RAMAIAH
Institute of Technology

SUBJECT NAME: INTRODUCTION TO PYTHON PROGRAMMING

SUBJECT CODE: PLC142

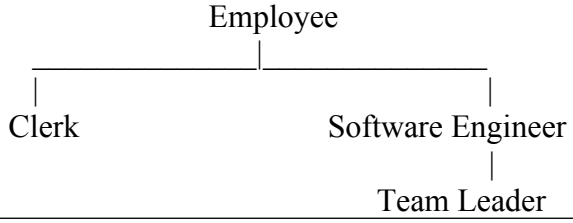
SEMESTER: I/II

TERM: 2022-2023

CREDITS: 2:0:1

LAB COMPONENT

SL. No.	QUESTIONS
1.	a) Write a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide) b) Write a python program to find the factorial of number using while loop. c) Write a python program to add 10 numbers by inputting each from the keyboard using for loop.
2.	a) Write a python function linearSearch() to read an array and search for the key element. Display the appropriate messages. Use the recursive function. b) Write a python program to define a function max_of_three() that takes three numbers as arguments and returns the largest of them using default arguments. c) Write a python program to define a function generate_n_chars() that takes an integer n and a character c and returns a string, n characters long. For example, generate_n_chars(5,"x") should return the string "xxxxx“ using keyword only parameters.
3.	a) Write a python program to implement a stack and queue using lists b) Write a python program to create a list of tuples having first element as the strings and the second element as the length of the string. Output the list of tuples sorted based on the length of the string. c) Write a python program to create a list and perform the following operations • Inserting an element • Removing an element • Appending an element • Displaying the length of the list • Popping an element • Clearing the list
4.	a) Write a tiny Python program numDict.py that makes a dictionary whose keys are the words ‘one’, ‘two’, ‘three’, and ‘four’, and whose corresponding values are the numerical equivalents, 1, 2, 3, and 4 (ints, not strings). b) Write a Python program to store PROFILE_DATA(user_id, name, DOB, qualification, work_experience) in a dictionary and pretty print the dictionary contents. (import pprint)
5.	a) Write a Python program to demonstrate built-in modules (Random,Time, Math, etc,) b) Create a user defined module using python to execute the following a) area of circle b) area of triangle c) area of rectangle.
6.	a) Write a python program to create a text file and ask the user to enter 5-6 lines of text. Display the longest and the shortest word from the file. Display the length of these words. b) Develop a python program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()].
7.	a) Develop a python program that uses class Student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details. [Hint: Use list to store the marks in three subjects and total marks. Use

	__init__() method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display() method to display the score card details.] b) Write a python program for the following: c) Create a class called time. Its three members all type int should be called hours, minutes and seconds. Write a python program that prompts the user to enter a time values separately. The Program should then store the time in the object and finally printout the total no of seconds represented by this value. Use appropriate member functions.
8.	a) Write a python program to create a class called Mylist that shadows a python list: it should overload + operator to append the data to the list. Also provide constructor for your class that takes an existing list. b) Write a python program to implement the following using Inheritance 
9.	a) Write a python program to Build a Number guessing game. When user enters not an integer raise the Exception and print total number of Guesses. b) Write a python function named DivExp which takes TWO parameters a, b and returns a value c (c=a/b). Write suitable assertion for a>0 in function DivExp and raise an exception for when b=0. Develop a suitable program which reads two values from the console and calls a function DivExp.

10.	a) Write a python program to implement the following using strings
	The third person singular verb form in English is distinguished by the suffix -s, which is added to the stem of the infinitive form: run -> runs. A simple set of rules can be given as follows:
	• If the verb ends in y, remove it and add ies
	• If the verb ends in o, ch, s, sh, x or z, add es
	• By default just add s
	• Test your function with words like try, brush, run and fix. Tip: Check out the string method endswith().
	b) Write a python function partition() that splits a list of soccer players into two groups. More precisely, it takes a list of first names (strings) as input and prints the names of those soccer players whose first name starts with a letter between and including A and M.
	>>> partition(['Eleanor', 'Evelyn', 'Sammy', 'Owen', 'Gavin'])
	Eleanor
	Evelyn
	Gavin
	>>> partition(['Xena', 'Sammy', 'Owen'])

Course outcome:

At the end of the course the student will be able to:

1. Demonstrate proficiency in handling loops and creation of functions
2. Identify the methods to create and manipulate lists, tuples and dictionaries.
3. Develop programs using modules and files.
4. Interpret the concepts of Object-Oriented Programming as used in Python.
5. Demonstrate the use of built-in functions for string processing and exception handling.

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE): 50 Marks		
Assessment Tool	**Marks**	**Course outcomes addressed**
Internal test-I	30	CO1, CO2,CO3
Internal test-II	30	CO3, CO4, CO5
Average of the two internal tests shall be taken for 30 marks.		
Other components		
Lab Component Evaluation	20	CO1, CO2, CO3, CO5, CO5
Semester-End Examination (SEE)	100	CO1, CO2, CO3, CO4, CO5

PYTHON PROGRAMS LAB-1

QUESTION 1

a) Write a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide)

```
#Python program-1 to perfrom arithmetic operations
print("**************** PROGRAM TO PERFROM ARITHMETIC OPERATIONS**********")
print("ENTER YOUR CHOICE\n1.ADD\n2.Subtract\n3.Multiply\n4.Divide\n")
a=int(input('Enter an integer: '))
b=int(input('Enter an integer: '))
c=int(input('Enter your choice operand: '))
if(c==1):
  d=a+b
  print("The sum is: ",d)
elif(c==2):
  e=a-b
  print("The diffrence is: ",e)
elif(c==3):
  f=a*b
  print("The product is: ",f)
elif(c==4):
  g=a//b
  print("The division is: ",g)
else:
  print("Invalid choice entered")
```

OUTPUT-

```
**************** PROGRAM TO PERFROM ARITHMETIC OPERATIONS**********
ENTER YOUR CHOICE
1.ADD
2.Subtract
3.Multiply
4.Divide

Enter an integer: 30
Enter an integer: 20
Enter your choice operand: 4
The division is:  1
```

b) Write a python program to find the factorial of number using while loop.

```
#Python program-2 find factorial of a number using while loop
a=int(input('Enter a number to find its factorial: '))
num=1
while(a>=1):
  num=num*a
```

```
  a=a-1
print('The Factorial of the number is:',num)
```

OUTPUT-

```
Enter a number to find its factorial: 5
The Factorial of the number is: 120
```

c)Write a python program to add 10 numbers by inputting each from the keyboard using for loop.

```
#Python program-3 to add 10 numbers from keyboard using for loop
sum=0
for i in range(10):
  num=int(input('Please Enter a Number: '))
  sum=sum+num
print('The Sum of The 10 Numbers is: ',sum)
```

OUTPUT-

```
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
Please Enter a Number: 10
The Sum of The 10 Numbers is:  100
```

QUESTION 2

a) Write a python function linearSearch() to read an array and search for the key element. Display the appropriate messages. Use the recursive function

```
#Python program-4 linear search using recursive function
def linear_search(arr,key,size):
  if(size==0):
    return -1
  elif(arr[size -1]==key):
    return size -1
  else:
    return linear_search(arr,key,size-1)
arr=[10,20,30,40,50]
key=20
size=len(arr)
ans=linear_search(arr,key,size)
if(ans!=-1):
  print('The element',key,'is found at',ans,'index of the given array')
else:
  print('The element',key,'is not found')
```

OUTPUT-

```
The element 20 is found at 1 index of the given array
```

b)Write a python program to define a function max_of_three() that takes three numbers as arguments and returns the largest of them using default arguments

```
#Python program -5 to return  largest of 3 numbers
def max_of_three(a,b,c):
  if(a>b and a>c):
    print(a,'Is The Largest Number')
    return
  elif(b>a and b>c):
    print(b,'Is The Largest Number')
    return
  else:
    print(c,'Is The Largest Number')
    return
a=int(input('Enter the value of a: '))
b=int(input('Enter the value of b: '))
c=int(input('Enter the value of c: '))
x=max_of_three(a,b,c)
print(x)
```

OUTPUT-

```
Enter the value of a: 500
Enter the value of b: 800
Enter the value of c: 600
800 Is The Largest Number
None
```

c)Write a python program to define a function generate_n_chars() that takes an integer n and a character c and returns a string, n characters long. For example, generate_n_chars(5,"x") should return the string "xxxxx“ using keyword only parameters.

```
#Python program-6 to generate n chars taking 2 parameters
def generate_n_chars(n,c,i):
  if(n==0):
    print('No output')
    return
  else:
    i=n*c
    print(i)
    return
c=input('Enter a character: ')
n=int(input('Enter number of times to generate character: '))
i=0
p=generate_n_chars(n,c,i)
print(p)
```

OUTPUT-

```
Enter a character: X
Enter number of times to generate character: 5
XXXXX
None
```

PYTHON PROGRAMS LAB-3

QUESTION 3

a) Write a python program to implement a stack and queue using lists

```
#Python Program-7- implement stack and queues using lists
stack=['Ronaldo','Messi','Neymar','Mbappe']
stack.append('Benzema')
stack.append('Bale')
print('********The Top Football Players In The World are-*********\n',stack)
print('-------------------------------------------------')
print(stack.pop())
print('-------------------------------------------------')
print(stack)
print('-------------------------------------------------')
print(stack.pop())
print('-------------------------------------------------')
print(stack)
# Implementing Queues using Lists
print()
print()
queue=['Pele','Maradona','Cafu','Zidane']
queue.append('Ronaldinho')
queue.append('Beckham')
print("******** The Top Football Legends are-**********\n",queue)
print('-------------------------------------------------')
print(queue.pop(0))
print('-------------------------------------------------')
print(queue)
print("-------------------------------------------------")
print(queue.pop(0))
print('-------------------------------------------------')
print(queue)
```

OUTPUT-

```
********The Top Football Players In The World are-*********
 ['Ronaldo', 'Messi', 'Neymar', 'Mbappe', 'Benzema', 'Bale']
-------------------------------------------------
Bale
-------------------------------------------------
['Ronaldo', 'Messi', 'Neymar', 'Mbappe', 'Benzema']
-------------------------------------------------
```

```
Benzema
------------------------------------------------
['Ronaldo', 'Messi', 'Neymar', 'Mbappe']

******** The Top Football Legends are-**********
 ['Pele', 'Maradona', 'Cafu', 'Zidane', 'Ronaldinho', 'Beckham']
------------------------------------------------
Pele
------------------------------------------------
['Maradona', 'Cafu', 'Zidane', 'Ronaldinho', 'Beckham']
------------------------------------------------
Maradona
------------------------------------------------
['Cafu', 'Zidane', 'Ronaldinho', 'Beckham'
```

b) Write a python program to create a list of tuples having first element as the strings and the second element as the length of the string. Output the list of tuples sorted based on the length of the string.

```
#Python Program-8-List of Tuples
list=[('Neymar',6),('Messi',5),('Ronaldo',7),('Mbappe',6),('Dani',4),('Lewandowski',11)]
list.sort(key=lambda a:a[1])
print(list)
```

OUTPUT-

```
[('Dani', 4), ('Messi', 5), ('Neymar', 6), ('Mbappe', 6), ('Ronaldo', 7), ('Lewandowski', 11)]
```

b) Write a python program to create a list and perform the following operations

- Inserting an element
- Removing an element
- Appending an element
- Displaying the length of the list
- Popping an element

- Clearing the list

```
#Python Program-9 - Creating and Doing operations on lists
list1=['Neymar','Messi','Ronaldo','Suarez','Kane','Hazard']
#To insert an element-
list1.insert(4,'Maguire')
print('Inserted List:\n',list1)
print()
#To remove an element
del list1[2]
print('Modified List:\n',list1)
print()
#To append an element
list1.append('Kante')
print('Appended List:\n',list1)
print()
#Displaying length of the list
a=len(list1)
print('Length of List: ',a)
print()
#Popping an Element
b=list1.pop()
print("List after Pop:\n",b)
print()
#Clearing the list
list1.clear()
print('Clearing the list:\n',list1)
```

OUTPUT-

```
Inserted List:
 ['Neymar', 'Messi', 'Ronaldo', 'Suarez', 'Maguire', 'Kane', 'Hazard']

Modified List:
 ['Neymar', 'Messi', 'Suarez', 'Maguire', 'Kane', 'Hazard']

Appended List:
 ['Neymar', 'Messi', 'Suarez', 'Maguire', 'Kane', 'Hazard', 'Kante']

Length of List:  7

List after Pop:
 Kante

Clearing the list:
 []
```

QUESTION 4

a) Write a tiny Python program numDict.py that makes a dictionary whose keys are the words 'one', 'two', 'three', and 'four', and whose corresponding values are the numerical equivalents, 1, 2, 3, and 4 (ints, not strings).

```
#Python Program-10- Python Dictionaries
numdict={'one':1,'two':2,'three':3,'four':4}
print("Displaying Dictionary-\n",numdict)
```

OUTPUT-

```
Displaying Dictionary-
 {'one': 1, 'two': 2, 'three': 3, 'four': 4}
```

b) Write a Python program to store PROFILE_DATA(user_id, name, DOB, qualification, work_experience) in a dictionary and pretty print the dictionary contents. (import pprint)

```
#Python Program 11- Python Dictionaries
import pprint
profile_data=[{'USER_ID':'001','NAME':'HAADI','DOB':'18/08/2003','QUALIFICATION':'
BE IN CS','WORK EXPERIENCE':'2 YEARS'},
        {'USER_ID':'002','NAME':'ADI','DOB':'10/04/2002','QUALIFICATION':'BE IN EC'
,'WORK EXPERIENCE':'1 YEAR'},
        {'USER_ID':'003','NAME':'Tiya','DOB':'01/01/2002','QUALIFICATION':'BE IN CS'
,'WORK EXPERIENCE':'3 YEARS'},
        {'USER_ID':'004','NAME':'MARIA','DOB':'15/05/2003','QUALIFICATION':'BE IN
EI','WORK EXPERIENCE':'1 YEAR'},
        {'USER_ID':'005','NAME':'BEN','DOB':'30/06/2000','QUALIFICATION':'BE IN EC'
,'WORK EXPERIENCE':'4 YEARS'}]

pretty_print=pprint.pprint(profile_data)
```

OUTPUT-

```
[{'DOB': '18/08/2003',
  'NAME': 'HAADI',
  'QUALIFICATION': 'BE IN CS',
  'USER_ID': '001',
  'WORK EXPERIENCE': '2 YEARS'},
```

```
{'DOB': '10/04/2002',
 'NAME': 'ADI',
 'QUALIFICATION': 'BE IN EC',
 'USER_ID': '002',
 'WORK EXPERIENCE': '1 YEAR'},
{'DOB': '01/01/2002',
 'NAME': 'Tiya',
 'QUALIFICATION': 'BE IN CS',
 'USER_ID': '003',
 'WORK EXPERIENCE': '3 YEARS'},
{'DOB': '15/05/2003',
 'NAME': 'MARIA',
 'QUALIFICATION': 'BE IN EI',
 'USER_ID': '004',
 'WORK EXPERIENCE': '1 YEAR'},
{'DOB': '30/06/2000',
 'NAME': 'BEN',
 'QUALIFICATION': 'BE IN EC',
 'USER_ID': '005',
 'WORK EXPERIENCE': '4 YEARS'}]
```

QUESTION 5

a) Write a Python program to demonstrate built-in modules (Random,Time, Math, etc,)

```
#Python Program 12- Python Modules
#Random Module
import random
dice1=[1,2,3,4,5,6]
print(random.choice(dice1))
print('----------------------')
#Math Module
import math
print(math.e)
print(math.pi)
print(math.tau)
print(math.inf)
print(math.nan)
print(math.asin(1.0))
print('----------------------')
#Time Module
import time
second=time.time()
print(second)
seconds=second
local_time=time.ctime(seconds)
print('local time: ',local_time)
print('printed immediately')
time.sleep(2.5)
print('printed after 2.5 seconds')
```

OUTPUT

```
5
----------------------
2.718281828459045
3.141592653589793
6.283185307179586
inf
nan
1.5707963267948966
----------------------
1674665371.5737932
local time:  Wed Jan 25 16:49:31 2023
printed immediately
printed after 2.5 seconds
```

b) Create a user defined module using python to execute the following a) area of circle

b) area of triangle c) area of rectangle.

```
# CREATING MODUE IN A NEW FILE(mymodule) AND THEN IMPORTING IT IN MA
IN FILE
import math

def areacircle(r):
  area=(math.pi*r*r)
  return round(area,2)
def arearectangle(a,b):
  area=a*b
  return area
def areatriangle(a,b):
  area=a*b/2
  return area
```

```
#Importing the module created in new file
import mymodule
print("On which geometric shape would you like to operate on?")
print("1. Rectangle")
print("2. Triangle")
print("3. Circle")
inp = int(input("Enter your choice(1-37) : "))
if(inp==1):
  a=int(input('Enter the value of length: '))
  b=int(input('Enter the value of breadth: '))
  ans=mymodule.arearectangle(a,b)
  print('The area of Rectangle is: ',ans,'units')
elif(inp==2):
  a=int(input('Enter the value of base: '))
  b=int(input('Enter the value of height: '))
  ans=mymodule.areatriangle(a,b)
  print('The area of the Triangle is: ',ans,'units')
elif(inp==3):
  r=int(input('Enter the value of radius: '))
  ans=mymodule.areacircle(r)
  print('The area of the circle is: ',ans,'units')
else:
  print('Invalid input ')
```

OUTPUT

```
On which geometric shape would you like to operate on?
1. Rectangle
2. Triangle
3. Circle
Enter your choice(1-3) : 1
Enter the value of length: 8
Enter the value of breadth: 7
```

```
The area of Rectangle is:  56 units


On which geometric shape would you like to operate on?
1. Rectangle
2. Triangle
3. Circle
Enter your choice(1-3) : 2
Enter the value of base: 15
Enter the value of height: 20

The area of the Triangle is:  150.0 units


On which geometric shape would you like to operate on?
1. Rectangle
2. Triangle
3. Circle
Enter your choice(1-3) : 3
Enter the value of radius: 11
    The area of the circle is:  380.13 units


On which geometric shape would you like to operate on?
1. Rectangle
2. Triangle
3. Circle
Enter your choice(1-3) : 4
    Invalid input
```

PYTHON PROGRAMS LAB-6

QUESTION 6:

a) Write a python program to create a text file and ask the user to enter 5-6 lines of text. Display the longest and the shortest word from the file. Display the length of these words.

```
  def smallest_largest_words(str1):

word = "";
all_words = [];
str1 = str1 + " ";
for i in range(0, len(str1)):
  if(str1[i] != ' '):
    word = word + str1[i];
  else:
    all_words.append(word);
    word = "";

small = large = all_words[0];

#Find smallest and largest word in the str1
for k in range(0, len(all_words)):
  if(len(small) > len(all_words[k])):
      small = all_words[k];
  if(len(large) < len(all_words[k])):
     large = all_words[k];
     return small,large;
myfile=open("test.txt","w")
line=input("Enter text:")
myfile.write(line)
myfile.close()
myfile=open("test.txt","r")
lines=myfile.read()
#lines=lines.split()
small, large = smallest_largest_words(lines)
print("Smallest word: " + small);
print("Length of the shortest word=",len(small))
print("Largest word: " + large);
print("Length of the longest word=",len(large))
```

OUTPUT-

```
Enter text:Hi Haadi
Smallest word: Hi
Length of the shortest word= 2
Largest word: Haadi
Length of the longest word= 5
```

b) Develop a python program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()].

```
def sorting(filename):
        infile=open(filename)
        words=[]
        for line in infile:
            temp=line.split()
           for i in temp:
                words.append(i)
        infile.close()
        words.sort()
        outline=open("result.txt","w")
        for i in words:
                outline.writelines(i)
                outline.writelines(" ")
                 outline.close()
sorting("sample.txt")
```

OUTPUT

INPUT FILE:

Hi all

Good Morning

I hope you all are doing good

Welcome to python Laboratary …

OUTPUT FILE:

… Good Hi I Laboratory Morning python Welcome all all are doing good hope to you

QUESTION 7

a) Develop a python program that uses class Student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details. [Hint: Use list to store the marks in three subjects and total marks. Use __init__() method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display() method to display the score card details.]

```
from  functools import reduce
class Student:
    def __init__(self, name, USN):
        self.name = name
        self.USN = USN
        self.list = []

    def getMarks(self) -> None:
        print("Enter the marks of the student: ")
        for i in range(0, 3):
            self.list.append(int(input()))

    def display(self):
        max_marks = int(input("enter max marks"))
        TotalMarks = reduce(lambda a, b: a + b, self.list)
        percentage = (TotalMarks / max_marks) * 100
        print("Name:", self.name)
        print("USN", self.USN)
        print("Marks: ", *self.list, sep="\n")
        print("max marks", max_marks)
        print("obtained marks", TotalMarks)
        print("percentage obtained", percentage)


print("Enter the details of student 1")
name = input("Enter the name: ")
USN = input("Enter the usn: ")
s1 = Student(name, USN)
s1.getMarks()
print("\n")
s1.display()
```

OUTPUT

```
Enter the details of student 1
Enter the name: Haadi
Enter the usn: 1ms22ii11
Enter the marks of the student:
89
```

```
90
91


enter max marks300
Name: Haadi
USN 1ms22ii11
Marks:
89
90
91
max marks 300
obtained marks 270
percentage obtained 90.0
```

b) Create a class called time. Its three members all type int should be called hours, minutes and seconds. Write a python program that prompts the user to enter a time values separately. The Program should then store the time in the object and finally printout the total no of seconds represented by this value. Use appropriate member functions.

```
class Time:
    def __init__(self, hours=0, minutes=0, seconds=0):
        self.hours = hours
        self.minutes = minutes
        self.seconds = seconds

    def to_seconds(self):
        return self.hours * 3600 + self.minutes * 60 + self.seconds


time = Time(hours=int(input("Enter the hours: ")),
            minutes=int(input("Enter the minutes: ")),
            seconds=int(input("Enter the seconds: ")))
print("Total seconds:", time.to_seconds())
```

OUTPUT

```
Enter the hours: 5
Enter the minutes: 10
Enter the seconds: 20
Total seconds: 18620
```

PYTHON PROGRAMS LAB-8

QUESTION 8

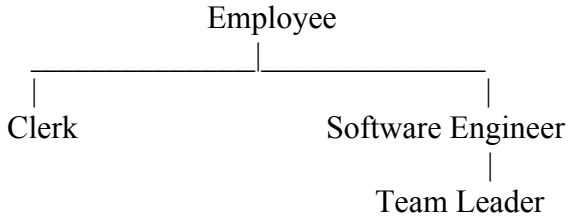
a) Write a python program to create a class called Mylist that shadows a python list: it should overload + operator to append the data to the list. Also provide constructor for your class that takes an existing list.

```
class Mylist:
    def __init__(self, lst=None):
        if lst is None:
            self.items = []
        else:
            self.items = lst

    def __add__(self, other):
        if isinstance(other, list):
            return self.items.extend(other)
        elif isinstance(other, (int, str)):
            return self.items.append(other)
        else:
            return NotImplemented

    def __repr__(self):
        return repr(self.items)

my_list = Mylist([1, 2, 3])
my_list + [4, 5, 6]
print(my_list)
my_list + 7
print(my_list)
```

OUTPUT

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6, 7]
```

b) Write a python program to implement the following using Inheritance




```
class Employee:


    def __init__(self, name, salary):
        self.name = name
        self.salary = salary


    def get_details(self):
        print("from Employee ->")
        return "Name: " + self.name + ", Salary: " + str(self.salary)


class Clerk(Employee):
    print("from clerk->")
    pass


class SoftwareEngineer(Employee):
    def __init__(self, name, salary, programming_languages):
        super().__init__(name, salary)
        self.programming_languages = programming_languages

    def get_details(self):
        print("from software engineer->")
        return super().get_details() + ", Programming Languages: " + self.programming_languag
es


class TeamLeader(SoftwareEngineer):
    def __init__(self, name, salary, programming_languages, team_size):
        super().__init__(name, salary, programming_languages)
        self.team_size = team_size

    def get_details(self):
        print("Team leader->")
        return super().get_details() + ", Team Size: " + str(self.team_size)
```

```
e1 = Employee("John", 5000)
print(e1.get_details())

c1 = Clerk("Jane", 3000)
print(c1.get_details())

se1 = SoftwareEngineer("Mike", 6000, "Python, Java")
print(se1.get_details())

tl1 = TeamLeader("Steve", 8000, "Python, Java, C++", 10)
print(tl1.get_details())
```

OUTPUT

```
from clerk->
from Employee ->
Name: John, Salary: 5000
from Employee ->
Name: Jane, Salary: 3000
from software engineer->
from Employee ->
Name: Mike, Salary: 6000, Programming Languages: Python, Java
Team leader->
from software engineer->
from Employee ->
Name: Steve, Salary: 8000, Programming Languages: Python, Java, C++, Team Size: 10
```

PYTHON LAB PROGRAMS- 9

QUESTION 9

a) Write a python program to Build a Number guessing game. When user enters not an integer raise the Exception and print total number of Guesses.

```
#Write a python program to Build a Number guessing game.
#When user enters not an integer raise the Exception and print total number of Guesses.

from pickle import TRUE
import random
t = 0
g = int(input("Total Guesses: "))
low = int(input("Enter the lower range: "))
high = int(input("Enter the upper range: "))
x = random.randint(low, high)
n = int(input("Enter an integer between the given range: "))



while (x != 'n'):
    if(t<(g-1)):
        if n < x:
            print("The number guessed is low")
            t = t+1
            n = int(input("Enter an integer between the given range: "))
        elif (n > x):
            print("The number guessed is high")
            t = t+1
            n = int(input("Enter an integer between the given range: "))
        else:
            print("The number guessed is right")
            print("Total guesses taken: ", t+1)
            break
    else:
        print("Ran out of tries!")
        break
```

OUTPUT

```
Total Guesses: 3
Enter the lower range: 22
Enter the upper range: 26
Enter an integer between the given range: 24
The number guessed is right
Total guesses taken:  1
```

b) Write a python function named DivExp which takes TWO parameters a, b and returns a value c (c=a/b). Write suitable assertion for a>0 in function DivExp and raise an exception for when b=0. Develop a suitable program which reads two values from the console and calls a function DivExp.

```
#Write a python function named DivExp which takes TWO parameters a, b and returns a value c (c=a/b).
#Write suitable assertion for a>0 in function DivExp and raise an exception for when b=0.
#Develop a suitable program which reads two values from the console and calls a function DivExp.
def DivExp(a,b):
  assert a>0

  try:

   c = a/b
   print("Quotient is", result)

  except ZeroDivisionError:
    print("Error: Denominator cannot be 0.")
    quit()

a=int(input("enter the numerator"))
b=int(input("enter the denominator"))
DivExp(a,b)
```

OUTPUT

```
enter the numerator25
enter the denominator0
Error: Denominator cannot be 0.
```

PYTHON LAB PROGRAMS – 10

QUESTION 10

a) Write a python program to implement the following using strings

The third person singular verb form in English is distinguished by the suffix -s, which is added to the stem of the infinitive form: run -> runs. A simple set of rules can be given as follows:

- If the verb ends in y, remove it and add ies
- If the verb ends in o, ch, s, sh, x or z, add es
- By default just add s
- Test your function with words like try, brush, run and fix. Tip: Check out the string method endswith().

```
import re


def make_3sg_form(verb):
    es = ('o', 'ch', 's', 'sh', 'x', 'z')

  if verb.endswith('y'):
      return re.sub('y$', 'ies', verb)
  elif verb.endswith(es):
      return verb + 'es'
  else:
      return verb + 's'


if __name__ == "__main__":
    print (make_3sg_form('try'))
    print (make_3sg_form('brush'))
    print (make_3sg_form('run'))
    print (make_3sg_form('fix'))
```

OUTPUT

```
tries
brushes
runs
fixes
```

b.Write a python function partition() that splits a list of soccer players into two groups. More precisely, it takes a list of first names (strings) as input and prints the names of those soccer players whose first name starts with a letter between and including A and M.

```
>>> partition(['Eleanor', 'Evelyn', 'Sammy', 'Owen', 'Gavin'])
Eleanor
Evelyn
Gavin
>>> partition(['Xena', 'Sammy', 'Owen'])
```

```
def partition(lst):
  for player in lst:
    if player[0] >= 'A' and player[0] <= 'M':
      print(player)



partition(['Eleanor', 'Evelyn', 'Sammy', 'Owen', 'Gavin'])
```

OUTPUT

```
Eleanor
Evelyn
Gavin
```