SAMBHRAM INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belgaum)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BANGALORE

LABORATORY MANUAL

PYTHON PROGRAMMING

(21CSL46)

(Effective from the academic year 2023 -2024)

for

BE: CS- IV Semester

SAMBHRAM INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering
BANGALORE

LAB MANUAL

PYTHON PROGRAMMING (21CSL46)

For

IV - SEMESTER LAB

PREPARED BY:

FACULTY NAME:

(1) …………………………………………………………………….

(2) …………………………………………………………………….

BRANCH: ………………………………….

Department of Computer Science and Engineering

Vision of the Department:
To professionally drive and train the young minds by placing them in forefront of technological, scientific and engineering discipline to glorify the future of Sambhram institute of technology as a shining star in the world of computer technology

Mission of the Department:
M1. To act as the nurturing ground for young professionals who seek to make a difference and create a talent pool for various industries so that there may be synergistic growth for both.
M2. To establish state-of-the-art facilities meeting the requirements of specified curriculum to inculcate learning process of the students.
M3. To provide exposure of latest tools and technologies in the area of engineering and technologies
M4. To produce successful graduates with personal and professional responsibilities and commitment to lifelong learning.

Program Educational Objectives
PEO-1: To produce graduates having a strong background of basic science, Mathematics & Engineering and ability to use these tools.
PEO-2: To produce graduates who can demonstrate technical competence in the field of computer science and engineering and develop solutions to the complex problems.
PEO-3: To produce graduates having professional competence through life-long learning such as advanced degrees, professional skills and other professional activities related globally to engineering & society.
PEO- 4: To produce graduates who function effectively in a multi-disciplinary environment and individually, within a societal and environmental context.
PEO-5: To produce graduates who would be able to take individual responsibility and work as a part of a team towards the fulfillment of both individual and organizational goals

Program Specific Outcomes
PSO 1: should be able to understand the concepts computer science and engineering and their applications in the field of emerging technologies and other relevant areas.
PSO 2: Should have an ability to apply technical knowledge and usage of modern hardware & software tools related to computer science and engineering for solving real world problems.
PSO 3: Should have the capability to analyze, comprehend, design & develop subsystems/ systems for a variety of engineering applications and thus demonstrating professional ethics & concern for societal wellbeing.

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design / development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities withan understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member orleader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DO’S AND DON’TS

Do’s

1. Do wear ID card and follow dress code.
2. Do log off the computers when you finish.
3. Do ask the staff for assistance if you need help.
4. Do keep your voice low when speaking to others in the LAB.
5. Do ask for assistance in downloading any software.
6. Do make suggestions as to how we can improve the LAB.
7. In case of any hardware related problem, ask LAB in charge for solution.
8. If you are the last one leaving the LAB, make sure that the staff in charge of theLAB is informed to close the LAB.
9. Be on time to LAB sessions.
10. Do keep the LAB as clean as possible.

Don’ts

1. Do not use mobile phone inside the lab.
2. Don’t do anything that can make the LAB dirty (like eating, throwing waste papers etc.).
3. Do not carry any external devices without permission.
4. Don’t move the chairs of the LAB.
5. Don’t interchange any part of one computer with another.
6. Don’t leave the computers of the LAB turned on while leaving the LAB.
7. Do not install or download any software or modify or delete any system files onany lab computers.
8. Do not damage, remove, or disconnect any labels, parts, cables, or equipment.
9. Don’t attempt to bypass the computer security system.
10. Do not read or modify other user’s file.
11. If you leave the lab, do not leave your personal belongings unattended. We are notresponsible

PYTHON PROGRAMMING LABORATORY

Course Code	**21CSL46**	CIE Marks	50
Teaching Hours/Weeks (L: T: P: S)	0: 0: 2: 0	SEE Marks	50
Total Hours of Pedagogy	24	Total Marks	100
Credits	01	Exam Hours	03

Course Objectives:

CLO 1. Demonstrate the use of IDLE or PyCharm IDE to create Python Applications

CLO 2. Using Python programming language to develop programs for solving real-world problemsCLO 3. Implement the Object-Oriented Programming concepts in Python.

CLO 4. Appraise the need for working with various documents like Excel, PDF, Word and Others

CLO 5. Demonstrate regular expression using python programming

Note: two hours tutorial is suggested for each laboratory sessions.

Prerequisite

- Students should be familiarized about Python installation and setting Python environment
- Usage of IDLE or IDE like PyCharm should be introduced

Python Installation: https://www.youtube.com/watch?v=Kn1HF3oD19c PyCharm Installation: https://www.youtube.com/watch?v=SZUNUB6nz3g

Sl. No.	***PART A – List of problems for which student should develop program and execute inthe Laboratory***
1	**Aim:** Introduce the Python fundamentals, data types, operators, flow control and exceptionhandling in Python a) Write a python program to find the best of two test average marks out of three test'smarks accepted from the user. b) Develop a Python program to check whether a given number is palindrome or not andalso count the number of occurrences of each digit in the input number. Datatypes: https://www.youtube.com/watch?v=gCCVsvgR2KU Operators: https://www.youtube.com/watch?v=v5MR5JnKcZI Flow Control: https://www.youtube.com/watch?v=PqFKRqpHrjwFor loop: https://www.youtube.com/watch?v=0ZvaDa8eT5s While loop: https://www.youtube.com/watch?v=HZARImviDxg Exceptions: https://www.youtube.com/watch?v=6SPDvPK38tw
2	**Aim:** Demonstrating creation of functions, passing parameters and return values a) Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed. b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions. Functions: https://www.youtube.com/watch?v=BVfCWuca9nw Arguments: https://www.youtube.com/watch?v=ijXMGpoMkhQ Return value: https://www.youtube.com/watch?v=nuNXiEDnM44
3	**Aim:** Demonstration of manipulation of strings using string methods a) Write a Python program that accepts a sentence and find the number of words, digits,uppercase letters and lowercase letters.

	b) Write a Python program to find the string similarity between two given strings **Sample Output:** Original string: Python Exercises Python Exercises Similarity between two said strings: **Sample Output:** Original string: Python Exercises Python Exercise Similarity between two said strings:1.0 0.967741935483871 Strings: https://www.youtube.com/watch?v=lSItwlnF0eU String functions: https://www.youtube.com/watch?v=9a3CxJyTq00
4	**Aim:** Discuss different collections like list, tuple and dictionary a) Write a python program to implement insertion sort and merge sort using lists b) Write a program to convert roman numbers in to integer values using dictionaries. Lists: https://www.youtube.com/watch?v=Eaz5e6M8tL4 List methods: https://www.youtube.com/watch?v=8-RDVWGktuITuples: https://www.youtube.com/watch?v=bdS4dHIJGBc Tuple operations: https://www.youtube.com/watch?v=TItKabcTTQ4 Dictionary: https://www.youtube.com/watch?v=4Q0pW8XBOkc Dictionary methods: https://www.youtube.com/watch?v=oLeNHuORpNY
5	**Aim:** Demonstration of pattern recognition with and without using regular expressions a) Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression. b) Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com) Regular expressions: https://www.youtube.com/watch?v=LnzFnZfHLS4
6	**Aim:** Demonstration of reading, writing and organizing files. a) Write a python program to accept a file name from the user and perform thefollowing operations 1. Display the first N line of the file 2. Find the frequency of occurrence of the word accepted from the user in thefile b) Write a python program to create a ZIP file of a particular folder which containsseveral files inside it. Files: https://www.youtube.com/watch?v=vuyb7CxZgbU https://www.youtube.com/watch?v=FqcjKewJTQ0 File organization: https://www.youtube.com/watch?v=MRuq3SRXses
7	**Aim:** Demonstration of the concepts of classes, methods, objects and inheritance

	a) By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle. b) Write a python program by creating a class called Employee to store the details of Name, Employee_ID, Department and Salary, and implement a method to update salary of employees belonging to a given department. OOP’s concepts: https://www.youtube.com/watch?v=qiSCMNBIP2g Inheritance: https://www.youtube.com/watch?v=Cn7AkDb4pIU
8	**Aim:** Demonstration of classes and methods with polymorphism and overriding a) Write a python program to find the whether the given input is palindrome or not (forboth string and integer) using the concept of polymorphism and inheritance. Overriding: https://www.youtube.com/watch?v=CcTzTuIsoFk
9	**Aim:** Demonstration of working with excel spreadsheets and web scraping a) Write a python program to download the all XKCD comics b) Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet Web scraping: https://www.youtube.com/watch?v=ng2o98k983k Excel: https://www.youtube.com/watch?v=nsKNPHJ9iPc
10.	**Aim:** Demonstration of working with PDF, word and JSON files a) Write a python program to combine select pages from many PDFs b) Write a python program to fetch current weather data from the JSON file PDFs: https://www.youtube.com/watch?v=q70xzDG6nls https://www.youtube.com/watch?v=JhQVD7Y1bsA https://www.youtube.com/watch?v=FcrW-ESdY-A Word files: https://www.youtube.com/watch?v=ZU3cSl51jWE JSON files: https://www.youtube.com/watch?v=9N6a-VLBa2I

PART B – Practical Based Learning

A problem statement for each batch is to be generated in consultation with the co-examiner and student should develop an algorithm, program and execute the program for the given problem with appropriate outputs.

Course Outcomes:

CO 1. Demonstrate proficiency in handling of loops and creation of functions.

CO 2. Identify the methods to create and manipulate lists, tuples and dictionaries.

CO 3. Discover the commonly used operations involving regular expressions and file system. CO 4. Interpret the concepts of Object-Oriented Programming as used in Python. CO 5. Determine the need for scraping websites and working with PDF, JSON and other file formats

Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each course. The student has to secure not less than 35% (18 Marks out of 50) in the semester-end examination (SEE). The student has to secure 40% of sum of the maximum marks of CIE and SEE to qualify in the course.

Continuous Internal Evaluation (CIE):

- CIE marks for the practical course is **50 Marks**.
- The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.
- Each experiment to be evaluated for conduction with observation sheet and record write-up.
- Rubrics for the evaluation of the journal/write-up for hardware/software experiments designed by the faculty who is handling the laboratory session and is made known to students at the beginning of the practical session.
- Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10 marks.
- Total marks scored by the students are scaled downed to 30 marks (60% of maximum marks).
- Weightage to be given for neatness and submission of record/write-up on time.
- Department shall conduct 02 tests for 100 marks, the first test shall be conducted after the 8th week of the semester and the second test shall be conducted after the 14th week of the semester.
- In each test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% for viva-voce. suitable rubrics can be designed to evaluate each student's performance and learning ability. The Rubrics suggested in Annexure-II of Regulation book
- The average of 02 tests is scaled down to **20 marks** (40% of the maximum marks).
- The Sum of scaled-down marks scored in the report write-up/journal and average marks of two tests is the total CIE marks scored by the student.

Semester End Evaluation (SEE):

- SEE marks for the practical course is 50 Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the University
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. **OR** based on the course requirement evaluation rubrics shall be decided jointly by examiners.

- Students can pick one question (experiment) from the questions lot prepared by the internal /external examiners jointly.
- Students can pick one experiment from the questions lot of PART A with equal choice to all the students in a batch.
- For PART B examiners should frame a question for each batch, student should develop an algorithm, program, execute and demonstrate the results with appropriate output forthe given problem.
- Weightage of marks for PART A is 80% and for PART B is 20%. General rubrics suggested to befollowed for part A and part B.
- Change of experiment is allowed only once and Marks allotted to the procedure part to bemade zero (Not allowed for Part B).
- The duration of SEE is 03 hour
- Rubrics suggested in Annexure-II of Regulation book

Textbooks:

1. Al Sweigart, “Automate the Boring Stuff with Python”,1stEdition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
2. Reema Thareja “Python Programming Using Problem Solving Approach” Oxford University Press.
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”,2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf)

PART A – List of problems for which student should develop program and execute in the Laboratory

1. Aim: Introduce the Python fundamentals, data types, operators, flow control and exception handling in Python

1 a) Write a python program to find the best of two test average marks out of three test's marks accepted from the user.

```
m1 = int (input("Enter the marks in the first test: "))
m2 = int (input("Enter the marks in second test: "))
m3 = int (input("Enter the marks in third test: "))

if (m1 > m2):
    if (m2 > m3):
        total = m1 + m2
    else:
        total = m1 + m3
elif (m1 > m3):
    total = m1 + m2
else:
    total = m2 + m3

Avg = total / 2
print ("The average of the best two test marks is: “, Avg)
```

Output

```
1. Enter marks for test1: 45
   Enter marks for test2: 39
   Enter marks for test3: 48
   Average of best two test marks out of three test’s marks is 46.5
```

1 b) Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.

```
num = input("Enter a number: ")
if num == num[::-1]:
    print(num, "is a palindrome")
else:
    print(num, "is not a palindrome")
digit_count = [0]*10
for digit in num:
  digit_count[int(digit)] += 1
        print("Number of occurrences of each digit in", num, ":")
        for i in range(10):
    print(i, ":", digit_count[i])
```

Output

```
1.  Enter a value: 1234234
    Not Palindrome
    1 appears 1 times
    2 appears 2 times
    3 appears 2 times
    4 appears 2 times
2.  Enter a value: 12321
    Palindrome
    1 appears 2 times
    2 appears 2 times
     3 appears 1 times
```

2. Aim: Demonstrating creation of functions, passing parameters and return values

2 a) Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

```
def f(n):
    if n <= 0:
        return "Invalid input"
    elif n == 1:
        return 0
    elif n == 2:
        return 1
    else:
        return int(f(n-1))+ int(f(n-2))

n = int(input("Enter a value for N: "))
print(f(n))
```

Output

```
1.  Enter a number: 5
    fn (5) = 3
    Enter a number: -1
    Error in input
```

2b. Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

```
def BinToDec(b):
    return int(b, 2)

def OctToHex(o):
    return hex(int(o, 8))

print("Enter the Binary Number: ", end="")
bnum = input()

dnum = BinToDec(bnum)
print("\nEquivalent Decimal Value = ", dnum)

print("Enter an Octal Number: ", end="")
onum = input()

hnum = OctToHex(onum)
print("\nEquivalent Hexadecimal Value =", hnum[2:].upper())
```

Output

```
1. Enter a binary number: 10111001
   185
2. Enter an octal number: 675
   1BD
```

3. Aim: Demonstration of manipulation of strings using string methods

3a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

```
sentence = input("Enter a sentence : ")
wordList = sentence.split(" ")
print("This sentence has", len(wordList), "words")
digCnt = upCnt = loCnt = 0
for ch in sentence:
    if '0' <= ch <= '9':
        digCnt += 1
    elif 'A' <= ch <= 'Z':
        upCnt += 1
    elif 'a' <= ch <= 'z':
        loCnt += 1
print("This sentence has", digCnt, "digits", upCnt, "upper case letters", loCnt, "lower case letters")
```

Output

```
1. Enter a sentence: Rama went to Deva raja market to pick 2 kgs of vegetable
   This sentence has 11 words
   This sentence has 1 digits 2 upper case letters 42 lower case letters
   String Similarity
```

3b) Write a Python program to find the string similarity between two given strings.

```
str1 = input("Enter String 1 \n")
str2 = input("Enter String 2 \n")
if len(str2) < len(str1):
    short = len(str2)
    long = len(str1)
else:
    short = len(str1)
    long = len(str2)
matchCnt = 0
for i in range(short):
    if str1[i] == str2[i]:
        matchCnt += 1
print("Similarity between two said strings:")
print(matchCnt/long)
```

Output

```
1.  Enter String 1
Python Exercises
Enter String 2
Python Exercises
Similarity between two said strings:
1.0

2.  Enter String 1
Python Exercises
Enter String 2
Python Exercise
Similarity between two said strings:
0.9375
```

4.Aim: Discuss different collections like list, tuple and dictionary

4.a) Write a python program to implement insertion sort and merge sort using lists.
import random

```
import random

def merge_sort(lst):
    if len(lst) > 1:
        mid = len(lst) // 2
        left_half = lst[:mid]
        right_half = lst[mid:]

        merge_sort(left_half)
        merge_sort(right_half)

        i = j = k = 0

        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                lst[k] = left_half[i]
                i += 1
            else:
                lst[k] = right_half[j]
                j += 1
            k += 1

        while i < len(left_half):
            lst[k] = left_half[i]
            i += 1
            k += 1

        while j < len(right_half):
            lst[k] = right_half[j]
            j += 1
            k += 1

    return lst


def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

my_list = []

for i in range(10):
    my_list.append(random.randint(0, 999))

print("\nUnsorted List")
print(my_list)
print("Sorting using Insertion Sort")
insertion_sort(my_list)
print(my_list)


my_list = []

for i in range(10):
    my_list.append(random.randint(0, 999))

print("\nUnsorted List")
print(my_list)
print("Sorting using Merge Sort")
merge_sort(my_list)
print(my_list)
```

Output

```
Unsorted List
[932, 111, 226, 685, 543, 589, 918, 539, 294, 717]
Sorting using Insertion Sort
[111, 226, 294, 539, 543, 589, 685, 717, 918, 932]
Unsorted List
[613, 176, 828, 265, 65, 326, 359, 919, 514, 868]
Sorting using Merge Sort
[65, 176, 265, 326, 359, 514, 613, 828, 868, 919]
```

4b) Write a program to convert roman numbers in to integer values using dictionaries.

```
def roman2Dec(romStr):
    roman_dict ={'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
    # Analyze string backwards
    romanBack = list(romStr)[::-1]
    value = 0
    # To keep track of order
    rightVal = roman_dict[romanBack[0]]
    for numeral in romanBack:
        leftVal = roman_dict[numeral]
        # Check for subtraction
        if leftVal < rightVal:
           value -= leftVal
        else:
            value += leftVal
        rightVal = leftVal
    return value


romanStr = input("Enter a Roman Number : ")
print(roman2Dec(romanStr))
```

Output

```
1. Enter a Roman Number: XVII
   17
2. Enter a Roman Number: MLXVI
    1066
```

5. **Aim: Demonstration of pattern recognition with and without using regular expressions**

5a) Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

```
import re
def isphonenumber(numStr):
 if len(numStr) != 12:
 return False
 for i in range(len(numStr)):
 if i==3 or i==7:
 if numStr[i] != "-":
 return False
 else:
 if numStr[i].isdigit() == False:
 return False
 return True
```

```
def chkphonenumber(numStr):
 ph_no_pattern = re.compile(r'^\d{3}-\d{3}-\d{4}$')
 if ph_no_pattern.match(numStr):
 return True
 else:
 return False
ph_num = input("Enter a phone number : ")
print("Without using Regular Expression")
if isphonenumber(ph_num):
 print("Valid phone number")
else:
 print("Invalid phone number")
print("Using Regular Expression")
if chkphonenumber(ph_num):
 print("Valid phone number")
else:
 print("Invalid phone number")
```

Output

```
1. Enter a phone number: 444-654-5656
   Without using Regular Expression
   Valid phone number
   Using Regular Expression
   Valid phone number

2. Enter a phone number: 45A4-444-878
   Without using Regular Expression
   Invalid phone number
   Using Regular Expression
   Invalid phone number
```

5b) Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com)

```
import re
# Define the regular expression for phone numbers
phone_regex = re.compile(r'\+\d{12}')
email_regex = re.compile(r'[A-Za-z0-9._]+@[A-Za-z0-9]+\.[A-Z|a-z]{2,}')
# Open the file for reading
with open('example.txt', 'r') as f:
 # Loop through each line in the file
 for line in f:
 # Search for phone numbers in the line
 matches = phone_regex.findall(line)
 # Print any matches found
```

```
    for match in matches:
    print(match)
    matches = email_regex.findall(line)
    # Print any matches found
    for match in matches:
    print(match)
```

Output

```
+918151894220
+829392938876
+918768456234
prakash81.82@gmail.in
```

6. Aim: Demonstration of reading, writing and organizing files

6a) Write a python program to accept a file name from the user and perform the following operations

- **Display the first N line of the file**
- **Find the frequency of occurrence of the word accepted from the user in the file**

```
import os.path
import sys
fname = input("Enter the filename : ")
if not os.path.isfile(fname):
 print("File", fname, "doesn't exists")
 sys.exit(0)
infile = open(fname, "r")
lineList = infile.readlines()
for i in range(20):
 print(i+1, ":", lineList[i])
word = input("Enter a word : ")
cnt = 0
for line in lineList:
 cnt += line.count(word)
print("The word", word, "appears", cnt, "times in the file")
```

Output

```
Enter the filename: example.txt
1 : this is phone number +918151894220
2 : no phone number here
3 : here we have one +829392938876
4 : we have an email prakash81.82@gmail.in and a number +918768456234
```

```
5 : nothing of that sort here
6 : Better hope the life-inspector doesn't come around while you have your
7 : life in such a mess.
8 : You can create your own opportunities this week. Blackmail a senior executive.
9 : Be different: conform.
10 : Be cheerful while you are alive.
11 : -- Phathotep, 24th Century B.C.
12 : Q: How many journalists does it take to screw in a light bulb?
13 : A: Three. One to report it as an inspired government program to bring
14 : light to the people, one to report it as a diabolical government plot

15 : to deprive the poor of darkness, and one to win a Pulitzer prize for
16 : reporting that Electric Company hired a light bulb-assassin to break
17 : the bulb in the first place.
18 : Q: Why did the astrophysicist order three hamburgers?
19 : A: Because he was hungry.
20 : Q: Why haven't you graduated yet?
Enter a word : the
The word the appears 7 times in the file
Zip operation on a folder
```

6b) 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")
```

Output

```
Enter Directory name that you want to backup : zipDemo
```

```
Archive myZip.zip created successfully
```

7. Aim: Demonstration of the concepts of classes, methods, objects and inheritance

7a) By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle.

```
import math
class Shape:
 def __init__(self):
    self.area = 0
    self.name = ""

 def showArea(self):
    print("The area of the", self.name, "is", self.area, "units")

class Circle(Shape):
 def __init__(self,radius):
    self.area = 0
    self.name = "Circle"
    self.radius = radius

 def calcArea(self):
    self.area = math.pi * self.radius * self.radius

class Rectangle(Shape):
 def __init__(self,length,breadth):
    self.area = 0
    self.name = "Rectangle"
    self.length = length
    self.breadth = breadth

 def calcArea(self):
    self.area = self.length * self.breadth
class Triangle(Shape):
 def __init__(self,base,height):
    self.area = 0
    self.name = "Triangle"
    self.base = base
    self.height = height

 def calcArea(self):
    self.area = self.base * self.height / 2
```

```
c1 = Circle(5)
c1.calcArea()
c1.showArea()
r1 = Rectangle(5, 4)
r1.calcArea()
r1.showArea()
t1 = Triangle(3, 4)
t1.calcArea()
t1.showArea()
```

OUTPUT:

```
The area of the Circle is 78.53981633974483 units
The area of the Rectangle is 20 units
The area of the Triangle is 6.0 units
```

7b) Write a python program by creating a class called Employee to store the details of Name, Employee_ID, Department and Salary, and implement a method to update salary of employees belonging to a given department.

```
class Employee:
 def __init__(self):
   self.name = ""
   self.empId = ""
   self.dept = ""
   self.salary = 0

 def getEmpDetails(self):
   self.name = input("Enter Employee name : ")
   self.empId = input("Enter Employee ID : ")
   self.dept = input("Enter Employee Dept : ")
   self.salary = int(input("Enter Employee Salary : "))

 def showEmpDetails(self):
   print("Employee Details")
   print("Name : ", self.name)
   print("ID : ", self.empId)
   print("Dept : ", self.dept)
   print("Salary : ", self.salary)

 def updtSalary(self):
   self.salary = int(input("Enter new Salary : "))
   print("Updated Salary", self.salary)
```

```
e1 = Employee()
e1.getEmpDetails()
e1.showEmpDetails()
e1.updtSalary()
```

OUTPUT:

```
Enter Employee name : Manish
Enter Employee ID : 2727
Enter Employee Dept : R&D
Enter Employee Salary : 500000

Employee Details
Name :  Manish
ID :  2727
Dept :  R&D
Salary :  500000
Enter new Salary : 600000
Updated Salary 600000
```

8. Aim: Demonstration of classes and methods with polymorphism and overriding

8a) Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of polymorphism and inheritance.

```
class PaliStr:
 def __init__(self):
  self.isPali = False

 def chkPalindrome(self, myStr):
    if myStr == myStr[::-1]:
       self.isPali = True
    else:
       self.isPali = False

    return self.isPali

class PaliInt(PaliStr):
 def __init__(self):
    self.isPali = False

 def chkPalindrome(self, val):
    temp = val
    rev = 0
    while temp != 0:
       dig = temp % 10
       rev = (rev*10) + dig
```

```
                temp = temp //10

            if val == rev:
                self.isPali = True
            else:
                self.isPali = False
            return self.isPali
st = input("Enter a string : ")
stObj = PaliStr()
if stObj.chkPalindrome(st):
 print("Given string is a Palindrome")
else:
 print("Given string is not a Palindrome")
val = int(input("Enter a integer : "))
intObj = PaliInt()
if intObj.chkPalindrome(val):
 print("Given integer is a Palindrome")
else:
 print("Given integer is not a Palindrome")
```

OUTPUT:

```
1. Enter a string : malayalam
   Given string is a Palindrome
   Enter a integer : 56767
   Given integer is not a Palindrome

2. Enter a string : sunshine
   Given string is not a Palindrome
   Enter a integer : 2727
   Given integer is not a Palindrome
```

9. Aim: Demonstration of working with excel spreadsheets and web scraping

9a) Write a python program to download the all XKCD comics

```
import requests
import os
from bs4 import BeautifulSoup
# Set the URL of the first XKCD comic
url = 'https://xkcd.com/1/'
# Create a folder to store the comics
if not os.path.exists('xkcd_comics'):
 os.makedirs('xkcd_comics')
# Loop through all the comics
while True:
 # Download the page content
 res = requests.get(url)
 res.raise_for_status()
 # Parse the page content using BeautifulSoup
 soup = BeautifulSoup(res.text, 'html.parser')
 # Find the URL of the comic image
 comic_elem = soup.select('#comic img')
 if comic_elem == []:
 print('Could not find comic image.')
 else:
 comic_url = 'https:' + comic_elem[0].get('src')
 # Download the comic image
 print(f'Downloading {comic_url}...')
 res = requests.get(comic_url)
 res.raise_for_status()
 # Save the comic image to the xkcd_comics folder
 image_file = open(os.path.join('xkcd_comics', os.path.basename(comic_url)), 'wb')
 for chunk in res.iter_content(100000):
 image_file.write(chunk)
 image_file.close()
 # Get the URL of the previous comic
 prev_link = soup.select('a[rel="prev"]')[0]
 if not prev_link:
 break
 url = 'https://xkcd.com' + prev_link.get('href')
print('All comics downloaded.')
```

Output

```
Downloading https://imgs.xkcd.com/comics/barrel_cropped_(1).jpg...
Downloading https://imgs.xkcd.com/comics/radians_are_cursed.png...
Downloading https://imgs.xkcd.com/comics/presents_for_biologists.png...
Downloading https://imgs.xkcd.com/comics/launch_window.png...
Downloading https://imgs.xkcd.com/comics/obituary_editor.png...
Downloading https://imgs.xkcd.com/comics/fanservice.png...
Downloading https://imgs.xkcd.com/comics/hand_dryers.png...
```

9b) Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet

```
from openpyxl import Workbook
from openpyxl.styles import Font
wb = Workbook()
sheet = wb.active
sheet.title = "Language"
wb.create_sheet(title = "Capital")
lang = ["Kannada", "Telugu", "Tamil"]
state = ["Karnataka", "Telangana", "Tamil Nadu"]
capital = ["Bengaluru", "Hyderabad", "Chennai"]
code =['KA', 'TS', 'TN']
sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Language"
sheet.cell(row = 1, column = 3).value = "Code"
ft = Font(bold=True)
```

```
for row in sheet["A1:C1"]:
 for cell in row:
 cell.font = ft
for i in range(2,5):
 sheet.cell(row = i, column = 1).value = state[i-2]
 sheet.cell(row = i, column = 2).value = lang[i-2]
 sheet.cell(row = i, column = 3).value = code[i-2]
wb.save("demo.xlsx")
sheet = wb["Capital"]
sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Capital"
sheet.cell(row = 1, column = 3).value = "Code"
ft = Font(bold=True)
for row in sheet["A1:C1"]:
 for cell in row:
 cell.font = ft
for i in range(2,5):
 sheet.cell(row = i, column = 1).value = state[i-2]
 sheet.cell(row = i, column = 2).value = capital[i-2]
 sheet.cell(row = i, column = 3).value = code[i-2]
wb.save("demo.xlsx")
srchCode = input("Enter state code for finding capital ")
for i in range(2,5):
 data = sheet.cell(row = i, column = 3).value
 if data == srchCode:
 print("Corresponding capital for code", srchCode, "is", sheet.cell(row = i, column =2).value)
sheet = wb["Language"]
srchCode = input("Enter state code for finding language ")
for i in range(2,5):
 data = sheet.cell(row = i, column = 3).value
 if data == srchCode:
print("Corresponding language for code", srchCode, "is", sheet.cell(row = i, column =2).value)
wb.close()
```

Output

```
Enter state code for finding capital KA
Corresponding capital for code KA is Bengaluru
Enter state code for finding language TS
Corresponding language for code TS is Telugu
Merge selected pages from Multiple PDFs to a new PDF
```

10. Aim: Demonstration of working with PDF, word and JSON files

10a) Write a python program to combine select pages from many PDFs from PyPDF2 import Pdf Writer, Pdf Reader

```
num = int(input("Enter page number you want combine from multiple documents "))
```

```
pdf1 = open('birds.pdf', 'rb')
pdf2 = open('birdspic.pdf', 'rb')
pdf_writer = PdfWriter()
pdf1_reader = PdfReader(pdf1)
page = pdf1_reader.pages[num - 1]
pdf_writer.add_page(page)
pdf2_reader = PdfReader(pdf2)
page = pdf2_reader.pages[num - 1]
pdf_writer.add_page(page)
with open('output.pdf', 'wb') as output:
 pdf_writer.write(output)
```

Output

This program allows you to extract specific pages from two PDF files, “birds.pdf” and “birdspic.pdf,” by entering the page numbers as user input. Once you input the desired page numbers, the program fetches those pages from both PDF files and combines them into a new file called “output.pdf.” This way, you can easily compile the desired pages from multiple PDF files into one document for your convenience.

```
Enter page number you want combine from multiple documents 3
birdsDownload
birdspicDownload
outputDownload
```

10b) Write a python program to fetch current weather data from the JSON file

```
import json
# Load the JSON data from file
with open('weather_data.json') as f:
 data = json.load(f)
# Extract the required weather data
current_temp = data['main']['temp']
humidity = data['main']['humidity']
weather_desc = data['weather'][0]['description']
# Display the weather data
print(f"Current temperature: {current_temp}°C")
print(f"Humidity: {humidity}%")
print(f"Weather description: {weather_desc}")
```

JSON File:

```
{
 "coord": {
 "lon": -73.99,
 "lat": 40.73
 },
 "weather": [
 {
 "id": 800,
 "main": "Clear",
 "description": "clear sky",
 "icon": "01d"
 }
],
 "base": "stations",
 "main": {
 "temp": 15.45,
 "feels_like": 12.74,
 "temp_min": 14.44,
 "temp_max": 16.11,
 "pressure": 1017,
 "humidity": 64
 },
 "visibility": 10000,
 "wind": {
 "speed": 4.63,
 "deg": 180
 },
 "clouds": {
 "all": 1
 },
 "dt": 1617979985,
 "sys": {
 "type": 1,
 "id": 5141,
 "country": "US",
 "sunrise": 1617951158,
 "sunset": 1618000213
 },
 "timezone": -14400,
 "id": 5128581,
 "name": "New York",
 "cod": 200
}
```

Output

```
Current temperature: 15.45°C
Humidity: 64%
Weather description: clear sky
```