



KSSEM
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

K.S. GROUP OF INSTITUTIONS

K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

#15, Mallasandra, Near Vajarahalli, Off. Kanakapura Road, Bengaluru - 560 109

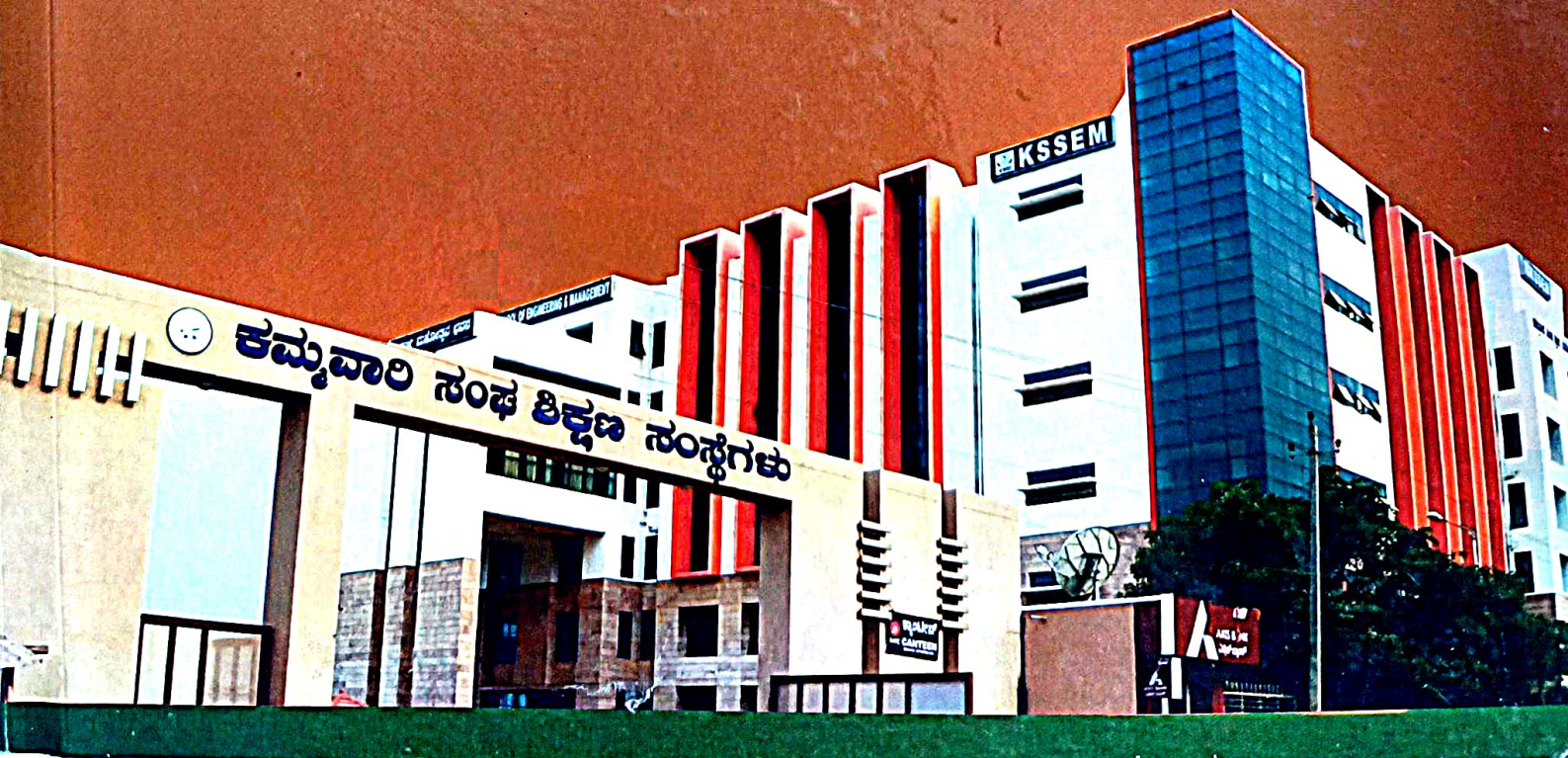
PRACTICAL RECORD BOOK

NAME : M.B. CHETHANA

Class : CSE "B"

Lab : Python

USN : 1 K G 2 3 C S 0 6 7



K.S. GROUP OF INSTITUTIONS
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

#15, Mallasandra, Near Vajarahalli, Off. Kanakapura Road, Bengaluru - 560 109



KSSEM
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

Laboratory Certificate

This is to certify that

Mr. / Ms. M.B. Chethana
has satisfactorily completed the course of experiment in
.....Python.....programming..... Laboratory, Code BPLCK205B
Prescribed by Visvesvaraya Technological University, Belagavi
for the2nd..... Semester B.E.CSE..... branch in this
college during in the academic year 20 ...23.. - 20 ..24..

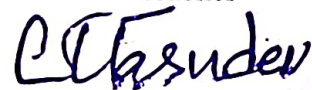
Name of the Candidate : M.B. Chethana

USN : 1KG23CS067 Lab (with code) BPLCK205B

Marks	
Maximum	Obtained
25	24

Date : 01-07-2024


Signature of the
Teacher



Head of the Department

Dr. C. VASUDEV

Professor & HOD

Department of Applied Science
K.S. School of Engineering & Management
Bangalore - 560 109

Particulars of the Experiments Performed

CONTENTS

[illegible]

Introduction to Python programming :-

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990.

- Python is interpreted - python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is interactive - you can actually sit at a python prompt and interact with the interpreter directly to write your programs.
- Python is object-oriented - python supports object-oriented style or technique of programming that encapsulates codes within objects.
- Python is a Beginner's language - python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to www browsers to games.

Python Features :-

- Easy-to-learn - python has few keywords, simple structure and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read - python code is more clearly defined and visible to the eyes.
- Easy-to-maintain - Python's bulk of the library is very portable and cross-platform compatible on UNIX, windows, and Macintosh.
- Interactive Mode - Python has support for an interactive mode which allows interactive testing and debugging.

of snippets of code.

- Portable - Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable - You can add low-level modules to the python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases - python provides interface to all major commercial databases.
- GUI programming - python supports GUI applications that can be created and ported to many system calls, libraries and windows.
Systems, such as windows MEC, Macintosh, and the X window system of Unix.
- Scalable - python provides a better structure and support for large programs than shell scripting.

First Python Program :-

Open jupyter notebook and type following program:-

1. `print("Hello world")`

Output :- Hello world

2. `a = 10`

`b = 20`

`print(a+b)`

Output :- 30

3. `a = int(input("enter the value of a "))`

`b = int(input("enter the value of b "))`

`c = a + b`

DATE 14-03-2024

EXPT TITLE:

PAGE NO.

3

EXP. NO.

```
print(c)
```

Output:- enter the value of a 10

enter the value of b 25

35

4. a = input("enter the value for a ")

b = input("enter the value for b ")

c = a + b

```
print(c)
```

Output:- enter the value for a 10

enter the value for b 25

1025

Punit
21/3/24

10
10

Program :- 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 student : ")
```

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

```
marks1 = int(input("Enter marks in subject 1 : "))
```

```
marks2 = int(input("Enter marks in subject 2 : "))
```

```
marks3 = int(input("Enter marks in subject 3 : "))
```

```
total = marks1 + marks2 + marks3
```

```
percentage = (marks1 + marks2 + marks3) / 300 * 100
```

```
print("Student Details \n")
```

```
print("Name : ", name)
```

```
print("USN : ", USN)
```

```
print("Marks 1 : ", marks1)
```

```
print("Marks 2 : ", marks2)
```

```
print("Marks 3 : ", marks3)
```

```
print("Total : ", total)
```

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

Output :-

Enter the name of the student : chethanabrahma

Enter the USN of the student : 1KG23C6067

Enter the marks in subject 1 : 85

Enter the marks in subject 2 : 93

Enter the marks in subject 3 : 80

DATE 14-03-2024

EXPT TITLE: Program to display person is
a senior citizen or not

EXP. NO. 1(b)

PAGE NO. 05

Student details

Name : chethanabrahma

USN : 1KG23CS067

Marks 1 : 85

Marks 2 : 93

Marks 3 : 80

Percentage : 86.0

Program:- 1(b)Punit
2/3/24

A

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
```

```
Name = input("Enter the name of the person : ")
```

```
DOB = int(input("Enter his name year of birth : "))
```

```
curYear = date.today().Year
```

```
Age = curYear - DOB
```

```
if (Age > 60):
```

```
    print(Name, "aged", Age, "Years is a senior citizen")
```

```
else:
```

```
    print(Name, "aged", Age, "Years is not a senior citizen")
```

Output:-

(1) Enter the name of the person : ramesh

Enter his year of birth : 1975

ramesh aged 49 years is not a senior citizen.

DATE ...14.03.2024.

EXPT TITLE:

PAGE NO. 06

EXP. NO.

(2) Enter the name of the person : ravi
Enter his year of birth : 1950
ravi aged 74 years is a senior citizen

~~penit~~
21/3/24

10
10

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



DATE 21-03-2024

EXPT TITLE: Fibonacci Sequence

PAGE NO. 07

EXP. NO. 2(a)

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

```
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
```

Punit
28/3/24

Output:-

Enter the value of N: 5

120

Enter the value of N: 5

Enter the value of R (R cannot be negative or greater than N):

50 = 5

28/12/21
12/12/21

Write a function to calculate factorial of a number. Develop a program to compute binomial co-efficient (Given N and R).

```
def fact(num):
```

```
    if num == 0:
```

```
        return 1
```

```
    else:
```

```
        return num * fact(num-1)
```

```
x = int(input("Enter the value of num: "))
```

```
print(fact(x))
```

```
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, "=", " %.2f" % nCr, sep=" ")
```

Submitted
28/3/24

10
10

Output :-

Enter the number of elements in your list : 5

Enter the element : 45

Enter the element : 36

Enter the element : 34

Enter the element : 26

Enter the element : 38

The length of list 1 is 5

List contents [45, 36, 34, 26, 38]

Mean = 35.8

Variance = 37.76

Standard Deviation = 6.14



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 list 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)
```

✓
Punitha
4/4/24

10
10

Output:-

② Enter a number : 100
The number entered is : 100
{ '1', '0' }
1 occurs 1
0 occurs 2

① Enter a number : 100
The number entered is : 100
1 occurs 1
0 occurs 2

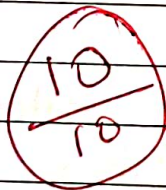
14/12/23
Page No. 10

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

```
① num = input("Enter a number: ")
print("The number entered is: ", num)
uniqdig = set(num)
for elem in uniqdig:
    print(elem, "occurs", num.count(elem))
```

```
② 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))
```

Punit
4/4/24



DATE 04-04-2024

EXPT TITLE:

Program to print 10 most frequently appearing words in textfile

EXP. NO. 05

PAGE NO. 11

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

```
import sys
import string
import os.path
fname = input("Enter the filename: ")
if not os.path.isfile(fname):
    print("file", fname, "doesn't exist")
    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 + ' '
wordfreq = {}
wordlist = filecontents.split()
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)
```

Output :-

Enter the filename : chethana.txt

10 most frequently appearing words with their count

dictionary occurs 3 times

of occurs 3 times

a occurs 2 times

to occurs 2 times

words occurs 2 times

in occurs 2 times

and occurs 2 times

frequency occurs 2 times

the occurs 2 times

Develop occurs 1 times

DATE 04-04-2024

EXPT TITLE:

EXP. NO. 05

PAGE NO.

12

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

~~Run it~~
12/4/24

10
10

Output :-

Enter the filename whose contents are to be sorted : chetandrab
file containing sorted content sorted.txt created successfully
sorted.txt contains 2 lines
contents of sorted.txt

Develop a program to sort the contents of a text file and
write the sorted contents into a separate text file.

Hint : Use string method `strip()`, `line()`, list methods `sort()`,
`append()`, and file methods `open()`, `readlines()`, and `writeln()`

Develop a program to sort the contents of a text file and write the sorted contents into a separate textfile. [Hint: Use string method strip(), len(), list methods sort(), append(), and the methods open(), readlines(), and write().]

```
import os.path
import sys
fname = input("Enter the filename whose contents are to be
sorted: ")
if not os.path.isfile(fname):
    print("File", fname, "doesn't exist")
    sys.exit(0)
infile = open(fname, "r")
myList = infile.readlines()
lineList = []
for line in myList:
    lineList.append(line.strip())
lineList.sort()
outfile = open("sorted.txt", "w")
for line in lineList:
    outfile.write(line + "\n")
infile.close()
outfile.close()
if os.path.isfile("sorted.txt"):
    print("\n File containing sorted content sorted.txt
    created successfully")
    print("sorted.txt contains", len(lineList), "lines")
    print("contents of sorted.txt")
```

DATE 04-04-2024

EXPT TITLE:

EXP. NO.06.....

PAGE NO. (4)

```
rdfile = open("sorted.txt", "r")
```

```
for line in rdfile:
```

```
    print(line, end = " ")
```

Punit
28/4/24

10
10

Output :-

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

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 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")
```

~~Run into~~
18/4/24

10
10

Output :-

① Enter a value for a : 7
Enter a value for b : 5
 $7/5 = 1.4$

② Enter a value for a : 5
Enter a value for b : 0
value of b cannot be 0

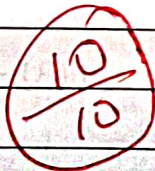


Program -08 :-

Write a function named DivExp which takes two parameters a, b and returns 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):
    assert a>0, "a should be greater than 0"
    try:
        c = a/b
    except zero-DivisionError:
        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)
```

Run it
2/5/24



Output :-

Complex Number 1

$$(3) + i(5)$$

Complex Number 2

$$(6) + i(4)$$

Sum of two Complex Number

$$(9) + i(9)$$

Enter the value for N : 3

Object 1

Enter the real part : 2

Enter the imaginary part : 6

Object 2

Enter the real part : 6

Enter the imaginary part : 4

Object 3

Enter the real part : 5

Enter the imaginary part : 4

Entered Complex Numbers are:

$$(2) + i(6)$$

$$(6) + i(4)$$

$$(5) + i(4)$$

Sum of N complex numbers is $(13) + i(14)$

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 ($N \geq 2$) 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 imaginary 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 Number")
c3.showComplex()
complist = []
num = int(input("\n Enter the value for N: "))
for i in range(num):
    print("Object ", i+1)
    obj = Complex()
    obj.readComplex()
    complist.append(obj)
print("\n Entered complex numbers are: ")
for obj in complist:
    obj.showComplex()
sumObj = Complex()
for obj in complist:
    sumObj = add2Complex(sumObj, obj)
print("\n Sum of N Complex numbers is, end=")
sumObj.showComplex()
```

main()

~~Run this~~
4/5/24

10
10

Output :-

Enter the student name : chethana

Enter student usn : 1KG23CS067

Enter marks in subject 1 : 90

Enter marks in subject 2 : 95

Enter marks in subject 3 : 87

SCORE CARD DETAILS

NAME	USN	MARKS 1	MARKS 2	MARKS 3	TOTAL	PERCENTAGE
chethana	1KG23CS067	90	95	87	272	90.67

90
95
87

272
90.67

Develop a program that uses class student which prompts the user to enter marks in three subjects and calculates total marks and percentage and display 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()
```

Punit
11/11/24

10
10