Task

**Course**: Artificial Intelligence

(Machine Learning & Deep Learning)

Name: Abbas Shafi

Father Name: Dr Muhammad Shafi

Week: 04

Submitted to: Sir Syed Nazir Afridi



National Vocational & Technical Training Commission

National Center For Big Data & Cloud Computing

University of Engineering & Technology

Peshawar

**LAB NO. 1 Recursion**

## Python Programming

**Introduction:**

Python programming is general purpose object oriented dynamic programming language used mainly in AI, ML, DL and data science. This language is known for their easiness and high level language. Contain large number of libraries through which programmer can easily handle tasks. **Tool used**:

* Jupyter Notebook
* Annaconda

**Task 1**: Write python program to find sum of natural number using recursion

## Code:

#t1

n=int(input("enter number ")) def sum(n):

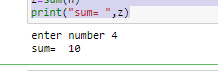
if(n==1): return 1

else:

return (n+ sum(n-1)) z=sum(n)

print("sum= ",z)

## Output:

****

**Task 2:**Write a program to reverse a string using recursion

### Code:

#t2

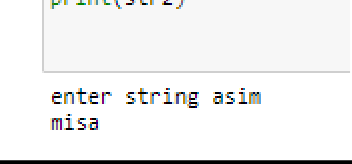
str=input("enter string ") def reverseorder(str):

#str="asim" if(str[3]):

return str[::-1] else:

return reverseorder(str)-1 str2=reverseorder(str) print(str2)

## Output:



**Task 3:** Find x^y using recursion

## Code:

#t3

x=int(input("enter number")) y=int(input("enter number")) def power(x,y):

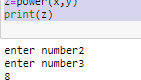
if(y==0):

return 1 else:

return(x \* power(x,y-1)) z=power(x,y)

print(z)

## Output:



**Task 4:**Write python program to find factoraial of number using recursion.

## Code:

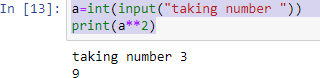
#t4

n=int(input("enter num")) def factorial(n):

if(n==0): return 1

else:

return(n \* factorial(n-1)) fact=factorial(n) print("factorial = ",fact) **Output:**



## Conclusion:

In this lab we learned basic of recursions of python programming and how to implement them. All the tasks were performed and submitted to the instructor.

# LAB NO. 2 OOP

**Task 1**: Write a calculator capable of finding square , cube , square root

## Code:

#t2

from math import \* class Calculator:

def init (self,square,cube,squareroot): square=square\*\*2

cube=cube\*\*3 squareroot=sqrt(squareroot) self.square=square self.cube=cube self.squareroot=squareroot

def getInfo(self):

print(f"square is {self.square} cube is {self.cube} squareroot is

{self.squareroot}")

cal=Calculator(2,2,4) cal.getInfo()

## Output:

****

**Task 2:**Use static method in problem 2 to greet user with word “Heloo welcome to calculator”

### Code:

#t3

from math import \* class Calculator:

def init (self,square,cube,squareroot): square=square\*\*2

cube=cube\*\*3 squareroot=sqrt(squareroot) self.square=square self.cube=cube self.squareroot=squareroot

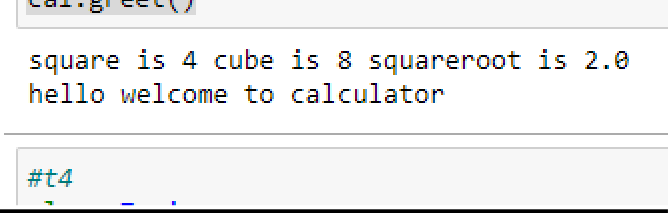
def getInfo(self):

print(f"square is {self.square} cube is {self.cube} squareroot is {self.squareroot}") @staticmethod

def greet():

print("hello welcome to calculator") cal=Calculator(2,2,4)

cal.getInfo() cal.greet() **Output:**



**Task 3:** Write a class train which has method to book a train ticket, get number of seats and get fare information

Code:

class Train:

def init (self,bookTrain,name,departure\_time,seats,seatprice): self.bookTrain=bookTrain

print("Train booking ")

def bookSeats(self): self.bookTrain=seats print(seats)

def Seatsprice(self): self.bookTrain=seatprice print(seatprice)

def personbooktrain(self): self.bookTrain=name self.bookTrain=departure\_time print(name) print(departure\_time)

def getInfo(self):

print(f"{self.name} book train on {self.departure\_time} of total seats of {self.seats} on this {self.seatprice}")

#name=input("Name ") #departure\_time=input("on this time") #seats=int(input("of this seats")) #seatprice=int(input("on each seat price of ")) bktrain=Train("asim","9pm",40,250) bktrain.bookSeats()

bktrain.Seatsprice() bktrain.personbooktrain() bktrain.getInfo()

## Output:

**Conclusion:**

In this lab we learned basic of OOP, OBJECTS , CLASSES of python programming and how to implement them. All the tasks were performed and submitted to the instructor.

# LAB NO. 3 Exception Handling

**Task 1**: Write a program to handle type error

## Code:

try:

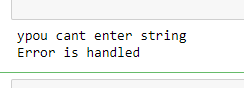
c="2"+2

except TypeError:

print("ypou cant enter string") finally:

print("Error is handled")

## Output:

****

**Task 2:** Write a program to handle zerodivision error

## Code:

a=10

b=0 try:

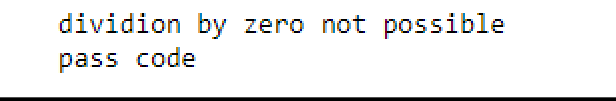
c=a/b

except ZeroDivisionError: print("dividion by zero not possible")

finally:

print("pass code")

## Output:



**Task 3:**

## Code:

try:

n=int(input("enter a number")) c=n+5

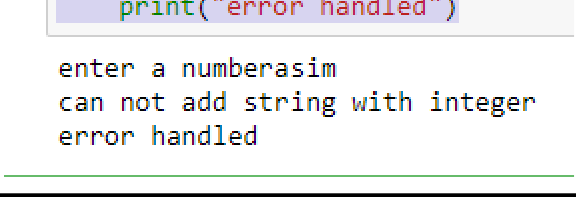
except ValueError:

print("can not add string with integer") else:

print(c) finally:

print("error handled")

## Output:

****

**Conclusion:**

In this lab we learned basic of exception handling and different types of errors in python programming and how to implement them. All the tasks were performed and submitted to the instructor.