VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELGAVI -590 014

Lab Manual (CSE STREAM)

“Introduction to Python Programming“

INTEGRATED

(Theory / Practical)

(Effective from the academic Year 2023-2024**)**

SEMESTER – II

Subject Code: BPLCK105B/205B

(Choice Based Credit System)

Prof. Poral NagarajB.E.,M.Tech. HOD. Dept. of CS&E, S.J.M.I.T, CHITRADURGA.	**Prof. Avinash G M** B.E.,M.Tech. Asst. Professor. Dept. of CS&E, S.J.M.I.T, CHITRADURGA.	**Prof. Monisha R** B.E.,M.Tech. Asst. Professor. Dept. of CS&E, S.J.M.I.T, CHITRADURGA.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SJM Vidyapeetha®

Sri Jagadguru Mallikarjuna Murugharajendra
Institute of Technology

CHITRADURGA -577 502

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.]

Note: CIE for the practical component of the Integrated Course

1. On completion of every experiment/program in the laboratory, the students shall be evaluated and marks shall be awarded on the same day. The15 marks are for conducting the experiment and preparation of the laboratory record, the other 10 marks shall be for the test conducted at the end of the semester.
2. The laboratory test (duration 02/03 hours) at the end of the 14th /15th week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to 10 marks.

Evaluation Type	**Maximum Marks**	**Minimum Passing Marks**
CIE Practical	15	06
CIE Practical Test	10	04
Total	**25**	10

Note: Scale down marks of experiments, record and test to 25

SJM Vidyapeetha®

SJM Institute of Technology, Chitradurga - 577502

Department of Computer Science & Engineering

Program Outcomes (POs)

PO 1: Apply the knowledge of Mathematics, Science and Computer Engineering to identify, formulate and solve any engineering problems with varied complexity.

PO 2: Design and develop a system, component or process to meet the desired needs within the realistic constraints to solve the real-time problems for betterment of society.

PO 3: Design and conduct experiments as well as analyze and interpret data.

PO 4: Communicate and Present the information effectively.

PO 5: Use the techniques, skills and modern engineering tools necessary for engineering practice.

PO 6: Handle various technical, administrative and managerial responsibilities successfully in any organizations globally.

PO 7: Get Recognize as successful Entrepreneur globally.

PO 8: Demonstrate commitment in handling any responsibilities with professional, ethical and social importance.

PO 9: Engage in lifelong learning to upgrade their engineering skills consistently.

PO 10: Adapt to any working environment of heterogeneous and multidisciplinary teams with good sustainability and high performance.

PO 11: Clear successfully the competitive exams for placement, higher studies and government services.

PO 12: Understand and demonstrate the impact of engineering solutions in a global , economic , environmental and societal context.

Program Specific Outcomes (PSOs)

PSO 1: An ability to design and develop hardware and software in emerging technology environments like cloud computing embedded products and real-time systems. (Orientation towards Systems Programming)

PSO 2: Knowledge of data management system like data acquisition, big data so as to enable students in solving problems using the techniques of data analytics like pattern recognition and knowledge discovery. (Orientation towards Data Sciences)

SJM Vidyapeetha®

SJM Institute of Technology, Chitradurga - 577502

Department of Computer Science & Engineering

Course Outcomes (Course Skill Set)

Subject:	**Introduction to Python Programming.**
Code:	**BPLCK105B / 205B**
At the end of the course the student will be able to:	
CO 1:	Demonstrate proficiency in handling loops and creation of functions.
CO 2:	Identify the methods to create and manipulate lists, tuples and dictionaries.
CO 3:	Develop programs for string processing and file organization.
CO 4:	Interpret the concepts of Object-Oriented Programming as used in Python.

Procedure to write and execute Python Program

1. Install Python 3.8: If you don't have Python 3.8 installed on your system, you can download and install it from the official Python website (https://www.python.org/downloads/). Follow the instructions specific to your operating system.
2. Open the terminal in IDLE 3.8 version.
3. Python terminal opens, go to the file and select the new file.
4. Type the program in the newly opened page.
5. Save the program: **Ctrl + S** and filename should be .py extension.
6. Run the program using F5 key. or click on RUN.
7. **To see the output:** The output will be displayed on the terminal prompt window.

PRACTICE PROGRAMS

1. write a python code to addition of two numbers

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

                                        # Add the numbers
sum = num1 + num2

                                        # Display the result
print("The sum of", num1, "and", num2, "is", sum)
```

OUTPUT

```
Enter the first number: 25
Enter the second number: 26
The sum of 25.0 and 26.0 is 51.0
-----------------------------------------OR------------------------------------------------------
```

```
num1=int(input("Enter first number: "))

num2=int(input("Enter second number: "))
                                        # Add the numbers
sum=float(num1)+float(num2)
                                        # Display the result
print('sum of two numbers like {0} and {1} is {2}'.format(num1,num2,sum))
```

```
-------------------------------------------------------------------------------------------------
```

OUTPUT

```
Enter first number: 25
Enter second number: 26
sum of two numbers like 25 and 26 is 51.0
```

2. Write a python code to Calculate the Simple Interest

```
principal = float(input("Enter the principal amount: "))
rate = float(input("Enter the interest rate (%): "))
time = float(input("Enter the time period (in years): "))

                              # Calculate simple interest
    simple_interest = (principal * rate * time) / 100
```

```
                # Calculate total amount
total_amount = principal + simple_interest

                # Display the results
print("Principal amount:", principal)
print("Interest rate:", rate, "%")
print("Time period:", time, "years")
print("Simple interest:", simple_interest)
print("Total amount:", total_amount)
```

OUTPUT

```
Enter the principal amount: 1000
Enter the interest rate (%): 2
Enter the time period (in years): 1
Principal amount: 1000.0
Interest rate: 2.0 %
Time period: 1.0 years
Simple interest: 20.0
Total amount: 1020.0
```

3. Write a python code to check whether the given number is even or odd.

```
number = int(input("Enter a number: "))
                # Check if the number is divisible by 2
if number % 2 == 0:
    print(number, "is an even number.")
else:
    print(number, "is an odd number.")
```

OUTPUT

```
Enter a number: 25
25 is an odd number.
Enter a number: 26
26 is an even number.
```

4. write a python code to given integer is palindrome or not

```
number = int(input("Enter an integer: "))
                    # Convert the number to a string
    number_str = str(number)

            # Reverse the string
    reversed_str = number_str[::-1]

                    # Check if the original and reversed strings are equal
    if number_str == reversed_str:
     print(number, "is a palindrome.")
    else:
      print(number, "is not a palindrome.")
```

OUTPUT

```
Enter an integer: 1221
1221 is a palindrome.
Enter an integer: 12345
12345 is not a palindrome.
```

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.**

```
name = input("Enter the name of the student:")

usn= input("Enter the USN of the student:")

print("Enter the marks scored by students in 3 subjects:")

marks_physics=int(input(" Enter the marks scored in physics:"))

marks_chemistry=int(input(" Enter the marks scored in chemistry:"))

marks_maths=int(input(" Enter the marks scored in maths:"))

total_marks=marks_physics + marks_chemistry + marks_maths

percentage =total_marks/3

print("The  student details are:")

print("Name:",name)

print("USN:",usn)

print("marks scored:")

print("physics:",marks_physics)

print("chemistry:",marks_chemistry)

print("maths:",marks_maths)

print("Total:",total_marks)

print("percentage:",percentage)
```

OUTPUT

```
Enter the name of the student: Manju
Enter the USN of the student: 4sm18CS011
Enter the marks scored by students in 3 subjects:
```

```
Enter the marks scored in physics :70
Enter the marks scored in chemistry:65
Enter the marks scored in maths:72

The student details are:
Name: manju
USN: 4sm18CS011
marks scored:
physics: 70
chemistry: 65
maths: 72
Total: 207
percentage:69.00
```

----

1 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.

```
import datetime
name=input("Enter the name of the person:")
birth_year=int(input("Enter the year of birth of person:"))
today = datetime.date.today( )
year = today.year
age = year - birth_year
if age>=60:
  print(name," is a senior citizon.")
else:
  print(name," is not senior citizon.")
```

OUTPUT

```
Enter the name of the person: Manukumar
Enter the year of birth of person:1965
Manukumar  is not senior citizon.
Enter the name of the person: Ravikumar
enter the year of birth of person:1950
Ravikumar  is a senior citizon.
```

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

```
nterms = int(input("Enter the required length of Fibonacci Sequence: "))
                                   # first two terms
n1, n2 = 0, 1
count = 0
                                   # check if the number of terms is valid
if nterms <= 0:
  print("Please enter a positive integer")
                                   # if there is only one term, return n1
elif nterms == 1:
  print("Fibonacci sequence upto",nterms,":")
  print(n1)
                                   # generate fibonacci sequence
else:
  print("Fibonacci sequence:")
  while count < nterms:
     print(n1)
     nth = n1 + n2
                                    # update values

     n1 = n2
     n2 = nth
     count += 1
```

OUTPUT

```
Enter the required length of Fibonacci Sequence: 5
Fibonacci sequence:
0
1
1
2
3
```

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

```
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: 5
Enter the value of R (R cannot be negative or greater than N): 3
5C3 = 10

Enter the value of N: 10
Enter the value of R (R cannot be negative or greater than N): 5
10C5 = 252

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

---

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

```
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: 3
Enter the element : 10
Enter the element : 20
Enter the element : 30
```

```
The length of list1 is 3

List Contents [10, 20, 30]

Mean = 20.0

Variance = 66.66666666666667

Standard Deviation = 8.16
```

----

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

```
num=input("Input multi-digit number:")
n=len(num)
n0,n1,n2,n3,n4,n5,n6,n7,n8,n9=0,0,0,0,0,0,0,0,0,0
for i in range(n):
    if num[i]=='0':
        n0+=1
    elif num[i]=='1':
            n1+=1
    elif num[i]=='2':
            n2+=1
    elif num[i]=='3':
            n3+=1
    elif num[i]=='4':
            n4+=1
    elif num[i]=='5':
            n5+=1
    elif num[i]=='6':
            n6+=1
    elif num[i]=='7':
            n7+=1
    elif num[i]=='8':
            n8+=1
    elif num[i]=='9':
            n9+=1
           dfreq=[n0,n1,n2,n3,n4,n5,n6,n7,n8,n9]
print("The number", num,"has:")
for i in range(10):
    if dfreq[i]==0:
        continue
print(i,"digit", dfreq[i],"times.")
```

OUTPUT

```
Input multi-digit number:112344566789
The number 112344566789 has:
1 digit 2 times.
2 digit 1 times.
3 digit 1 times.
4 digit 2 times.
5 digit 1 times.
6 digit 2 times.
7 digit 1 times.
8 digit 1 times.
9 digit 1 times.
```

---

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]

(**Example:** Filename.py→ 5.py)

```
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 \n with 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
                                        #Sort Dictionary based on values in descending order
sortedWordFreq = sorted(wordFreq.items(), key=lambda x:x[1], reverse=True )
```

```
                        #Display 10 most frequently appearing words with their count
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")
```

(Example:Filename.txt:text.txt)

How many lawyers does it take to change a light bulb?

Whereas the party of the first part, also known as "Lawyer", and the party of the second part, also known as "Light Bulb", do hereby and forthwith agree to a transaction wherein the party of the second part shall be removed from the current position as a result of failure to perform previously agreed upon duties, i.e., the lighting, elucidation, and otherwise illumination of the area ranging from the front (north) door, through the entryway, terminating at an area just inside the primary living area, demarcated by the beginning of the carpet, any spillover illumination being at the option of the party of the second part and not required by the aforementioned agreement between the parties.

The aforementioned removal transaction shall include, but not be limited to, the following. The party of the first part shall, with or without elevation at his option, by means of a chair, stepstool, ladder or any other means of elevation, grasp the party of the second part and rotate the party of the second part in a counter-clockwise direction, this point being tendered non-negotiable. Upon reaching a point where the party of the second part becomes fully detached from the receptacle, the party of the first part shall have the option of disposing of the party of the second part in a manner consistent with all relevant and applicable local, state and federal statutes.

Once separation and disposal have been achieved, the party of the first part shall have the option of beginning installation. Aforesaid installation shall occur in a manner consistent with the reverse of the procedures described in step one of this self-same document, being careful to note that the rotation should occur in a clockwise direction, this point also being non-negotiable.

The above described steps may be performed, at the option of the party of the first part, by any or all agents authorized by him, the objective being to produce the most possible revenue for the Partnership.

Don't read everything you believe.

OUTPUT

```
Enter the filename : text.txt
=======================================================
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
```

----

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()].*

(Example: Filename.py→6.py)

```
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="")
```

(**Example:**filename.txt→unsorted.txt)

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

OUTPUT

```
Enter the filename whose contents are to be sorted : unsorted.txt
File containing sorted content sorted.txt created successfully
sorted.txt contains 16 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.
The surest protection against temptation is cowardice.
To be or not to be.
Try dressing before you leave the house.
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.
```

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.

(**Example:** Filename.py→7.py)

```
import os

import zipfile

def backup_folder(folder_path):

    # Get the folder name from the path

    folder_name = os.path.basename(folder_path)

    # Create a ZIP file with the folder name

    zip_filename = f"{folder_name}.zip"

    zip_path = os.path.join(os.getcwd(), zip_filename)

    with zipfile.ZipFile(zip_path, 'w') as zip_file:

        # Iterate through all the files and subdirectories in the folder

        for root, _, files in os.walk(folder_path):

            for file in files:

                # Get the full path of the file

                file_path = os.path.join(root, file)

                # Add the file to the ZIP file with the corresponding relative path

                zip_file.write(file_path, os.path.relpath(file_path, folder_path))

    print(f"Backup created: {zip_filename}")

# Example usage: backup the folder "my_folder" in the current working directory

folder_to_backup = os.path.join(os.getcwd(), "New_Folder")

backup_folder(folder_to_backup)
```

OUTPUT

Backup created: New_Folder.zip

---

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.

(**Example:** Filename.py→8.py)

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

```
Enter a value for a : 0
Enter a value for b : 5
Traceback (most recent call last):
  File "C:/Users/AGM/AppData/Local/Programs/Python/Python38-32/8.py", line 16,
in <module>
    val3 = DivExp(val1, val2)
  File "C:/Users/AGM/AppData/Local/Programs/Python/Python38-32/8.py", line 4, in
DivExp
  assert a>0, "a should be greater than 0"
  AssertionError: a should be greater than 0
------------------------------------------------------------------------------------------------
Enter a value for a : -3
Enter a value for b : 6
Traceback (most recent call last):
  File "C:/Users/AGM/AppData/Local/Programs/Python/Python38-32/8.py", line 16, in
<module>
    val3 = DivExp(val1, val2)
  File "C:/Users/AGM/AppData/Local/Programs/Python/Python38-32/8.py", line 4, in
DivExp
    assert a>0, "a should be greater than 0"
AssertionError: a should be greater than 0
------------------------------------------------------------------------------------------------
```

9. **Define a function which takes TWO objects representing complex numbers and returns new complex number with an 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, 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 : 2
Object 1
Enter the real part : 3
Enter the real part : 2
Object 2
Enter the real part : 6
Enter the real part : 5

Entered Complex numbers are :
(3)+i(2)
(6)+i(5)

Sum of N complex numbers is (9)+i(7)
```

```
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)
```

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, 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"%("PERCENTAGE"))
print(spcstr)
print("%15s"%self.name,"%12s"%self.usn,"%8d"%self.score[0],"%8d"%self.score[1]
,"%8d"%self.score[2],"%8d"%self.score[3],"%12.2f"%percentage)
 print(spcstr)
def main():
    s1 = Student()
    s1.getMarks()
    s1.display()

main()
```

OUTPUT

```
Enter student Name : Manukumar
Enter student USN : 4SM23cs021
Enter marks in Subject 1 : 65
Enter marks in Subject 2 : 75
Enter marks in Subject 3 : 62
=====================================================================
                         SCORE CARD DETAILS
=====================================================================
   NAME      USN    MARKS1   MARKS2   MARKS3   TOTAL   PERCENTAGE
=====================================================================
  Manukumar   4SM23CS021     65       75        62        202         67.33
=====================================================================
```