Functional programming

- At the end of this class, students will be able to-
  - Understand the basic concept of functional programming
  - How to implement programs in a functional style.
  - Cover why you might want to incorporate functional programming in your own code.

PES University

Functional programming

Functional programming is a style of programming that is characterized by short functions, lack of statements, and little reliance on variables.

- map
- filter
- reduce
- max
- min
- zip

PES University

Map

•The map() function takes in an iterable (ie. list), and creates a new iterable object, a special map object.

•**map()** function returns a list of the results after applying the given function to each item of a given iterable (list, tuple etc.)

•**Basic syntax**

map(function_object, iterable1, iterable2,...)

map : (E → F) × Seq<E> → Seq<F>

PES University

list(map(len, [[1], [2], [3]]))	[1, 1, 1]
list(map(len, [1, 2, 3]))	error
list(map(len, ['1', '2', '3']))	[1, 1, 1]
map(lambda x: x.split(' '), ['a b c'])	[['a', 'b', 'c']]

PES University

PES UNIVERSITY

Filter

•The **filter()** function takes in an iterable, creates a new iterable object (again, a special map object), and a first-class function that must return a bool value.

• The new map object is a filtered iterable of all elements that returned True.

•**Basic syntax**

filter(function, sequence)

PES University

PES UNIVERSITY

```
# function that filters vowels
def fun(variable):
    letters = ['a', 'e', 'i', 'o', 'u']
    if (variable in letters):
        return True
    else:
        return False
# sequence
sequence = ['g', 'e', 'e', 'j', 'k', 's', 'p', 'r']
# using filter function
filtered = filter(fun, sequence)
print('The filtered letters are:')
for s in filtered:
    print(s)
```

Output:

```
The filtered letters are:
e
e
```

PES University

Reduce

•The reduce() function takes in an iterable, and then reduces the iterable to a single value.

•Reduce is different from filter() and map(), because reduce() takes in a function that has two input values.

This function is defined in “**functools**” module.

• **Basic syntax**

```
 from functools import reduce
reduce(function,sequence)
```

PES University

PES UNIVERSITY

```
# importing functools for reduce()
import functools
# initializing list
lis = [ 1 , 3, 5, 6, 2, ]
# using reduce to compute sum of list
print ("The sum of the list elements is : ",end="")
print (functools.reduce(lambda a,b : a+b,lis))

# using reduce to compute maximum element from list
print ("The maximum element of the list is : ",end="")
print (functools.reduce(lambda a,b : a if a > b else b,lis))
```

Output:

```
The sum of the list elements is : 17
The maximum element of the list is : 6
```

PES University

Max

- The max() method returns the largest element in an iterable or largest of two or more parameters.
- If the values are strings, an alphabetically comparison is done.

- **Basic syntax**

max(*n1, n2, n3, ...*)
Or:
max(*iterable*)

PES University

PES UNIVERSITY

Min

•The min() function returns the item with the lowest value, or the item with the lowest value in an iterable.
•If the values are strings, an alphabetically comparison is done.

•**Basic syntax**
min(*n1, n2, n3, ...*)
Or:
min(*iterable*)

PES University

PES UNIVERSITY

Zip

•The purpose of zip() is to **map the similar index of multiple containers** so that they can be used just using as single entity.

•**Basic syntax:**
zip(*iterators)

.

PES University

```
numberList = [1, 2, 3]
strList = ['one', 'two', 'three']
# No iterables are passed
result = zip()
print(result)
# Converting itertor to list
resultList = list(result)
print(resultList)
# Two iterables are passed
result = zip(numberList, strList)
print(result)
# Converting itertor to set
resultSet = set(result)
print(resultSet)
```

Output:

```
<zip object at 0x0000024EC9412088>
[]
<zip object at 0x0000024EC9476308>
{(3, 'three'), (2, 'two'), (1, 'one')}
```

PES University

Summary

- Functional programming was dealt in detail
- Functional programming gets its name from writing functions which provides the main source of logic in a program.

PES University