Lab Manual

Introduction to Python Programming – BPLCK103
1st Semester

1. a. Develop a program to read the student details like Name, USN, and Marks in three subjects. Display the student details, total marks and percentage with suitable messages.

```
print("Enter Student details: ")
Name=input("Enter a Name: ")
USN=input("Enter USN: ")
sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))

total=sub1+sub2+sub3
Per=(total/300)*100

print("Student details: ")
print("Name: ",Name)
print("USN:  ",USN)
print("Total Marks: ",total)
print("Percentage: ",Per)
```

b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.

```
from datetime import
 date def cal_age(bday):
    today=date.today()
      age=today.year - bday.year -((today.month,today.day)<(bday.month,bday.day))
      return age
Name=input("Enter name of a person: ")
d=int(input("Enter day(dd) of birth: "))
m=int(input("Enter month(mm) of birth: "))
y=int(input("Enter year(yyyy) of birth: "))

age=cal_age(date(y,m,d)) if
age<0:
      print("Enter Valid Date") elif
age>=60:
   print("The person is Senior citizen") else:
   print("The person is not Senior citizen")
```

2. a. Develop a program to generate Fibonacci sequence of length (N). Read N from the console.

```
N = int(input("Enter N value: ")) n1,

n2 = 0, 1
count = 0

if N <= 0:
    print("Please enter a positive integer") elif
N == 1:
    print("Fibonacci sequence upto",N,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < N:
        print(n1)
        nxt = n1 + n2
        n1 = n2
        n2 = nxt
        count += 1              # count++
```

b. Write a function to calculate factorial of a number. Develop a program to compute binomial coefficient (Given N and R).

```
def factorial(n): if n
    == 0:
        return 1
    else:
        return n * factorial(n-1)

N = int(input("Enter a value for N: ")) R
= int(input("Enter a value for R: "))

if R == 1 or R == N: print(1)
elif R > N:
    print(0)
else:
    b_coef=factorial(N)/( factorial(N-R) * factorial(R) ) print(b_coef)
```

3. Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages.

```
import math

n=int(input("Enter N value:"))
Mylist=[]
print("Enter a value") for
i in range(n):
    v=int(input())
    Mylist.append(v)
lenv=len(Mylist)
if(lenv == 0):
    print("List should not be empty") else:
    sumv=sum(Mylist)
    meanval = sumv/lenv
    temp=0
    for i in range(lenv):
        temp += ((Mylist[i] - meanval) * (Mylist[i] - meanval)) vari =
    temp / lenv;
    sd = math.sqrt(vari) print("Mean
    value: ",meanval) print("Varience:
    ",vari) print("Standard Deviation:
    ",sd)
```

4. Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with suitable message.

```
str1 = input("Enter a multi-digit number: ") d
= dict()
for c in str1:
      if c in d.keys(): d[c] =
            d[c] + 1
      else:
            d[c] = 1
for k,v in d.items():
      print("Number ",k," has frequency: ",v)
```

5. Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items]

```
fname = input('Enter the file name: ') try:
    fhand = open(fname,"r")
    counts = dict()
    for line in fhand:
        words = line.split() for
        word in words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
    counts=sorted(counts,reverse=True)
    print(counts[:11])

except:
    print("File can't open")
```

6. Develop a 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()].

```
infile = input("Enter input file name: ")
fh_in = open(infile,"r")
data=fh_in.readlines()
lst = []
for i in range(len(data)): lst.append(data[i].strip())
lst.sort()
fh_in.close()
outfile=input("Enter output file name: ")
fh_out = open(outfile, "w")
for i in lst:
    fh_out.write(i)
    fh_out.write("\n")
fh_out.close()
```

7. Develop a program to backing Up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.

```
import os
import zipfile
fold=input("Enter Directory name: ") #zipf
= fold +".zip"
#zf  =  zipfile.ZipFile(zipf,  "w")
zf = zipfile.ZipFile("myzipfile.zip", "w")  for
dirname, subdirs, files in os.walk(fold):
      zf.write(dirname)
      for filename in files: zf.write(os.path.join(dirname,
            filename))
zf.close()
```

8. Write a 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.

```
import sys
def DivExp(a,b): try:
        assert a>0, "Expected number should be positive"
        assert b!=0, "Can't divide by zero"
        c=a/b
        return c
    except AssertionError as msg:
        sys.exit(msg)

v1=int(input("Enter first number: ")) v2=int(input("Enter
second number: ")) res=DivExp(v1,v2)
print(v1,"%",v2,"  is:  ",res)
```

9. Define a function which takes TWO objects representing complex numbers and returns new complex number with a addition of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition of N complex numbers.

```
class Complex():
    def _init____(self,r,i):
        self.real = r self.imag
        = i

    def addcom(self, other):
        return Complex(self.real + other.real, self.imag + other.imag)

N=int(input("Enter a number of complex numbers: "))
sum=Complex(0,0)
for i in range(N): print("\nComplex
    Number:",i+1)
    a=int(input("Enter real part: ")) b=int(input("Enter
    imaginary part: ")) c=Complex(a,b)
    sum=sum.addcom(c)

print("\nAfter Sum:",sum.real,"+",sum.imag,"i")
```

10. Develop a 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.]

```
class Student:
    def __init__(self):
        self.USN=input("Enter Student USN: ")
        self.Name=input("Enter student name: ")
        self.Marks=self.getMarks()

    def getMarks(self):
        mrk=[]
        print("Enter the marks of subjects out of 100") for
        i in range(3):
            print("Subject",i+1)
            m=int(input())
            mrk.append(m)
        mrk.append(sum(mrk))
        return mrk

    def display(self): print("\nName:
        ",self.Name) print("USN:
        ",self.USN)
        print("-----------------------------------------")
        print("Subject 1 Marks: ",self.Marks[0])
        print("Subject 2 Marks: ",self.Marks[1])
        print("Subject 3 Marks: ",self.Marks[2])
        print("============================")
        print("              Total:   ",self.Marks[3])
        per=(self.Marks[3]/300)*100
        print("-----------------------------------------")
        print("      percentage: ",per)

print("Enter student Details: ") s=Student()
s.display()
```