# Introduction to Python Programming Lab Manual for MCA/M.Sc

## Question 1

### a. Student Details

**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.**

**Python Code**

```
stName = input("Enter the name of the student : ")
stregno = input("Enter the Register Number of the student
: ")
stMarks1 = int(input("Enter marks in Subject 1 : "))
stMarks2 = int(input("Enter marks in Subject 2 : "))
stMarks3 = int(input("Enter marks in Subject 3 : "))


print("Student Details\n========================")
print("%12s"%("Name :"), stName)
print("%12s"%("Reg.No :"), stregno)
print("%12s"%("Marks 1 :"), stMarks1)
print("%12s"%("Marks 2 :"), stMarks2)
print("%12s"%("Marks 3 :"), stMarks3)
print("%12s"%("Total :"), stMarks1+stMarks2+stMarks3)
print("%12s"%("Percent :"), "%.2f"%((stMarks1+stMarks2+stMarks3)/3))
print("========================")
```

**Output**

```
Enter the name of the student : RAMESH
Enter the Register Number  of the student :
21MCA001
Enter marks in Subject 1 : 87
Enter marks in Subject 2 : 78
Enter marks in Subject 3 : 65
Student Details
========================
      Name : RAMESH
    Reg.No : 21MCA001
   Marks 1 : 87
   Marks 2 : 78
   Marks 3 : 65
     Total : 230
   Percent : 76.67
```

### b. Senior Citizen Check

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

**Python Code**

```
from datetime import date


perName = input("Enter the name of the person : ")
perDOB = int(input("Enter his year of birth : "))

curYear = date.today().year
perAge = curYear - perDOB

if (perAge > 60):
    print(perName, "aged", perAge, "years is a Senior Citizen.")
else:
    print(perName, "aged", perAge, "years is not a Senior Citizen.")
```

**Output**

```
Enter the name of the person : Akbar Khan
Enter his year of birth : 1978
Akbar Khan aged 44 years is not a Senior Citizen.

Enter the name of the person : George Best
Enter his year of birth : 1957
George Best aged 65 years is a Senior Citizen.
```

# Question 2

### a. Fibonacci Sequence

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

**Python Code**

```
num = int(input("Enter the Fibonacci sequence length to be generated : "))

firstTerm = 0
secondTerm = 1
print("The Fibonacci series with", num, "terms is :")
print(firstTerm, secondTerm, end=" ")
for i in range(2,num):
    curTerm = firstTerm + secondTerm
    print(curTerm, end=" ")
    firstTerm = secondTerm
    secondTerm = curTerm
```

### Output

```
Enter the Fibonacci sequence length to be generated : 8
The Fibonacci series with 8 terms is :
0 1 1 2 3 5 8 13


Enter the Fibonacci sequence length to be generated : 5
The Fibonacci series with 5 terms is :
0 1 1 2 3
```

## b. Factorial & Binomial Coefficient

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

### Python Code

```
def fact(num):
    if num == 0:
        return 1
    else:
        return num * fact(num-1)


n = int(input("Enter the value of N : "))
r = int(input("Enter the value of R (R cannot be negative or greater than
N): "))
nCr = fact(n)/(fact(r)*fact(n-r))

print(n,'C',r," = ","%d"%nCr,sep="")
```

### Output

```
Enter the value of N : 7
Enter the value of R (R cannot be negative or greater than N): 5
7C5 = 21

Enter the value of N : 5
Enter the value of R (R cannot be negative or greater than N): 5
5C5 = 1

Enter the value of N : 3
Enter the value of R (R cannot be negative or greater than N): 1
3C1 = 3

Enter the value of N : 8
Enter the value of R (R cannot be negative or greater than N): 0
8C0 = 1
```

# Question 3

## Mean, Variance and Standard Deviation

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

### Python Code

```
from math import sqrt

myList = []

num = int(input("Enter the number of elements in your list : "))

for i in range(num):
    val = int(input("Enter the element : "))
    myList.append(val)

print('The length of list1 is', len(myList))

print('List Contents', myList)


total = 0
for elem in myList:
    total += elem

mean = total / num

total = 0
for elem in myList:
    total += (elem - mean) * (elem - mean)

variance = total / num

stdDev = sqrt(variance)

print("Mean =", mean)
print("Variance =", variance)
print("Standard Deviation =", "%.2f"%stdDev)
```

### Output

```
Enter the number of elements in your list : 5
Enter the element : 45
Enter the element : 34
Enter the element : 86
Enter the element : 92
Enter the element : 35

The length of list1 is 5
List Contents [45, 34, 86, 92, 35]

Mean = 58.4
Variance = 642.64
Standard Deviation = 25.35
```

# Question 4

## Digit Frequency

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

### Python Code

```
num = input("Enter a number : ")
print("The number entered is :", num)

uniqDig = set(num)
#print(uniqDig)

for elem in uniqDig:
    print(elem, "occurs", num.count(elem), "times")
```

### Output

```
Enter a number : 234939
The number entered is : 234939
4 occurs 1 times
9 occurs 2 times
3 occurs 2 times
2 occurs 1 times

Enter a number : 7843338
The number entered is : 7843338
7 occurs 1 times
4 occurs 1 times
3 occurs 3 times
8 occurs 2 times
```

# Question 5

## Word Frequency in a File

**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]**

### Python Code

```
import sys
import string
import os.path


fname = input("Enter the filename : ")      #sample file text.txt also
provided


if not os.path.isfile(fname):
    print("File", fname, "doesn't exists")
    sys.exit(0)

infile = open(fname, "r")

filecontents = ""

for line in infile:
    for ch in line:
        if ch not in string.punctuation:
            filecontents = filecontents + ch
        else:
            filecontents = filecontents + ' '   #replace punctuations and
newline with a space

wordFreq = {}

wordList = filecontents.split()

#Calculate word Frequency

for word in wordList:
    if word not in wordFreq.keys():
        wordFreq[word] = 1
    else:
        wordFreq[word] += 1

sortedWordFreq = sorted(wordFreq.items(), key=lambda x:x[1], reverse=True )

print("\n==================================================")
print("10 most frequently appearing words with their count")
print("==================================================")
for i in range(10):
    print(sortedWordFreq[i][0], "occurs", sortedWordFreq[i][1], "times")
```

### Output

```
==================================================
10 most frequently appearing words with their count
==================================================
the occurs 45 times
of occurs 24 times
party occurs 12 times
part occurs 12 times
a occurs 9 times
and occurs 8 times
second occurs 7 times
to occurs 6 times
shall occurs 6 times
first occurs 5 times
```

# Question 6

## Sort File Contents

**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()].**

### Python Code

```
import os.path
import sys

fname = input("Enter the filename whose contents are to be sorted : ")
#sample file unsorted.txt also provided

if not os.path.isfile(fname):
    print("File", fname, "doesn't exists")
    sys.exit(0)

infile = open(fname, "r")

myList = infile.readlines()
# print(myList)

#Remove trailing \n characters
lineList = []
for line in myList:
    lineList.append(line.strip())

lineList.sort()

#Write sorted contents to new file sorted.txt

outfile = open("sorted.txt","w")


for line in lineList:
    outfile.write(line + "\n")

infile.close()  # Close the input file
```

```
outfile.close() # Close the output file


if os.path.isfile("sorted.txt"):
    print("\nFile containing sorted content sorted.txt created
successfully")
    print("sorted.txt contains", len(lineList), "lines")
    print("Contents of sorted.txt")

print("======================================================================")
    rdFile = open("sorted.txt","r")
    for line in rdFile:
        print(line, end="")
```

### Output

```
Enter the filename whose contents are to be sorted : unsorted.txt

File containing sorted content sorted.txt created successfully
sorted.txt contains 15 lines
Contents of sorted.txt
======================================================================
A deep C diva.
All the troubles you have will pass away very quickly.
Beware of a tall black man with one blond shoe.
Don't read everything you believe.
Exercise caution in your daily affairs.
He changes the domain.
How does a hacker fix a function which doesn't work for all of the elements
in its domain?
Lay on, MacDuff, and curs'd be him who first cries, "Hold, enough!".
People are beginning to notice you.  Try dressing before you leave the
house.
The surest protection against temptation is cowardice.
To be or not to be.
Tuesday is the Wednesday of the rest of your life.
What is the square root of 4b^2?
You display the wonderful traits of charm and courtesy.
You may be recognized soon.
```

# Question 7

## Backup Directory into Zip archive

**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.**

### Python Code

```
import os
import sys
import pathlib
import zipfile

dirName = input("Enter Directory name that you want to backup : ")

if not os.path.isdir(dirName):
```

```
    print("Directory", dirName, "doesn't exists")
    sys.exit(0)

curDirectory = pathlib.Path(dirName)

with zipfile.ZipFile("myZip.zip", mode="w") as archive:
    for file_path in curDirectory.rglob("*"):
        archive.write(file_path,
arcname=file_path.relative_to(curDirectory))

if os.path.isfile("myZip.zip"):
    print("Archive", "myZip.zip", "created successfully")
else:
    print("Error in creating zip archive")
```

### Output

```
Enter Directory name that you want to backup : zipDemo
Archive myZip.zip created successfully
```

# Question 8

## Assertions and Exceptions Demo

**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.**

### Python Code

```
import sys

def DivExp(a,b):
    assert a>0, "a should be greater than 0"
    try:
        c = a/b
    except ZeroDivisionError:
        print("Value of b cannot be zero")
        sys.exit(0)
    else:
        return c

val1 = int(input("Enter a value for a : "))
val2 = int(input("Enter a value for b : "))

val3 = DivExp(val1, val2)

print(val1, "/", val2, "=", val3)
```

### Output

```
Enter a value for a : 7
Enter a value for b : 6
7 / 6 = 1.1666666666666667
```

```
Enter a value for a : 0
Enter a value for b : 5
AssertionError: a should be greater than 0

Enter a value for a : -3
Enter a value for b : 6
AssertionError: a should be greater than 0

Enter a value for a : 6
Enter a value for b : 0
Value of b cannot be zero
```

# Question 9

## Complex Class Demo

**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.**

### Python Code

```
class Complex:
    def __init__(self, realp = 0, imagp=0):
        self.realp = realp
        self.imagp = imagp

    def setComplex(self, realp, imagp):
        self.realp = realp
        self.imagp = imagp

    def readComplex(self):
        self.realp = int(input("Enter the real part : "))
        self.imagp = int(input("Enter the real part : "))

    def showComplex(self):
        print('(',self.realp,')','+i','(',self.imagp,')',sep="")

    def addComplex(self, c2):
        c3 = Complex()
        c3.realp = self.realp + c2.realp
        c3.imagp = self.imagp + c2.imagp
        return c3

def add2Complex(a,b):
    c = a.addComplex(b)
    return c

def main():
    c1 = Complex(3,5)
    c2 = Complex(6,4)
```

```
    print("Complex Number 1")
    c1.showComplex()
    print("Complex Number 2")
    c2.showComplex()

    c3 = add2Complex(c1, c2)

    print("Sum of two Complex Numbers")
    c3.showComplex()

#Addition of N (N >=2) complex numbers

    compList = []

    num = int(input("\nEnter the value for N : "))

    for i in range(num):
        print("Object", i+1)
        obj = Complex()
        obj.readComplex()
        compList.append(obj)

    print("\nEntered Complex numbers are : ")
    for obj in compList:
        obj.showComplex()

    sumObj = Complex()
    for obj in compList:
        sumObj = add2Complex(sumObj, obj)


    print("\nSum of N complex numbers is", end = " ")
    sumObj.showComplex()

main()
```

## Output

```
Complex Number 1
(3)+i(5)
Complex Number 2
(6)+i(4)
Sum of two Complex Numbers
(9)+i(9)

Enter the value for N : 5
Object 1
Enter the real part : 1
Enter the real part : 9
Object 2
Enter the real part : 2
Enter the real part : 8
Object 3
Enter the real part : 3
Enter the real part : 7
Object 4
Enter the real part : 4
Enter the real part : 6
```

```
Object 5
Enter the real part : 5
Enter the real part : 5

Entered Complex numbers are :
(1) +i(9)
(2) +i(8)
(3) +i(7)
(4) +i(6)
(5) +i(5)

Sum of N complex numbers is (15)+i(35)
```

# Question 10

## Student Class Demo

**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.]**

### Python Code

```
class Student:
    def __init__(self, name = "", usn = "", score = [0,0,0,0]):
        self.name = name
        self.usn = usn
        self.score = score

    def getMarks(self):
        self.name = input("Enter student Name : ")
        self.usn = input("Enter student USN : ")
        self.score[0] = int(input("Enter marks in Subject 1 : "))
        self.score[1] = int(input("Enter marks in Subject 2 : "))
        self.score[2] = int(input("Enter marks in Subject 3 : "))
        self.score[3] = self.score[0] + self.score[1] + self.score[2]

    def display(self):
        percentage = self.score[3]/3
        spcstr = "=" * 81
        print(spcstr)
        print("SCORE CARD DETAILS".center(81))
        print(spcstr)
        print("%15s"%("NAME"), "%12s"%("USN"),
"%8s"%"MARKS1","%8s"%"MARKS2","%8s"%"MARKS3","%8s"%"TOTAL","%12s"%("PERCENT
AGE"))
        print(spcstr)
        print("%15s"%self.name, "%12s"%self.usn,
"%8d"%self.score[0],"%8d"%self.score[1],"%8d"%self.score[2],"%8d"%self.scor
e[3],"%12.2f"%percentage)
        print(spcstr)


def main():
    s1 = Student()
```

```
    s1.getMarks()
    s1.display()

main()
```

## Output

```
Enter student Name : Shivappa
Enter student USN : 1SI22CS065
Enter marks in Subject 1 : 87
Enter marks in Subject 2 : 79
Enter marks in Subject 3 : 92
================================================================================
======
                                  SCORE CARD DETAILS
================================================================================
======
           NAME           USN    MARKS1    MARKS2    MARKS3     TOTAL
PERCENTAGE
================================================================================
======
       Shivappa    1SI22CS065        87        79        92       258
86.00
================================================================================
======
```