Python 3.1”,WroxPublication.
2. https://www.python.org/
3. www.pythonforbeginners.com

IN-LAB Program:

A)To implement a python program to define functions for the following:

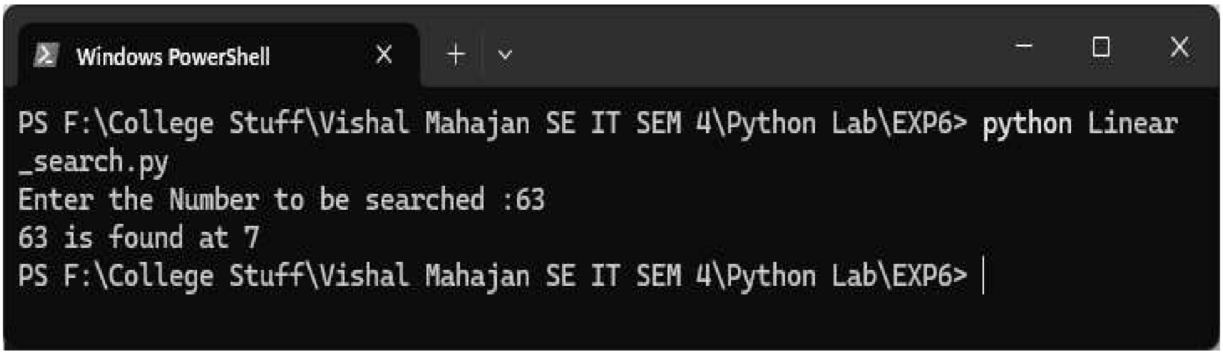
Linear Search Algorithm

Code:



```
def search(arr,x):
    for i in range(len(arr)):
        if arr[i] == x :
            return i
    return -1

x = int(input("Enter the Number to be searched :"))
arr =[10,20,60,33,46,78,94,63]

if search(arr,x)== -1 :
    print("No such element found")
else:
    print(x,"is found at",search(arr,x))
```

Output:



```
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> python Linear
_search.py
Enter the Number to be searched :63
63 is found at 7
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6>
```

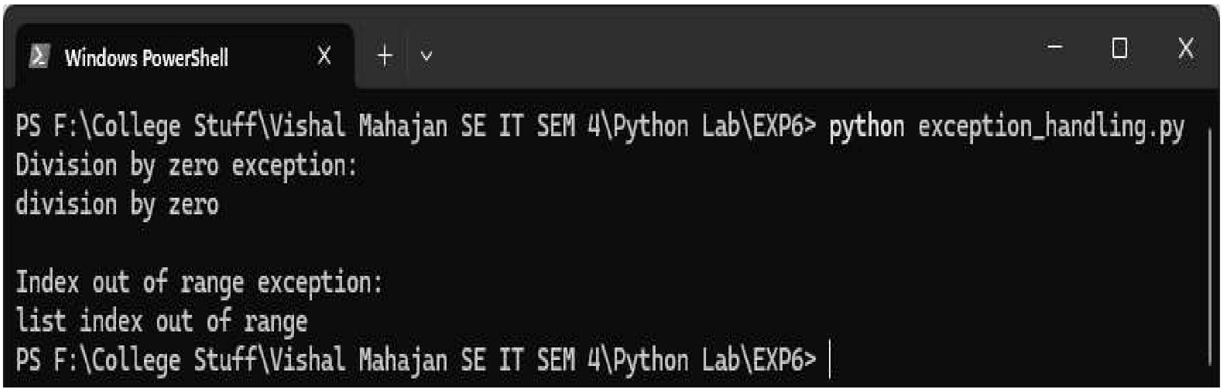
B) To implement a python program to define functions for the following:

1) To handle multiple exceptions.

Code:

```
print("Division by zero exception:")
def zero_exception(num1,num2):
    try:
        divide=num1/num2
    except ZeroDivisionError as e :
        print(e)

zero_exception(10,0)

print("\nIndex out of range exception:")
def array_exception():
    try:
        arr=[1,2,3,4,5]
        print(arr[5])
    except IndexError as e:
        print(e)
array_exception()
```

Output:



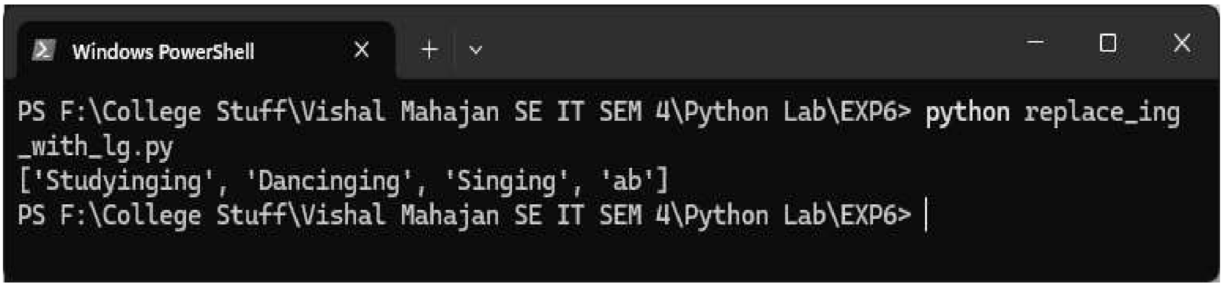
```
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> python exception_handling.py
Division by zero exception:
division by zero

Index out of range exception:
list index out of range
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6>
```

2)Create a list of Strings,if the string ends with ‘ing’ then replace it by ‘ly’

Code:

```
EXP6 = ["Studying","Dancing","Sing","ab"]

def add_string(element):
    length = len(element)

    if length > 2 :
        if element[3:]=="ing":
            element += "ly"
        else:
            element += "ing"

    return element



for i in range(len(EXP6)):
    element=EXP6[i]
    EXP6[i]=add_string(element)

print(EXP6)
```

Output:



```
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> python replace_ing
_with_lg.py
['Studyinging', 'Dancinging', 'Singing', 'ab']
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> |
```

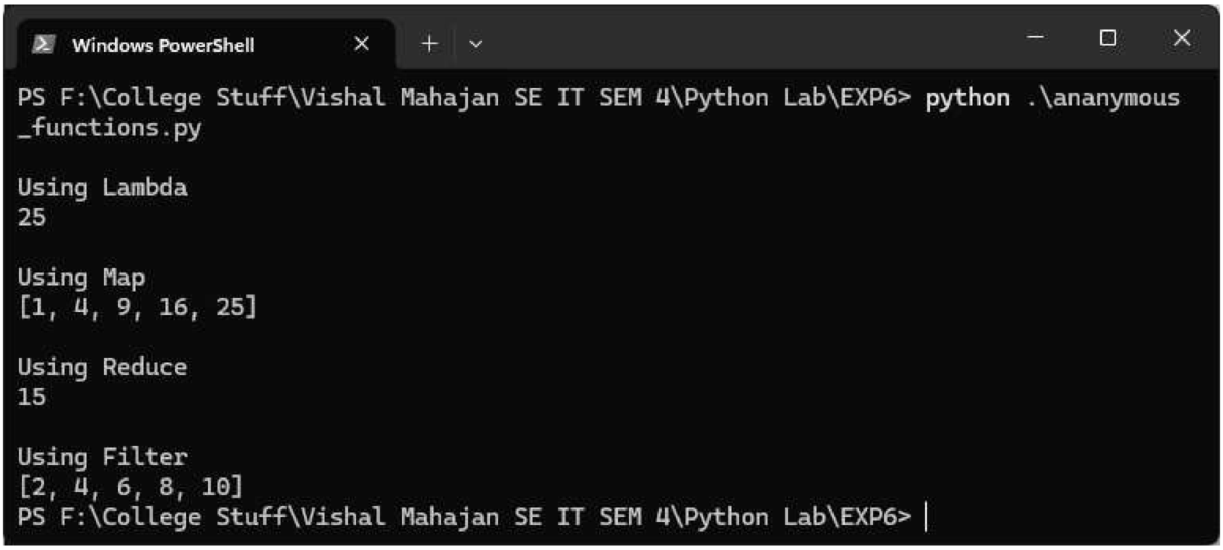
3)To implement anonymous functions(lambda,map,reduce,filter)

Code:

```
# Using Lambda
#Lambda is a small anonymous function. It can take any number of
arguments, but can only have one expression.
print("Using Lambda")
f=lambda x:x*x
print(f(5))

# Using Map
#The map() function executes a specified function for each item in an
iterable. The item is sent to the function as a parameter.
print("\nUsing Map")
def square(x):
    return x*x
numbers=[1,2,3,4,5]
result=map(square,numbers)
print(list(result))

# Using Reduce
#The reduce() function is defined in the functools module. It applies a
rolling computation to sequential pairs of values in a list.
print("\nUsing Reduce")
from functools import reduce
def add(x,y):
    return x+y
numbers=[1,2,3,4,5]
result=reduce(add,numbers)
print(result)
```

```
# Using Filter
#The  filter()  function  returns  an  iterator  were  the  items  are  filtered
through a function to test if the item is accepted or not.
print("\nUsing Filter")
def is_even(x):
    return x%2==0
numbers=[1,2,3,4,5,6,7,8,9,10]
result=filter(is_even,numbers)
print(list(result))
```

Output:



```
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> python .\ananymous
_functions.py

Using Lambda
25

Using Map
[1, 4, 9, 16, 25]

Using Reduce
15

Using Filter
[2, 4, 6, 8, 10]
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6>
```

Post-Lab Exercise:

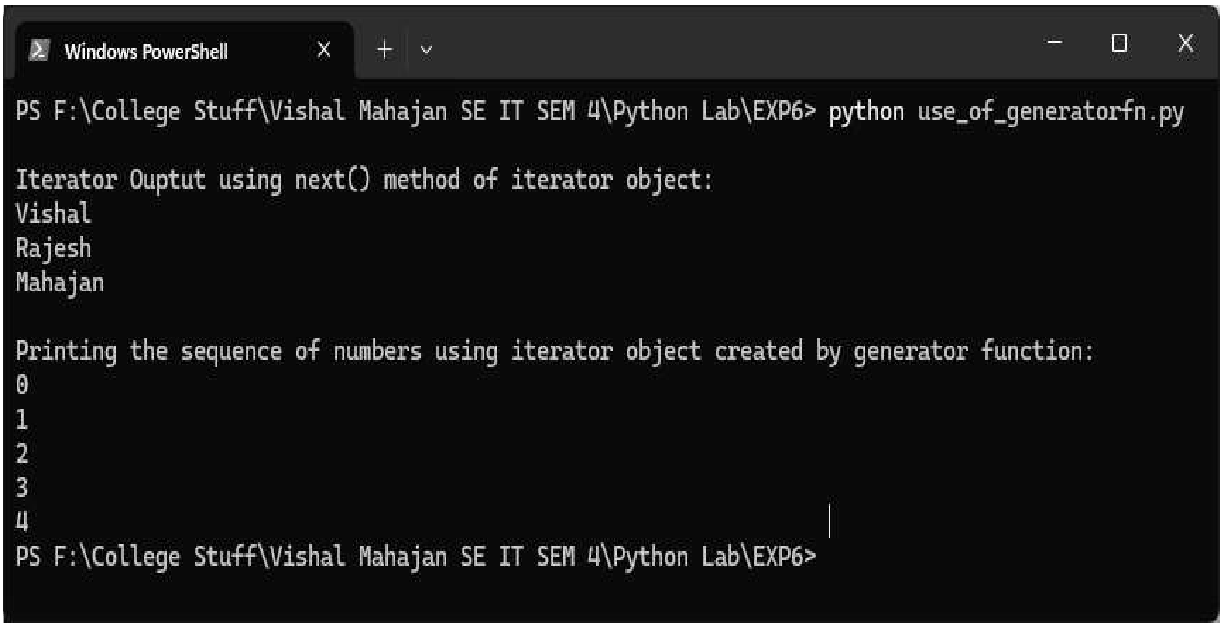
1) Write a Python program to demonstrate the use of iterator and generator functions

Code:

```
"""
Iterator: An iterator in Python is an object that can be iterated (looped)
upon. An object which will return data, one element at a time.
Generator: A generator in Python is a function that returns an iterator.
It looks like a normal function but contains yield statement.
yeild is used to return from a function without destroying the states of
its local variable and when the function is called, the execution starts
from the last yield statement.

"""

#Creating an iterator object using iter()
iterator = iter(["Vishal","Rajesh","Mahajan"])

#Iterator gives StopIteration error when it is exhausted so we use try and
except block to handle it
print("\nIterator Ouptut using next() method of iterator object: ")
while True:
    try:
        print(next(iterator))
    except StopIteration:
        break

#Generator Function to generate a sequence of numbers
def my_generator(start, end):
    while start < end:
        yield start
        start += 1
```

```
#Printing the sequence of numbers using iterator object created by generator function
print("\nPrinting the sequence of numbers using iterator object created by generator function: ")
for i in my_generator(0, 5):
    print(i)
```

Output:



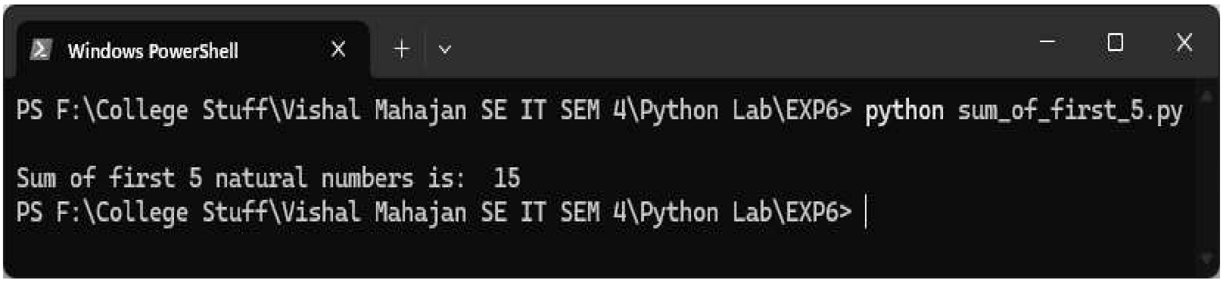
```
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> python use_of_generatorfn.py

Iterator Ouptut using next() method of iterator object:
Vishal
Rajesh
Mahajan

Printing the sequence of numbers using iterator object created by generator function:
0
1
2
3
4
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6>
```

2) Write a Python program to calculate sum of first 5 natural numbers using recursion.

Code:

```
def sum_of_first_5(n):
    if n==1:
        return 1
    else:
        return n+sum_of_first_5(n-1)

print("\nSum of first 5 natural numbers is: ",sum_of_first_5(5))
```

Output:



```
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> python sum_of_first_5.py

Sum of first 5 natural numbers is:  15
PS F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP6> 
```