PYTHON PRATICALS

1.CONTROL STRUCTURE OF PYTHON

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

b=int(input('enter b : '))

if a>b:

print('a greater')

elif b>a:

print('b greater')

else:

print('equals')

for i in range(1,n+1):

if i==1:

pass

if i==2:

continue

if i==3:

break

print(i)

2.List,Dictionary and Tuple

l=[]

for i in range(5):

x=int(input())

l.append(x)

print(l)

l.pop()

print(l)

t = tuple(l)

print(t)

d={}

for i in range(5):

k=int(input("key : "))

v=input('value : ')

d.update({k:v})

print(d)

d[6]=’six’

print(d)

del d

del l

del t

3.concept of functions ,scoping,recursion,and list mutability

n=int(input('enter no : '))

def fact(n):

if n<=1:

return 1

else:

return n\*fact(n-1)

print(fact(n))

4.object oriented programming

class mca():

def \_\_init\_\_(self,age):

self.age = age

def disp(self,name):

return "i'm {} at age {} ".format(name,self.age)

age=int(input('enter age : '))

name = input("enter name : ")

m = mca(age)

print(m.disp(name))

print(m.age)

5.exception handling

ls=[]

for i in range(5):

x=int(input())

ls.append(x)

for i in ls:

try :

div= 10/i

except Exception as e:

print(e)

finally:

print(div)

6.Armstrong No

no = input('enter no : ')

sum=0

for i in range(len(no)):

x = int(no[i])\*\*3

sum = sum + x

if int(no)== sum:

print('armstrong')

else:

print('not armstrong')

7.Factorial no

n=int(input())

fact=1

for i in range(1,n+1):

fact = fact\*i

print(fact)

8.prime no

n=int(input())

def prime(n):

for i in range(2,n):

if n<=1:

return 1

if n%i==0:

return 1

return 0

if prime(n):

print('not prime')

else:

print('prime')

9.Calculator

a = int(input(' a : '))

b = int(input(' b : '))

while(True):

print("enter 1/2/3/4 or 0")

ch=int(input('ch : '))

if ch==1:

print(a+b)

elif ch==2:

print(a-b)

elif ch==3:

print(a\*b)

elif ch==4:

print(a//b)

elif ch==0:

break

else:

print('wrong ch')

10.Lambda Function

sqr = lambda x:x\*\*2

x=int(input('enter no : '))

print(x)

x=sqr(x)

print(x)

ls=[1,2,3,4,5]

print(ls)

res = list(filter(lambda x:x%2==0,ls))

print(res)

11.Binary Function

# already sorted list

ls=[1,2,3,4,5,6,7,8,9,10]

s=int(input('Search : '))

def binary(ls,s):

l=0

h=len(ls)-1

while l<=h:

m=(l+h)//2

if ls[m]==s:

return True

else:

if ls[m]<s:

l=m+1

else:

h=m-1

return False

if binary(ls,s):

print('found')

else:

print('not found')

Regular expression

1. Develop programs to learn regular expressions using python.

import re

pattern = r"ABCD"

if re.match(pattern,"ABCD"):

print("match")

else:

print("no match")

# Program to show the use of lambda functions  
double = lambda x: x \* 2  
print(double(5))  
  
# Program to filter out only the even items from a list  
my\_list = [1, 5, 4, 6, 8, 11, 3, 12]  
new\_list = list(filter(lambda x: (x%2 == 0) , my\_list))  
print(new\_list)  
  
# Program to double each item in a list using map()  
my\_list = [1, 5, 4, 6, 8, 11, 3, 12]  
new\_list = list(map(lambda x: x \* 2 , my\_list))

from tkinter import \*

class ImageDemo(Frame):

def \_\_init\_\_(self):

Frame.\_\_init\_\_(self)

self.master.title("Image Demo")

self.grid()

self.\_image=PhotoImage(file="1dpmw.gif")

self.\_imageLabel=Label(self,image=self.\_image)

self.\_imageLabel.grid()

self.\_textLabel=Label(self,text="umesh")

self.\_textLabel.grid()

def main():

ImageDemo() .mainloop()

main()