Programming Exercises:

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.

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.

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

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

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

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

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]

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()].

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.

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.

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.

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.]

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.

```
# backupToZip.py - Copies an entire folder and its contents into
# a ZIP file whose filename increments.
import zipfile, os
def backupToZip(folder):
    folder = os.path.abspath(folder)
    number = 1
    while True:
        zipFilename = os.path.basename(folder) + '_' + str(number) +'.zip'
        if not os.path.exists(zipFilename):
            break
        number = number + 1
    print(f'Creating {zipFilename}...')
    backupZip = zipfile.ZipFile(zipFilename, 'w')
    for foldername, subfolders, filenames in os.walk(folder):
```

```
        print(f'Adding files in {foldername}...')
        backupZip.write(foldername)
        for filename in filenames:
            backupZip.write(os.path.join(foldername, filename))
    backupZip.close()
    print('Done.')
backupToZip('.\\delicious')
```

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.

```
>>> a=5+3j
>>> b=4+6.6j
>>> a+b
(9+9.6j)
>>> type(a+b)
<class 'complex'>
>>> type(a)
<class 'complex'>
>>> type(b)
<class 'complex'>
```

```
# 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, real=0.0, imaginary=0.0):
        self.real = real;
        self.imaginary = imaginary;
    def __str__(self):
        return 'Sum of complex number :'+str(self.real)+ '+ i'+ str(self.imaginary)
    def __add__(self,other):
        self.real = self.real + other.real;
        self.imaginary = self.imaginary + other.imaginary;
        return self;
c3 = Complex();
n=int(input('Enter number of complex numbers to be added '))
for i in range(n):
    c1= c3
    print('Enter Complex Number ',i+1)
    r=float(input('enter real '))
    i=float(input('enter imaginary'))
    c2 = Complex(r,i)
    c3 = c1+c2
print(c3)
```

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.]

```
# 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,n,u):
    self.name = n
    self.usn = u
    self.marks = []



  def getMarks(self):
      total=0
      for i in range(3):
        print('Enter the marks of subject:',i+1)
        m = int(input())
        total=total+m
        self.marks.append(m)
      self.marks.append(total)
      self.marks.append(total/3)


  def display(self):
    print('SCORE CARD OF STUDENT')
```

```
        print('---------------------------------------')
        print('NAME : ', self.name.upper())
        print('USN : ', self.usn.upper())
        for i in range(3):
            print ('Marks in subject',i+1,':',self.marks[i])
        print ('Total :', self.marks[3])
        print ('Percentage :',self.marks[4])

name = input('Enter the name of Student:')
usn= input('Enter the USN of Student:')
s1 = student(name,usn)
s1.getMarks()
s1.display()
```